LABORATORIO DI SICUREZZA DEI SISTEMI E PRIVACY

CICLO DI SVILUPPO SICURO DEL CODICE

ANNO ACCADEMICO 2024-2025

CYBERLOOP

QUALITÀ ISO 9001
SICUREZZA ISO 27001

CLASSIFICAZIONE

TLP:GREEN

VERSIONE	V1
DATA	09/05/2025
AUTORE	Marco Canducci

INDICE

CYBERLOOP

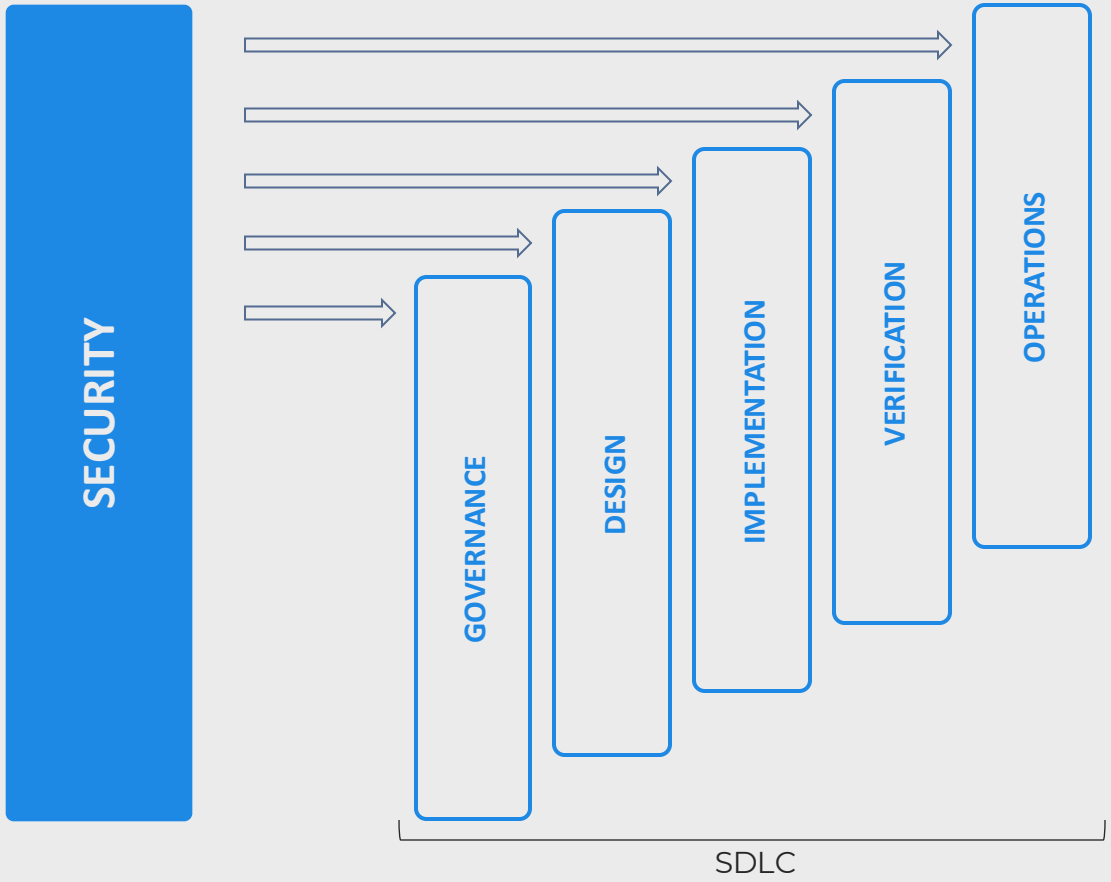
INTRODUZIONE

CICLO DI SVILUPPO SICURO DEL CODICE.

INTRODUZIONE

Il ciclo di vita dello sviluppo del software (**SDLC**) è un processo strutturato che consente lo sviluppo di software di alta qualità e in maniera efficiente.

Il Secure SDLC (**SSDLC**) integra la sicurezza nel processo.

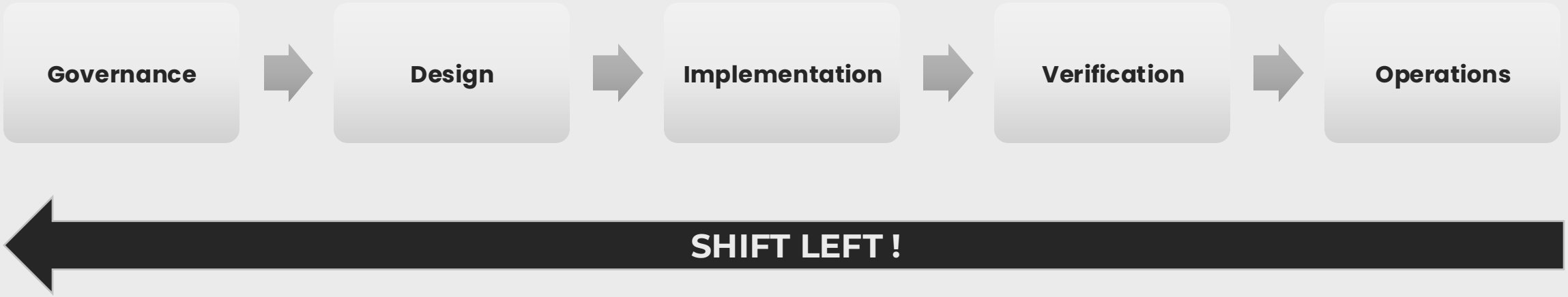


CYBERLOOP

SHIFT LEFT

La metodologia "Shift Left" è un approccio allo sviluppo del software che sposta l'attenzione sulla sicurezza **fin dalle prime fasi del ciclo di vita del software**.

Questo approccio è efficace nel garantire che la sicurezza sia presa in considerazione da subito nella progettazione della soluzione e nel permettere una rilevazione precoce di bug e problemi di sicurezza, **riducendo così i costi e il rischio di possibili violazioni.**



CYBERLOOP

MODELLO DI RIFERIMENTO

CICLO DI SVILUPPO SICURO DEL CODICE: OWASP SAMM.

MODELLO DI RIFERIMENTO

Il principale e più importante standard che regola il SSDLC è redatto da OWASP, ed è denominato SAMM: **Software Assurance Maturity Model**.

> *«La nostra missione è fornire un modo efficace e misurabile per tutti i tipi di organizzazioni di analizzare e migliorare la propria postura sulla sicurezza del software.*
> *Vogliamo sensibilizzare ed educare le organizzazioni su come progettare, sviluppare e distribuire software sicuro attraverso il nostro modello di autovalutazione»*

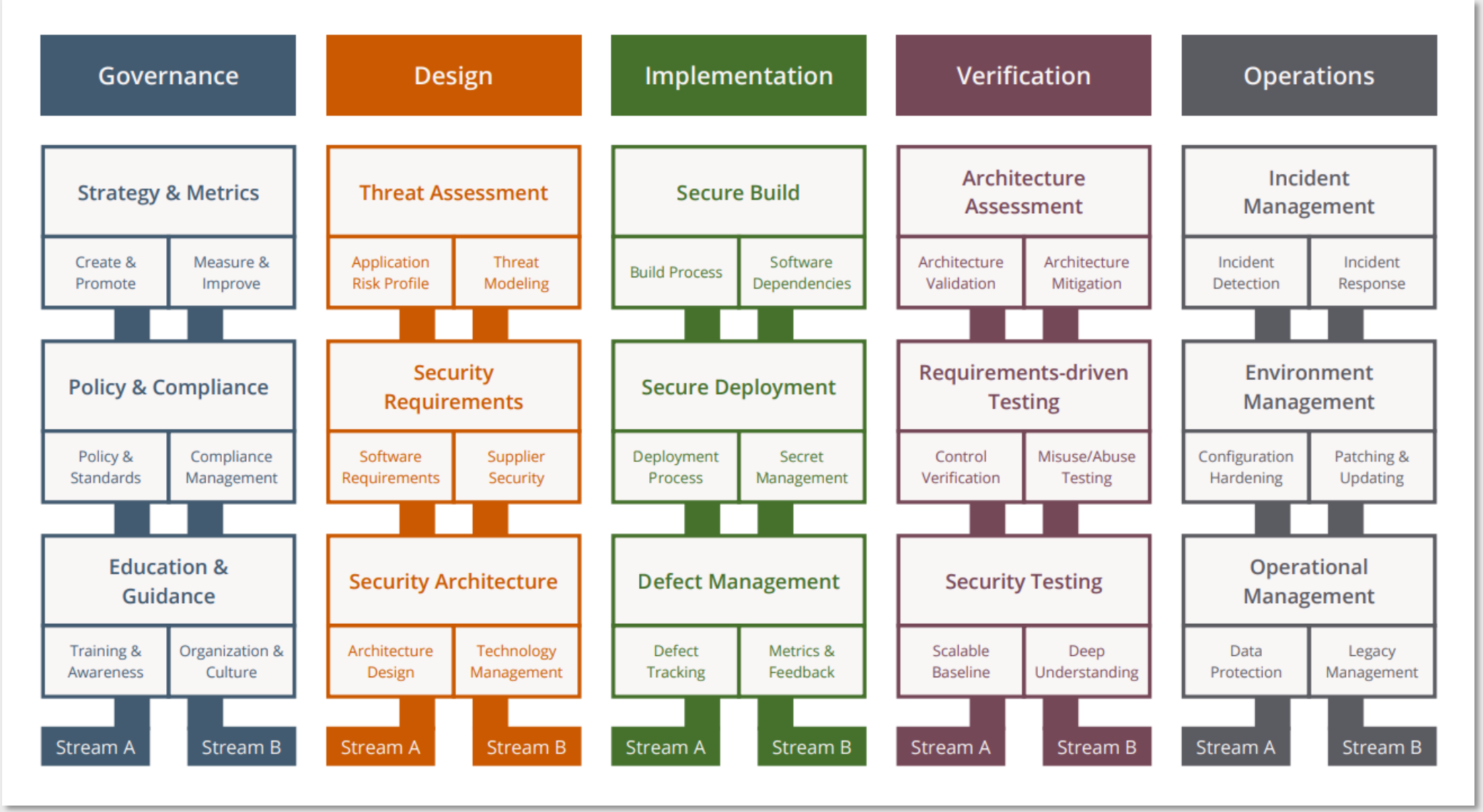
MODELLO DI RIFERIMENTO

OWASP SAMM supporta l'intero ciclo di vita del software ed è tecnologicamente e processualmente agnostico. Lo standard SAMM è pensato per essere evolutivo e guidato dal rischio, poiché non esiste una singola ricetta che funzioni per tutte le organizzazioni.

Pertanto, lo scopo è mantenere uno standard che sia:

- **MISURABILE** tramite livelli di maturità definiti tra le pratiche di sicurezza
- **ESEGUIBILE** tramite percorsi chiari per migliorare i livelli di maturità
- **VERSATILE** essendo agnostico rispetto a tecnologia, processo e organizzazione

OWASP SAMM V.2

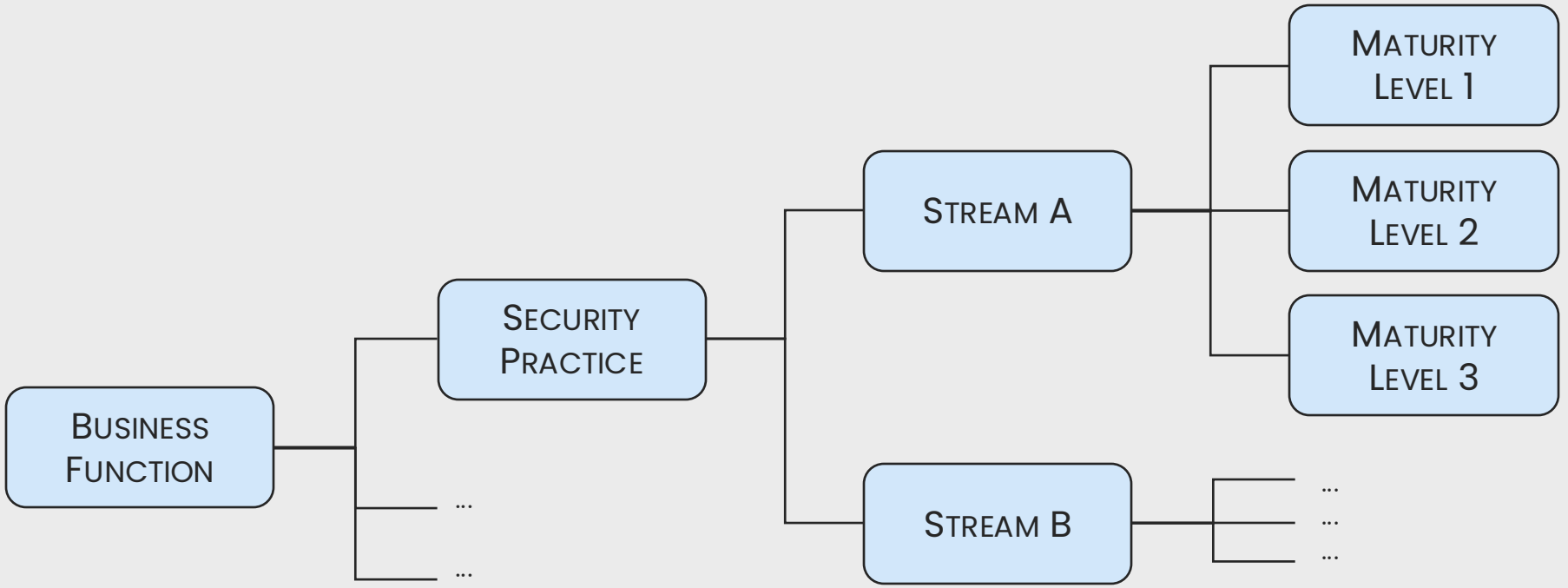


CYBERLOOP

OWASP SAMM V.2

Come visibile dalla slide precedente, SAMM si basa su 15 Pratiche di Sicurezza (**Security Practice**) raggruppate nelle 5 Funzioni Aziendali (**Business Function**) di *Governance*, *Design*, *Implementation*, *Verification* e *Operation*.

Ogni pratica di sicurezza prevede due tipologie di interventi (**Stream A e Stream B**), classificati in 3 livelli di maturità (**Maturity Level**).



CYBERLOOP

LIVELLI MATURITÀ OWASP SAMM V.2

Per ciascuna «practice» di sicurezza, **SAMM definisce tre livelli di maturità** come obiettivi di sicurezza, progressivamente più sofisticati e definiti da metriche di successo più stringenti rispetto al livello precedente.

I dettagli per ciascun livello differiscono tra le diverse pratiche di sicurezza, ma generalmente i singoli livelli rappresentano:

- **Livello 0** – Punto di partenza, rappresenta le attività che non soddisfano i requisiti di sicurezza
- **Livello 1** – Comprensione iniziale e fornitura ad hoc della pratica di sicurezza
- **Livello 2** – Aumento dell'efficienza e/o dell'efficacia della pratica di sicurezza
- **Livello 3** – Padronanza completa della pratica di sicurezza su larga scala

CYBERLOOP

LE FASI DEL PROCESSO DI SSDLC

CICLO DI SVILUPPO SICURO DEL CODICE SECONDO OWASP SAMM.

Le fasi del processo di SSDLC

Governance

GOVERNANCE

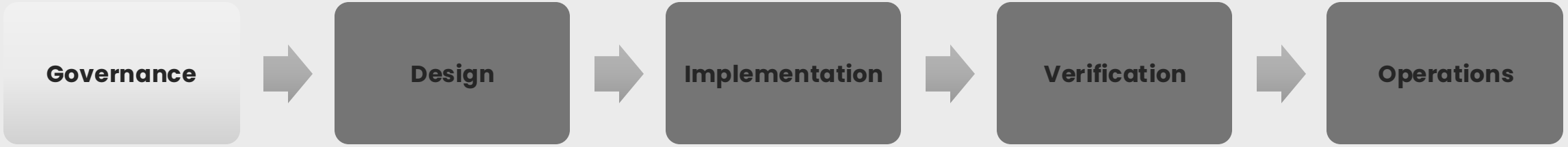
La fase di Governance si concentra sui processi e sulle modalità con cui un'organizzazione gestisce le attività di sviluppo del software.

Più specificamente, si considerano aspetti di sicurezza che influenzano le funzioni aziendali trasversali, nonché i processi aziendali stabiliti a livello organizzativo.

La fase di Governance è suddivisa in:

- **Strategy** & **Metrics**: viene costituita la base delle attività di sviluppo sicuro tramite la definizione di un piano generale.
- **Policy** & **Compliance**: copre l'adeguamento agli standard, così come il rispetto delle normative vigenti.
- **Education** & **Guidance**: si concentra sull'aumento delle competenze relative allo sviluppo sicuro all'interno dell'organizzazione aziendale.

GOVERNANCE



Security Practices	Stream A	Stream B
Strategy & Metrics	Create & Promote	Measure & Improve
Policy & Compliance	Policy & Standard	Compliance Management
Education & Guidance	Training & Awareness	Organization & Culture

STRATEGY & METRICS

Stream A – Create & Promote

Questo filone riguarda la creazione e la promozione di una roadmap per la sicurezza delle applicazioni sviluppate, al fine di definire gli obiettivi dell'azienda su questo tema e aumentare l'allineamento e la collaborazione tra le parti interessate.

Stream B – Measure & Improve

Questo filone mira a garantire la validità ed il miglioramento della roadmap per la messa in sicurezza degli applicativi attraverso opportune misurazioni all'interno dell'organizzazione.

STRATEGY & METRICS

Stream A - Create & Promote	Stream B - Measure & Improve
Livello di maturità 1: Identificare gli obiettivi e i mezzi per misurare l'efficacia del programma di sicurezza.	
Identificare i fattori trainanti (più rilevanti) in relazione alla sua tolleranza al rischio.	Definire metriche tramite un'analisi sull'efficacia del programma di messa in sicurezza delle applicazioni.
Livello di maturità 2: Stabilire una roadmap strategica unificata per la sicurezza del software all'interno dell'organizzazione.	
Stabilire una strategia unificata per la sicurezza delle applicazioni.	Stabilire obiettivi e KPI per misurare l'efficacia del programma di messa in sicurezza.
Livello di maturità 3: Allineare gli sforzi di sicurezza con gli indicatori organizzativi rilevanti e i valori degli asset strategici.	
Allineare il programma di sicurezza delle applicazioni per supportare la crescita dell'organizzazione.	Influenzare la strategia basandosi sulle metriche e sulle esigenze organizzative.

POLICY & COMPLIANCE

Stream A – Policy & Standard

Questo filone si concentra sulla gestione delle politiche e degli standard, oltre che sulla loro fornitura per supportare l'integrazione nel ciclo di vita dello sviluppo del software (SDLC).

Stream B – Compliance Management

Questo filone si concentra sull'individuazione e sulla fornitura dei requisiti di conformità per supportare l'integrazione nel ciclo di vita dello sviluppo del software (SDLC).

POLICY E COMPLIANCE

Stream A – Policy & Standard	Stream B – Compliance Management
Livello di maturità 1: Identificare e documentare i fattori di governance e conformità rilevanti per l'organizzazione.	
Determinare un livello di sicurezza di base dettato le politiche e gli standard dell'organizzazione.	Identificare i driver e i requisiti di conformità di terze parti e mapparli alle politiche e agli standard esistenti.
Livello di maturità 2: Stabilire un livello base di sicurezza e di conformità specifiche per gli applicativi che si stanno sviluppando.	
Sviluppare requisiti di sicurezza applicabili a tutte le applicazioni.	Pubblicare sia requisiti specifici di conformità per le applicazioni sia indicazioni per i test.
Livello di maturità 3: Misurare l'adesione alle politiche, agli standard e ai requisiti di terze parti.	
Misurare e riportare lo stato dell'aderenza individuale delle applicazioni alle politiche e agli standard.	Misurare e riportare la conformità delle applicazioni ai requisiti di terze parti.

EDUCATION & GUIDANCE

Stream A – Training & Awareness

La formazione e la sensibilizzazione si concentrano sull'aumento della conoscenza complessiva relativa alla sicurezza del software tra i diversi stakeholder all'interno dell'organizzazione.

Stream B – Organization & Culture

L'organizzazione e la cultura aziendale si concentrano sulla promozione della sicurezza delle applicazioni all'interno dell'organizzazione come un importante fattore di successo di un progetto di SDLC.

EDUCATION & GUIDANCE

Stream A – Training & Awareness	Stream B – Organization & Culture
Livello di maturità 1: Offrire al personale accesso alle risorse sui temi dello sviluppo e del deployment sicuro.	
Fornire formazione a tutto il personale coinvolto nello sviluppo del software.	Individuare un «Security champion» all'interno di ciascun team di sviluppo.
Livello di maturità 2: Garantire che l'educazione e le linee guida sulla sicurezza siano fornite sistematicamente a tutti i soggetti interessati all'interno dell'organizzazione. Questo include sviluppatori, tester, project manager e altri ruoli coinvolti nel ciclo di vita dello sviluppo software.	
Fornire formazione specifica per ruolo, contenuti avanzati e aggiornarli sulla base dei feedback.	Garantire il coinvolgimento della dirigenza e una comunicazione interna efficace nella stesura delle politiche di sicurezza
Livello di maturità 3: Garantire una formazione completa, certificata, incentivandone e monitorandone il miglioramento.	
Introdurre programmi di formazione completi, certificazioni e modalità di monitoraggio della formazione.	Implementare programmi di incentivazione e favorire la collaborazione interfunzionale.

Le fasi del processo di SSDLC

Design

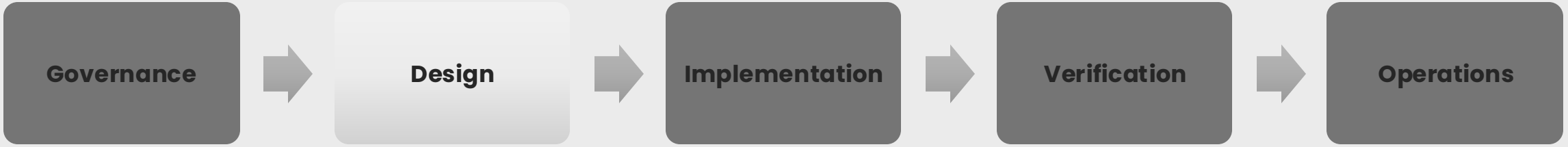
DESIGN

La **fase di design** riguarda i processi e le attività relativi a come un'organizzazione definisce gli obiettivi e crea software all'interno dei progetti di sviluppo. In generale, ciò includerà la raccolta dei requisiti, la specifica dell'architettura ad alto livello ed il design dettagliato.

La fase di Design è suddivisa in:

- **Valutazione delle minacce**: Questa pratica si concentra sull'identificazione delle potenziali minacce nelle applicazioni.
- **Requisiti di sicurezza**: Questa pratica si concentra sulla definizione di requisiti di sicurezza appropriati per il tuo software e per i fornitori di software.
- **Architettura di sicurezza**: Si concentra sulla gestione dei rischi architetturali per la soluzioni software in sviluppo.

DESIGN



Security Practices	Stream A	Stream B
Threat Assessment	Application Risk Profile	Threat Modeling
Security Requirements	Software Requirements	Supplier Security
Security Architecture	Architecture Design	Technology Management

THREAT ASSESMENT

Il **Threat Assessment** si concentra sull'identificazione e la comprensione dei rischi a livello di progetto, basati sulla funzionalità del software e sulle caratteristiche dell'ambiente di esecuzione.

Attraverso l’analisi delle minacce e degli attacchi più probabili l'organizzazione nel suo complesso opera in modo più efficace attraverso decisioni migliori sulla prioritizzazione delle iniziative di sicurezza.

THREAT ASSESMENT

Stream A – Application Risk Profile

Il profilo di rischio applicativo aiuta ad identificare quali applicazioni potrebbero rappresentare una seria minaccia per l'organizzazione se queste venissero attaccate o violante.

Stream B – Threat Modeling

La modellazione delle minacce è utile come supporto al team di sviluppo software al fine di capire quali rischi e quali minacce sussistano in ciò che sta venendo sviluppato, cosa potrebbe andare storto e come tali rischi potrebbero essere mitigati o risolti.

THREAT ASSESMENT

Stream A – Application Risk Profile	Stream B – Threat Modeling
Livello di maturità 1: Identificazione delle minacce di alto livello per l'organizzazione e per i singoli progetti.	
Viene eseguita una valutazione di base del rischio dell'applicazione per comprendere la probabilità e l'impatto di un attacco.	Effettuare un modello delle minacce basato sul rischio, utilizzando sessioni di brainstorming che utilizzino diagrammi pre-esistenti con semplici checklist di minacce.
Livello di maturità 2: Standardizzazione e analisi delle minacce a livello aziendale.	
Comprendere il rischio per tutte le applicazioni nell'organizzazione centralizzando l'inventario del profilo di rischio.	Standardizzare la formazione sulla modellazione delle minacce, i processi e gli strumenti per poter scalare su tutta l'organizzazione.
Livello di maturità 3: Miglioramento proattivo della copertura delle minacce in tutta l'organizzazione.	
Rivedere periodicamente i profili di rischio delle applicazioni a intervalli regolari per garantire maggior efficacia e precisione.	Ottimizzazione continua e automazione della metodologia di modellazione delle minacce.

SECURITY REQUIREMENTS

Stream A – Software Requirements

I requisiti software specificano gli obiettivi e le aspettative per proteggere i servizi e i dati relativi all'applicazione.

Stream B – Supplier Security

La sicurezza dei fornitori riguarda i requisiti relativi alle organizzazioni fornitrici all'interno del contesto di sviluppo dell'applicazione, in particolare per lo sviluppo esternalizzato.

SECURITY REQUIREMENTS

Stream A	Stream B
Livello di maturità 1: Considerare esplicitamente la sicurezza durante il processo di definizione dei requisiti software.	
Gli obiettivi di sicurezza dell'applicazione ad alto livello vengono mappati sui requisiti funzionali.	Valutare il fornitore in base ai requisiti di sicurezza dell'organizzazione.
Livello di maturità 2: Aumentare la granularità dei requisiti di sicurezza derivati dalle logiche aziendali e dai rischi conosciuti.	
Sono disponibili requisiti di sicurezza strutturati e utilizzati dai team di sviluppo.	Includere la sicurezza negli accordi con i fornitori al fine di garantire la conformità ai requisiti della propria organizzazione.
Livello di maturità 3: Imporre un processo di requisiti di sicurezza per tutti i progetti software e le dipendenze di terze parti.	
Costruire un framework di requisiti per consentire ai team di prodotto di utilizzarlo.	Garantire una corretta copertura della sicurezza per i fornitori esterni fornendo obiettivi chiari.

SECURITY REQUIREMENTS

	L1	L2	L3
Verificare che i token di sessione basati su cookie abbiano l'attributo 'Secure' impostato.	✓	✓	✓
Verificare che l'applicazione non registri credenziali o dettagli di pagamento. I token di sessione dovrebbero essere memorizzati nei registri solo sotto forma di hash, in modo irreversibile.	✓	✓	✓
Verificare che i security logs siano protetti dall'accesso e dalla modifica non autorizzati.		✓	✓
Verificare che sia in uso uno strumento di analisi del codice che possa rilevare codice potenzialmente dannoso.			✓

SECURE ARCHITECTURE

Stream A – Architecture Design

Il design di un'architettura software può influenzare significativamente la postura della sicurezza di un software: l'utilizzo di buone pratiche di sicurezza ne migliorerà, in generale, anche il design complessivo.

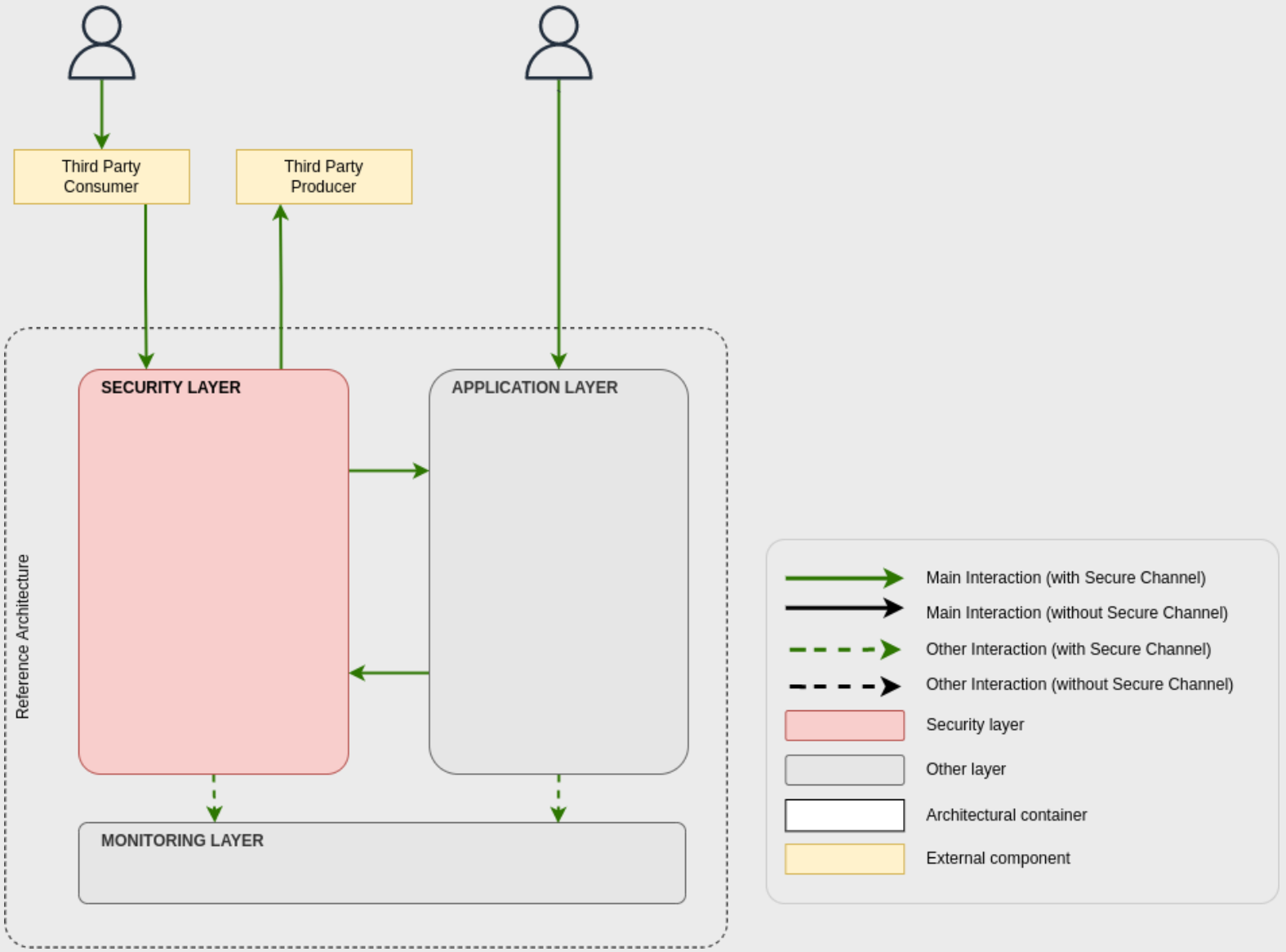
Stream B – Technology Management

Le tecnologie e i framework sono i pilastri di qualsiasi soluzione software. Le proprietà di sicurezza di questi devono essere esaminate per garantire un livello di sicurezza appropriato e per anticipare eventuali problemi.

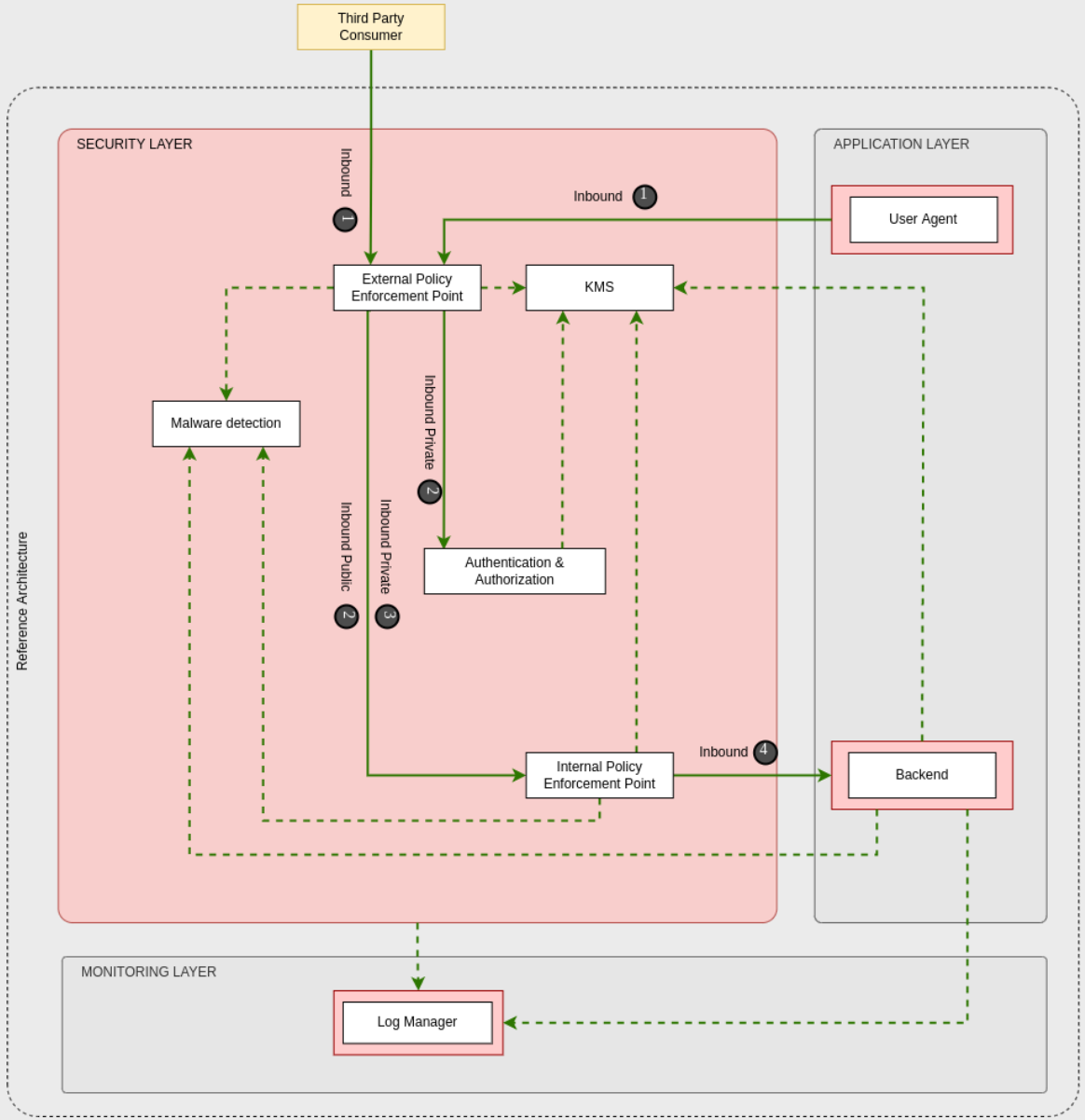
SECURE ARCHITECTURE

Stream A – Architecture Design	Stream B – Technology Management
Livello di maturità 1: Inserire la considerazione di orientamenti proattivi sulla sicurezza nel processo di progettazione del software.	
I team vengono formati sull'uso dei principi di base della sicurezza durante la fase di progettazione.	Raccogliere informazioni sulle tecnologie, i framework e le integrazioni all'interno della soluzione complessiva per identificare i rischi.
Livello di maturità 2: Direct the software design process toward known secure services and secure-by-default designs.	
Stabilire pattern di progettazione comuni e soluzioni di sicurezza per l'adozione.	Standardizzare le tecnologie e i framework da utilizzare in tutte le diverse applicazioni.
Livello di maturità 3: Controllare formalmente il processo di progettazione del software e convalidare l'utilizzo di componenti sicuri.	
Le architetture di riferimento vengono utilizzate e valutate continuamente per l'adozione e l'adeguatezza.	Imporre l'uso di tecnologie standard su tutti gli sviluppi software.

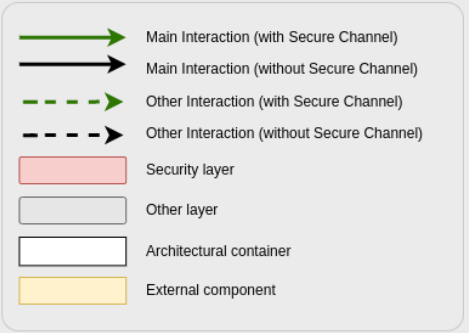
ARCHITETTURA SICURA



ARCHITETTURA SICURA

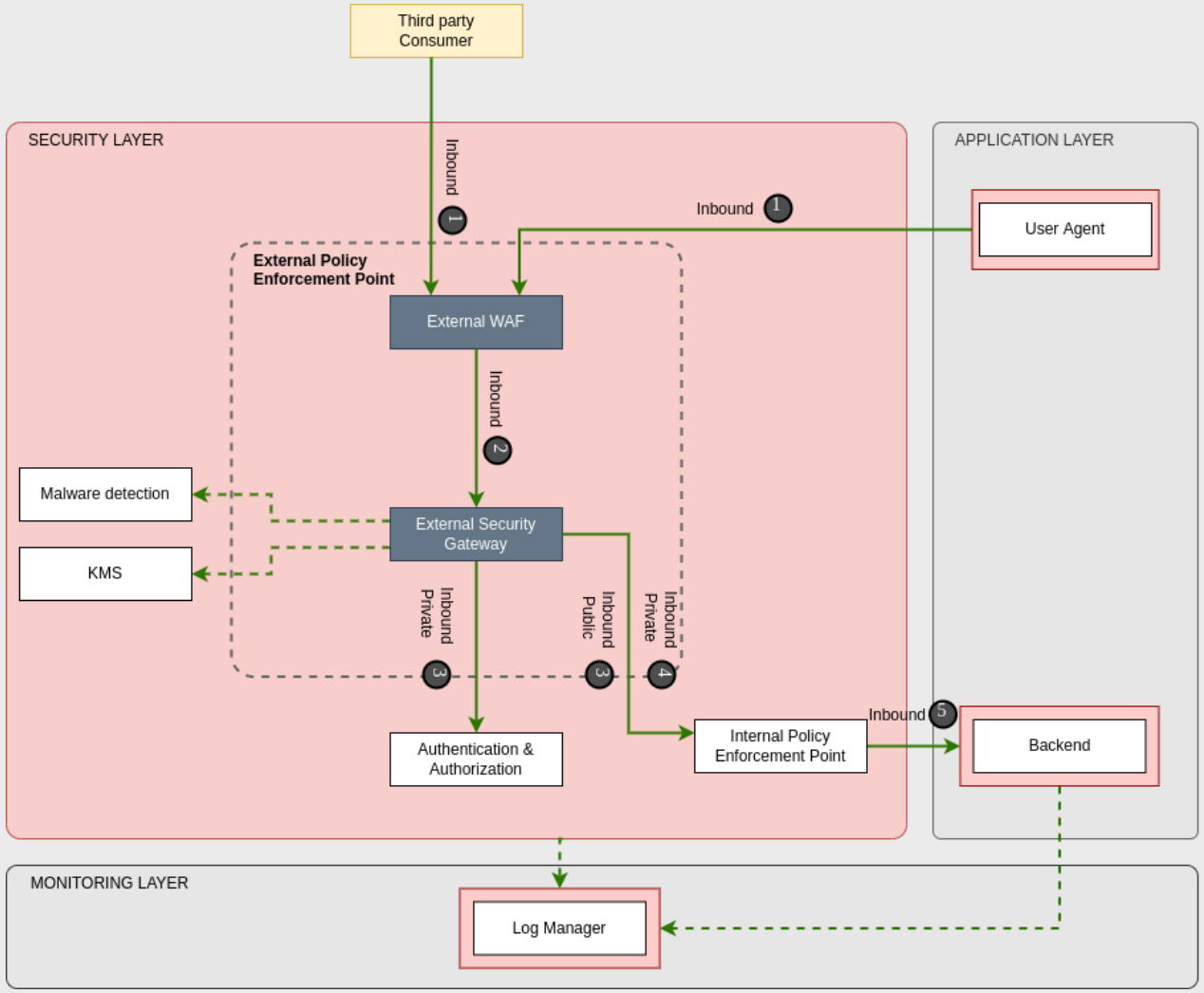


INBOUND COMMUNICATION

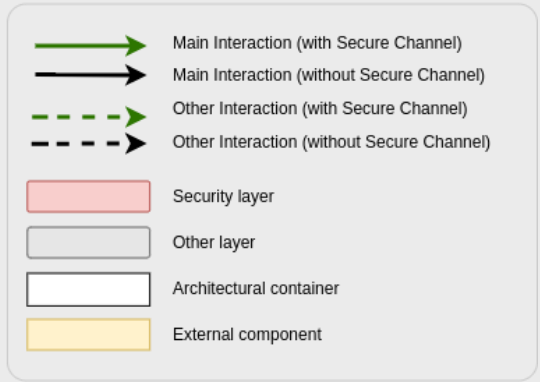


CYBERLOOP

ARCHITETTURA SICURA

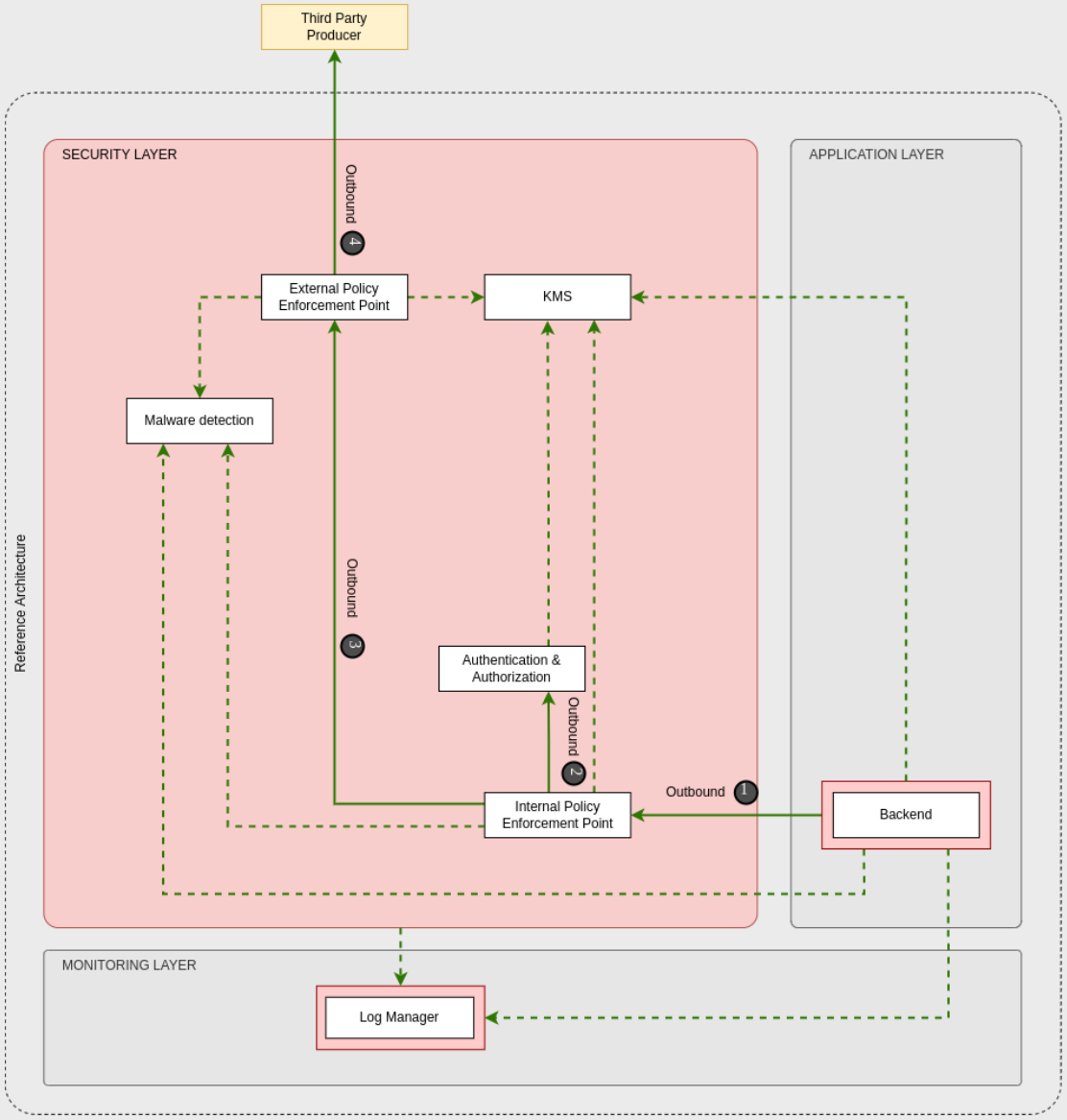


INBOUND COMMUNICATION

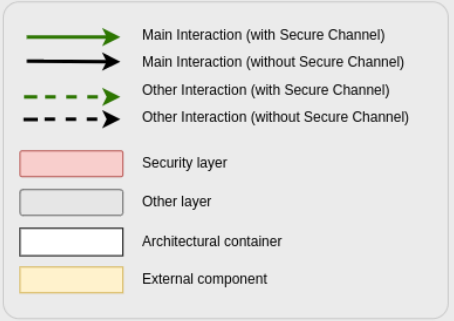


CYBERLOOP

ARCHITETTURA SICURA

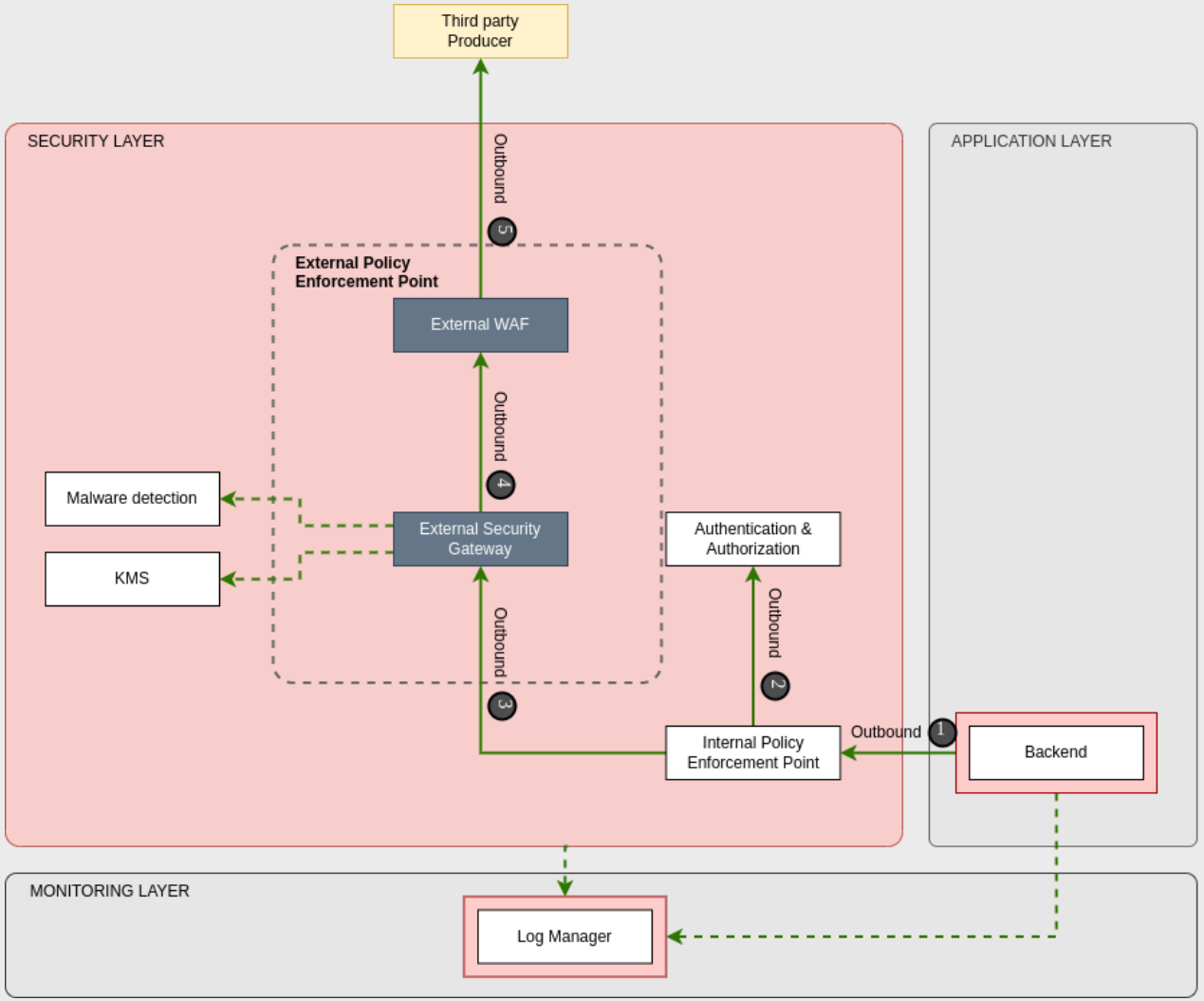


OUTBOUND COMMUNICATION

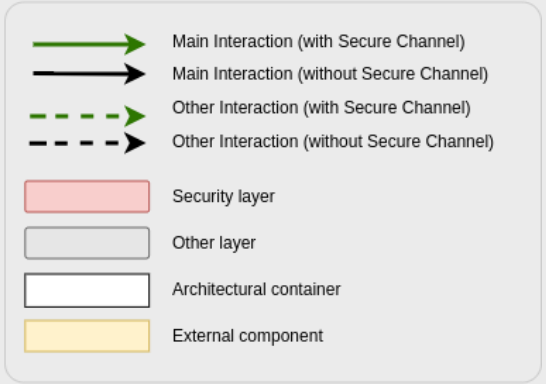


CYBERLOOP

ARCHITETTURA SICURA



OUTBOUND COMMUNICATION



Le fasi del processo di SSDLC

Implementation

IMPLEMENTATION

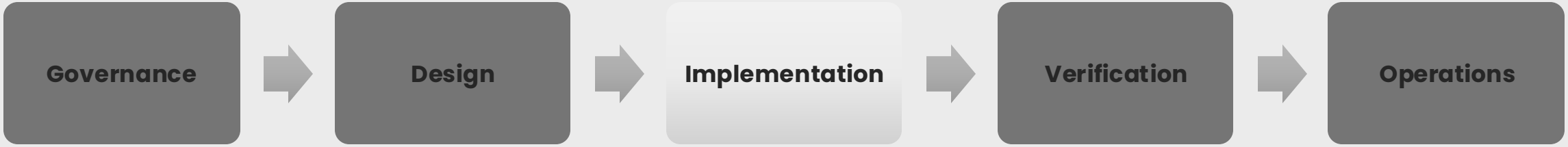
La fase di implementazione è focalizzata sui processi e sulle attività legate al modo in cui un'organizzazione costruisce e distribuisce i componenti software. Le attività all'interno della funzione di Implementazione hanno un forte impatto sulla vita quotidiana degli sviluppatori.

L'obiettivo comune è quello di rilasciare software che funzioni in modo affidabile con il minimo numero di vulnerabilità.

Divisa in 3 pratiche:

- **Secure Build**: Questa pratica si concentra sulla creazione di un processo di compilazione automatizzato e ripetibile, tenendo in considerazione anche lo stato di sicurezza sia degli applicativi in sviluppo sia delle dipendenze software da loro utilizzate.
- **Secure Deployment**: Questa pratica si concentra sull'aumentare la sicurezza delle distribuzioni software nell'ambiente di produzione e delle informazioni sensibili di supporto.
- **Defect Management**: Questa pratica si concentra sulla gestione dei difetti di sicurezza nel software e sulle metriche associate.

IMPLEMENTATION



Security Practices	Stream A	Stream B
Secure Build	Build Process	Software Dependencies
Secure Deployment	Deployment Process	Secret Management
Defect Management	Defect Tracking	Metrics & Feedback

SECURE BUILD

Stream A – Build Process

Un processo di «compilazione consistente» deve garantire che il software che si sta distribuendo sia prevedibile e direttamente collegato al codice sorgente. Inoltre, è possibile sfruttare il processo di compilazione del software per varie attività di sicurezza, come ad esempio scansioni di sicurezza statiche (SAST).

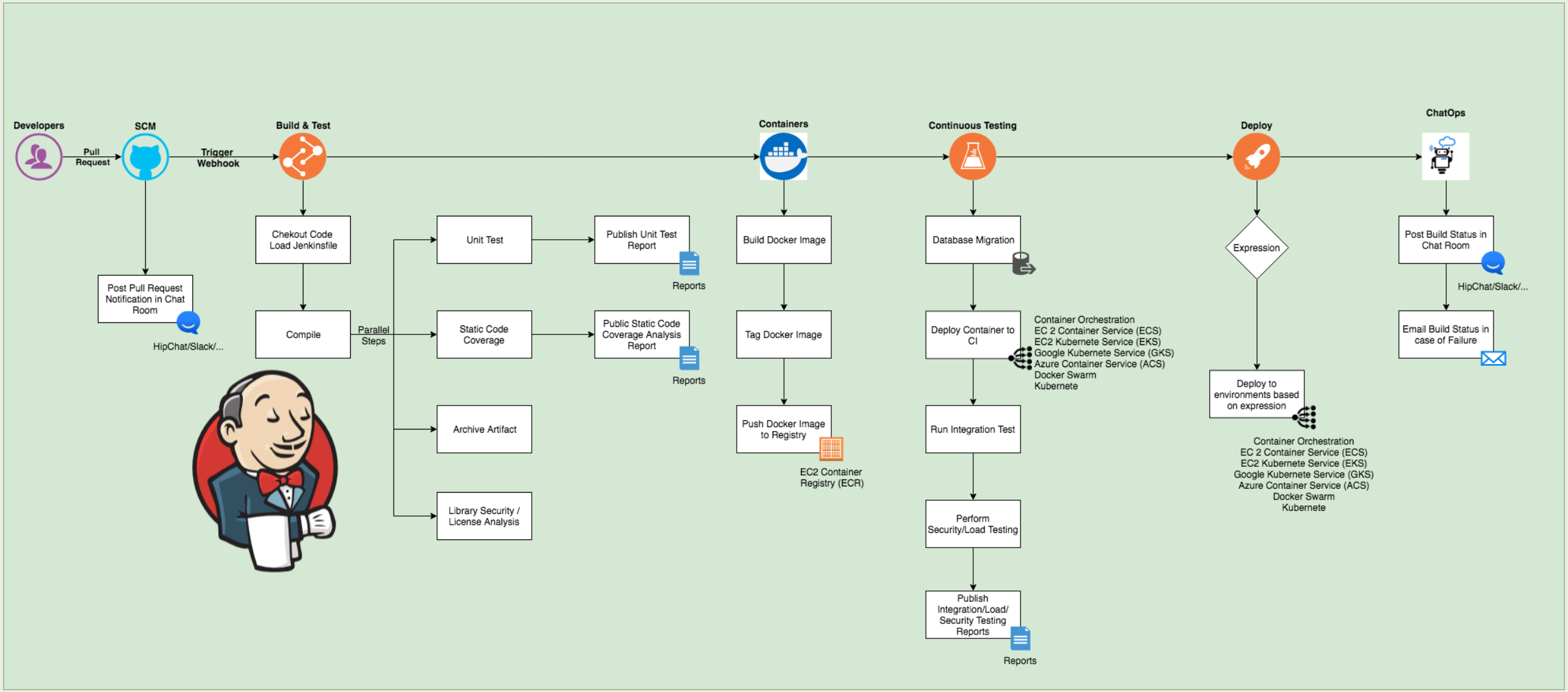
Stream B – Software Dependencies

Le librerie esterne sono una parte significativa del software moderno. Le attività in questo filone aiutano a creare una miglior visione delle librerie esterne e assicurano che la loro robustezza sia adeguata dal punto di vista della sicurezza.

SECURE BUILD

Stream A – Build Process	Stream B – Software Dependencies
Livello di maturità 1: Il processo di compilazione è ripetibile e consistente.	
Creare una definizione formale del processo di compilazione in modo che diventi consistente e ripetibile.	Creare almeno un elenco dei delle dipendenze utilizzate e analizzarle in modo opportunistico.
Livello di maturità 2: Il processo di compilazione è ottimizzato e completamente integrato nel flusso di sviluppo.	
Automatizzare la pipeline di compilazione e garantire la sicurezza degli strumenti utilizzati. Aggiungere, inoltre, controlli di sicurezza relativi al codice in sviluppo.	Valutare le dipendenze utilizzate e garantire una reazione tempestiva alle situazioni che rappresentano un rischio per le applicazioni sviluppate.
Livello di maturità 3: Il processo di compilazione aiuta a prevenire che difetti conosciuti entrino nell'ambiente di produzione.	
Definire controlli di sicurezza obbligatori nel processo di compilazione e garantire che la creazione di artefatti non conformi fallisca.	Analizzare le dipendenze utilizzate per individuare vulnerabilità in modo comparabile al proprio codice.

SECURE BUILD



CYBERLOOP

SECURE DEPLOYMENT

Uno degli ultimi passaggi nel fornire software sicuro è garantire che la sicurezza e l'integrità delle applicazioni sviluppate **non siano compromesse durante il rilascio**. La pratica di **Secure Deployment** (SD) si concentra su questo. A tal fine, il primo filone della pratica si concentra sulla rimozione degli errori manuali **automatizzando il processo di distribuzione** il più possibile e rendendo il suo successo condizionato ai risultati dei controlli integrati di verifica della sicurezza. Favorisce inoltre la Separazione dei Compiti rendendo responsabili del rilascio **persone adeguatamente formate** e non sviluppatori.

Il secondo filone va oltre la meccanica del rilascio e si concentra sulla **protezione della privacy e dell'integrità dei dati sensibili**, come password, token e altri segreti, necessari per il funzionamento delle applicazioni negli ambienti di produzione. Nella sua forma più semplice, i segreti di produzione appropriati vengono spostati dai repository e dai file di configurazione in cassette di sicurezza digitali gestite in modo adeguato. In forme più avanzate, i segreti vengono generati dinamicamente al momento del rilascio e i processi di routine rilevano e mitigano la presenza di eventuali segreti non protetti nell'ambiente.

SECURE DEPLOYMENT

Stream A – Deployment Process

Un processo di distribuzione ripetibile e coerente garantisce di distribuire solo artefatti software «corretti» in ambiente di produzione. Inoltre, prepara il terreno per ambienti di test rappresentativi prima della produzione.

Stream B – Secret Management

Poiché l'esecuzione sicura di qualsiasi sistema software richiede credenziali, questo filone garantisce una gestione adeguata di questi dati sensibili all'interno dell'ambiente dell'organizzazione.

SECURE DEPLOYMENT

Stream A – Deployment Process	Stream B – Secret Management
Livello di maturità 1: I processi di distribuzione sono completamente documentati.	
Formalizzare il processo di distribuzione e proteggere gli strumenti e i processi utilizzati.	Introdurre misure di protezione di base per limitare l'accesso ai segreti di produzione.
Livello di maturità 2: I processi di distribuzione includono tappe di verifica della sicurezza.	
Automatizzare il processo di distribuzione su tutte le fasi e introdurre test di verifica della sicurezza.	Fare injection dei segreti dinamicamente durante il processo di distribuzione da archivi sicuri e monitorare tutti gli accessi umani.
Livello di maturità 3: Il processo di distribuzione è completamente automatizzato e incorpora la verifica automatica di tutte le tappe critiche.	
Verificare automaticamente l'integrità di tutto il software distribuito, indipendentemente dal fatto che sia sviluppato internamente o esternamente.	Migliorare il ciclo di vita dei segreti generandoli regolarmente e garantendo un uso adeguato.

DEFECT MANAGEMENT

Stream A – Defect Tracking

Il tracciamento delle vulnerabilità gestisce la raccolta e il follow-up di tutti i potenziali problemi in una porzione di software, dalle falle architetturali ai problemi dovuti al codice fino alle vulnerabilità di runtime.

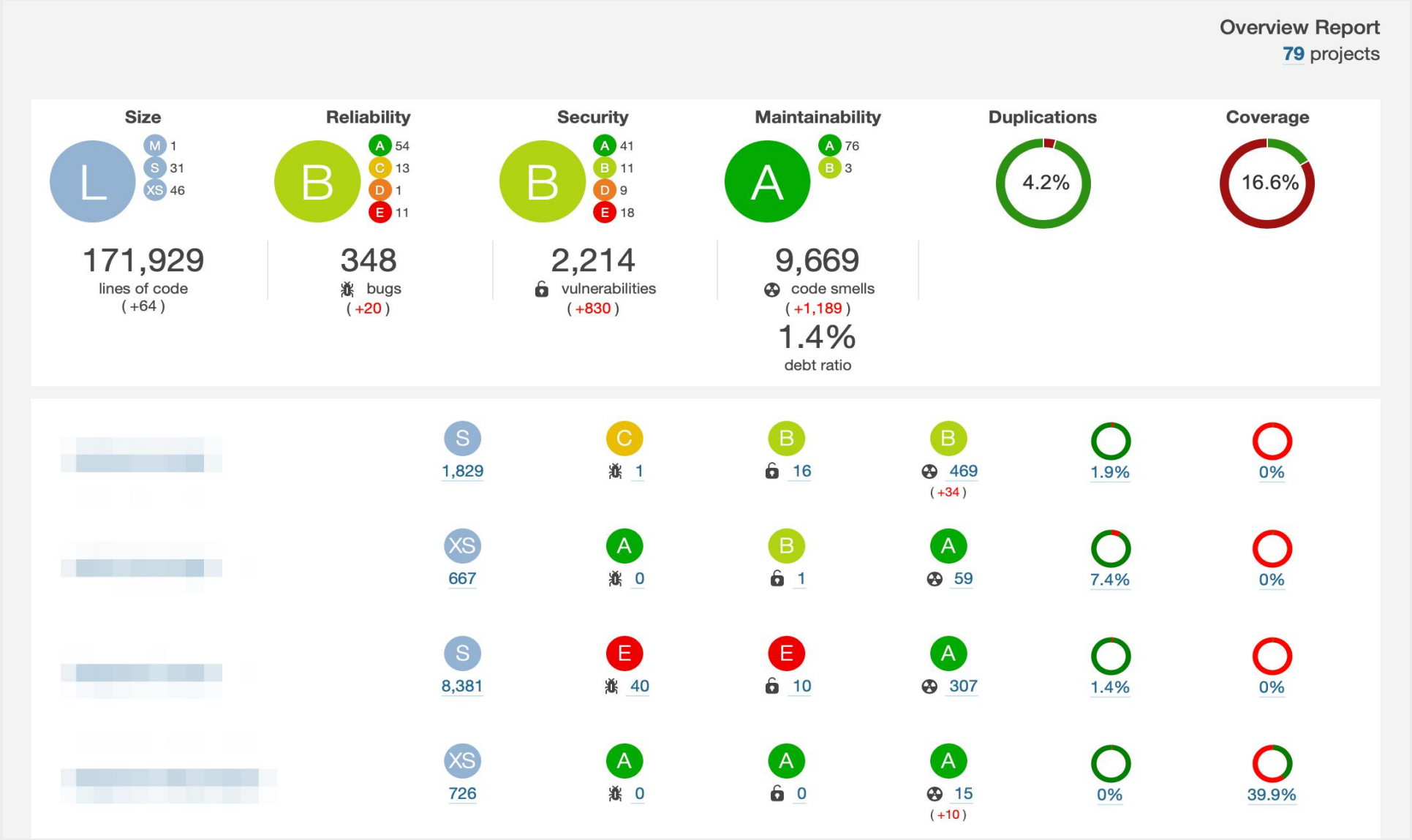
Stream B – Metrics & Feedback

Il tracciamento delle vulnerabilità può guidare il miglioramento delle attività di sicurezza all'interno dell'organizzazione attraverso metriche e feedback.

DEFECT MANAGEMENT

Stream A – Defect Tracking	Stream B – Metrics & Feedback
Livello di maturità 1: Tutte le vulnerabilità sono tracciate all'interno di ciascun progetto.	
Introdurre un tracciamento strutturato dei difetti di sicurezza e prendere decisioni informate basate su queste informazioni.	Rivedere regolarmente i difetti di sicurezza precedentemente registrati e trarre vantaggio da successi rapidi («quick win») basati su metriche di base facilmente risolvibili.
Livello di maturità 2: Il tracciamento delle vulnerabilità viene utilizzato per influenzare il processo di distribuzione.	
Valutare tutti i difetti di sicurezza su tutta l'organizzazione in modo coerente e definire SLA per specifiche classi di gravità.	Raccogliere metriche standardizzate di gestione dei difetti di sicurezza e utilizzarle anche per la prioritizzazione delle iniziative guidate centralmente.
Livello di maturità 3: Il tracciamento delle vulnerabilità attraverso più componenti software viene utilizzato per aiutare a ridurre l'insorgenza di nuove vulnerabilità.	
Imporre gli SLA definiti precedentemente e integrare il sistema di gestione delle vulnerabilità tra diversi strumenti.	Migliorare continuamente le metriche di gestione dei difetti di sicurezza e correlarle con altre fonti.

DEFECT MANAGEMENT



CYBERLOOP

Le fasi del processo di SSDLC

Verification

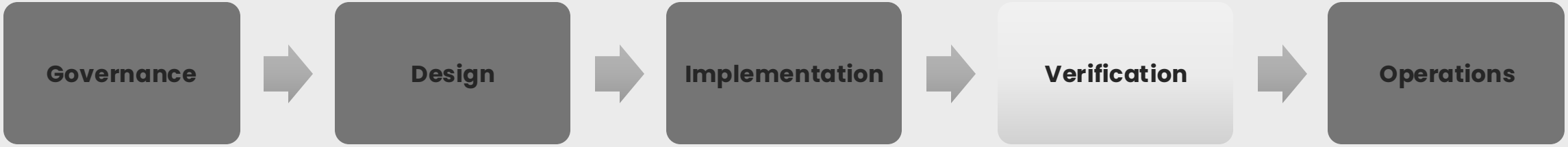
VERIFICATION

La **verifica** (verification) si concentra sui processi e sulle attività legate al controllo e alla verifica degli artefatti prodotti durante lo sviluppo del software. Questo include tipicamente lavori di assicurazione della qualità come i test, ma può anche includere altre attività di revisione e valutazione.

Si suddivide in:

- **Architecture Assessment**: Questa pratica si concentra sulla convalida della sicurezza e della conformità dell'architettura del software e dell'infrastruttura di supporto.
- **Testing guidato dai Requisiti:** Questa pratica si concentra sull'utilizzo sia di test di sicurezza positivi (verifica dei controlli) che negativi (test di abuso) basati sui requisiti (storie degli utenti).
- **Testing di Sicurezza:** Questa pratica si concentra sulla rilevazione e risoluzione di problemi di base di sicurezza attraverso l'automazione, consentendo ai test manuali di concentrarsi su vettori di attacco più complessi.

VERIFICATION



Security Practices	Stream A	Stream B
Architecture Assessment	Architecture Validation	Architecture Mitigation
Requirements-driven Testing	Control Verification	Misuse/Abuse Testing
Security Testing	Scalable Baseline	Deep Understanding

ARCHITECTURE ASSESSMENT

Stream A – Architecture Validation

Questo stream ha lo scopo di verificare la sicurezza del software e dell'architettura di supporto, identificando i componenti sia dell'architettura dell'applicazione sia dell'infrastruttura di supporto, e verificando il loro soddisfacimento degli obiettivi e dei requisiti di sicurezza.

Stream B – Architecture Mitigation

Questo stream si concentra nel garantire che tutte le minacce identificate durante la Valutazione delle Minacce siano adeguatamente mitigate e che le architetture di riferimento esistenti siano aggiornate per affrontare eventuali minacce non gestite

ARCHITECTURE ASSESSMENT

Stream A – Architecture Validation	Stream B – Architecture Mitigation
Livello di maturità 1: Revisionare l'architettura per assicurarsi che le mitigazioni di base siano in atto per i rischi più comuni.	
Identificare i componenti dell'architettura dell'applicazione e dell'infrastruttura e rivedere il soddisfacimento dei requisiti di sicurezza di base.	Revisione ad hoc dell'architettura per identificare minacce di sicurezza non mitigate.
Livello di maturità 2: Rivedere la completa fornitura di meccanismi di sicurezza nell'architettura.	
Validare i meccanismi di sicurezza dell'architettura.	Analizzare l'architettura rispetto alle minacce conosciute.
Livello di maturità 3: Rivedere l'efficacia dell'architettura e fornire feedback per migliorare l'architettura della sicurezza.	
Revisione dell'efficacia dei componenti di sicurezza all’interno dell'architettura.	Incorporare i risultati della revisione dell'architettura nei principi e nei modelli di progettazione aziendali, nelle soluzioni di sicurezza e nelle architetture di riferimento.

REQUIREMENTS-DRIVEN TESTING

L'obiettivo di questa fase è quello di **garantire che i controlli** di sicurezza implementati funzionino come previsto e **soddisfino i requisiti di sicurezza** dichiarati del progetto. Ciò avviene costruendo incrementalmente un insieme di test di sicurezza e di non regressione, ed eseguendoli regolarmente.

REQUIREMENTS-DRIVEN TESTING

Stream A – Control Verification

Convalida che i controlli di sicurezza e i requisiti siano soddisfatti attraverso test appositi, e impedisce l'introduzione di bug nelle versioni successive attraverso test di non regressione.

Stream B – Misuse/Abuse Testing

Sfrutta diverse tecniche come ad esempio il fuzzing, con l’obiettivo di identificare una qualsiasi funzionalità o risorsa nel software che può essere sfruttata per individuare debolezze nelle funzionalità di un'applicazione.

REQUIREMENTS-DRIVEN TESTING

Stream A – Control Verification	Stream B – Misuse/Abuse Testing
Livello di maturità 1: Identificare opportunisticamente vulnerabilità di base e altri problemi di sicurezza.	
Effettuare test per i controlli di sicurezza del software.	Effettuare test di fuzzing per la sicurezza.
Livello di maturità 2: Effettuare una revisione dell'implementazione per identificare rischi specifici dell'applicazione rispetto ai requisiti di sicurezza.	
Derivare casi di test dai requisiti di sicurezza conosciuti.	Creare e testare casi di abuso e test di vulnerabilità specifici per la logica di business dell'applicativo.
Livello di maturità 3: Mantenere il livello di sicurezza anche a seguito di correzioni di bug, modifiche o durante la manutenzione.	
Eseguire il testing di non regressione (coadiuvati da unit test).	Testare l'applicativo anche per attacchi di denial of service.

REQUIREMENTS-DRIVEN TESTING

OWASP ZAP Vulnerability Report

Report Name:			
Prepared For:	X Corp	Prepared By:	h4x0r
Scan Date:	Tue 09 Jun 2020 01:59:25 PM UTC	Scan Ver:	N/A
Report Date:	Tue 09 Jun 2020 01:59:25 PM UTC	Report Ver:	N/A
Description:	Home page vulnerability report of the Example project.		

Table of Contents

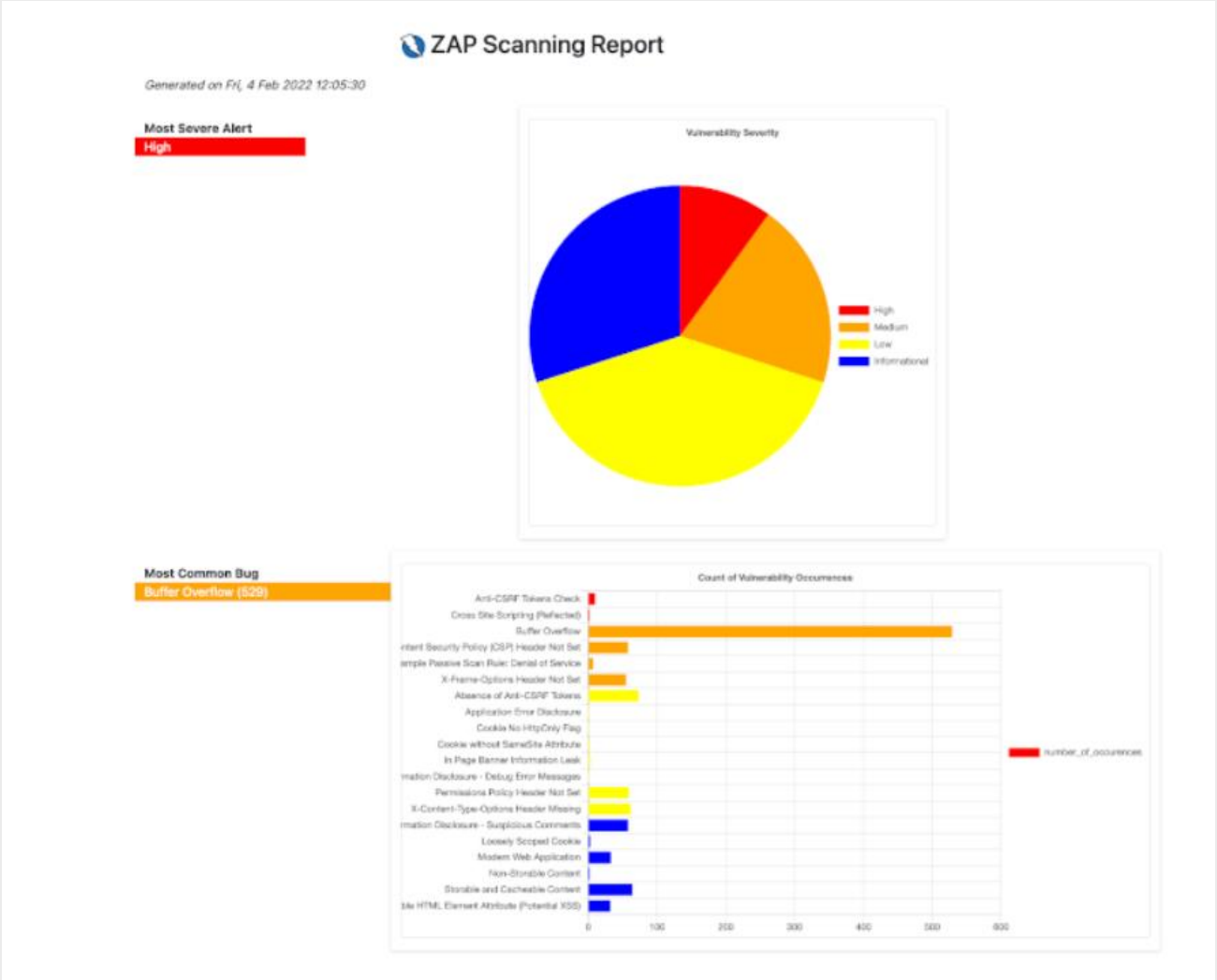
Site: http://example.com

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	0
Informational	0

Alert Details

Medium	X-Frame-Options Header Not Set	Top
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.	



CYBERLOOP

SECURITY TESTING

La pratica del Security Testing è necessaria in quanto, sebbene il testing di sicurezza automatizzato sia veloce e scalabile, per numerose applicazioni, il **testing approfondito** basato su una buona conoscenza di un'applicazione e della sua logica aziendale (fondamentale all'interno del SSDLC) è spesso possibile solo tramite testing manuale, condotto da un esperto.

A differenza della fase precedente, che si concentra sulla verifica che le applicazioni implementino correttamente i loro requisiti di sicurezza, l'obiettivo di questa pratica è scoprire **vulnerabilità tecniche** dell'applicazione e renderle visibili alla direzione e agli stakeholder aziendali, indipendentemente dai requisiti di sicurezza concordati.

SECURITY TESTING

Stream A – Scalable Baseline

Si concentra sull’utilizzo di strumenti di test automatizzati specifici dell'applicazione, i quali integrano la validazione della sicurezza nel processo di compilazione e distribuzione. L'obiettivo di questo stream è favorire la copertura (un ampio spettro di applicazioni) rispetto alla profondità dei test.

Stream B – Deep Understanding

Si concentra sull'esecuzione di test di sicurezza manuali di componenti ad alto rischio, utilizzando vettori di attacco complessi con l'obiettivo di rendere il testing avanzato parte integrante del processo di sviluppo. L'obiettivo di questo stream è favorire la profondità dei test (rigore nei test) rispetto alla copertura dei test (il portafoglio di applicazioni).

SECURITY TESTING

Stream A – Scalable baseline	Stream B – Deep Understanding
Livello di maturità 1: Non è presente automazione, e le analisi svolte sono limitate, così come le correzioni delle vulnerabilità riscontrate.	
I test di sicurezza vengono eseguiti in modo manuale e non coerente. Potrebbero essere utilizzati alcuni strumenti, ma non esiste un processo formale.	Le vulnerabilità vengono identificate e segnalate, ma l'analisi è limitata. Le correzioni sono reattive e non sistematiche.
Livello di maturità 2: I test sono automatizzati e integrati nelle singole pipeline di sviluppo. Le analisi sono più approfondite, seppur limitate ai singoli software.	
I test di sicurezza sono standardizzati e integrati nel ciclo di vita dello sviluppo. Gli strumenti automatici vengono utilizzati in modo coerente e la copertura dei test è monitorata.	Viene eseguita un'analisi delle cause profonde per i problemi significativi. Le lezioni apprese vengono condivise e utilizzate per migliorare i processi di sviluppo e test.
Livello di maturità 3: La strategia di testing è condivisa a livello aziendale ed i risultati sono utilizzati in modo proattivo per la prevenzione.	
Una strategia di test scalabile e basata sul rischio è implementata in tutti i progetti. I test sono automatizzati, ripetibili e soggetti a miglioramento continuo.	Si raggiunge una comprensione profonda e sistemica delle vulnerabilità. Le intuizioni ottenute vengono utilizzate in modo proattivo per prevenire problemi e guidare le decisioni architetturali e progettuali.

SECURITY TESTING

Il VAPT, acronimo di Vulnerability Assessment and Penetration Testing, è una metodologia di sicurezza informatica che combina due approcci distinti per valutare la robustezza di un sistema o di una rete.

Questi due approcci, distinti, sono:

- **Vulnerability Assessment**: è dedicato all'identificazione, alla quantificazione e alla classificazione delle vulnerabilità presenti nel sistema. Si tratta di un esame completo che utilizza strumenti automatizzati e tecniche manuali per rilevare possibili punti deboli o, in generale, falle di sicurezza.
- **Penetration Testing**: attività in cui esperti in sicurezza informatica, anche noti come «ethical hackers», tentano di sfruttare le vulnerabilità identificate per infiltrarsi all'interno del sistema. L'obiettivo è quello di simulare un attacco reale per verificare quanto siano efficaci le misure di sicurezza in atto.

In sintesi, il VAPT è un processo completo che identifica le potenziali vulnerabilità e che valuta la capacità del sistema di resistere agli attacchi informatici, fornendo così un quadro dettagliato e approfondito dello stato di sicurezza del sistema stesso.

Le fasi del processo di SSDLC

Operations

OPERATIONS

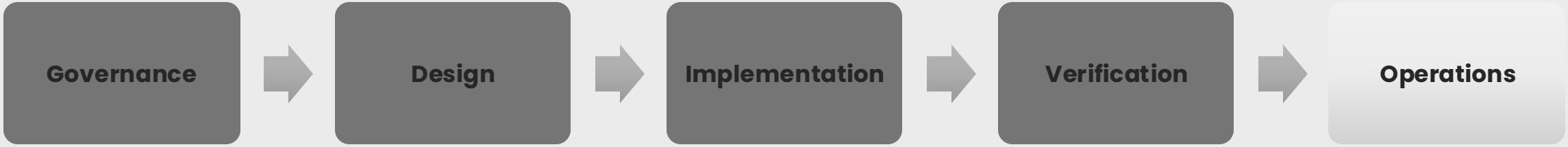
La fase di «operations» comprende quelle attività necessarie per garantire che la riservatezza, l'integrità e la disponibilità dei dati siano mantenute per tutta la durata operativa di un'applicazione e dei dati associati ad essa.

Un aumento di maturità riguardo a questa fase fornisce una maggiore garanzia che l'organizzazione sia resiliente di fronte a interruzioni operative e pronta a rispondere ai cambiamenti nel panorama operativo.

Si suddivide in:

- **Incident management**: Ha l'obiettivo di migliorare la capacità dell'organizzazione di individuare e rispondere agli incidenti di sicurezza.
- **Enviroment management**: Descrive le attività proattive svolte per migliorare e mantenere la sicurezza degli ambienti in cui operano le applicazioni dell'organizzazione.
- **Operational management:** Si concentra sulle attività di supporto operativo necessarie per mantenere la sicurezza durante tutto il ciclo di vita del prodotto.

OPERATIONS



Security Practices	Stream A	Stream B
Incident Management	Incident Detection	Incident Response
Environment Management	Configuration Hardening	Patching & Updating
Operational Management	Data Protection	Legacy Management

INCIDENT MANAGEMENT

Stream A – Incident Detection

Il rilevamento degli incidenti si riferisce al processo di determinare se un evento identificato sia effettivamente un incidente di sicurezza o meno.

Le attività in questo stream si concentrano sulla capacità dell'organizzazione di identificare gli incidenti di sicurezza quando si verificano e di avviare le appropriate attività di risposta agli incidenti.

Stream B - Risposta agli Incidenti

La risposta agli Incidenti inizia nel momento in cui si riconosce e si verifica un incidente di sicurezza. L'obiettivo è agire in modo coordinato ed efficiente in modo che ulteriori danni siano limitati il più possibile.

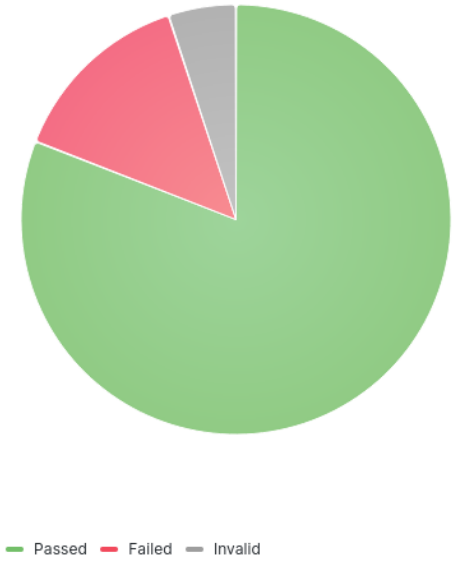
Le attività in questo stream si concentrano sulla capacità dell'organizzazione di rispondere in modo appropriato ed efficace agli incidenti di sicurezza segnalati.

INCIDENT MANAGEMENT

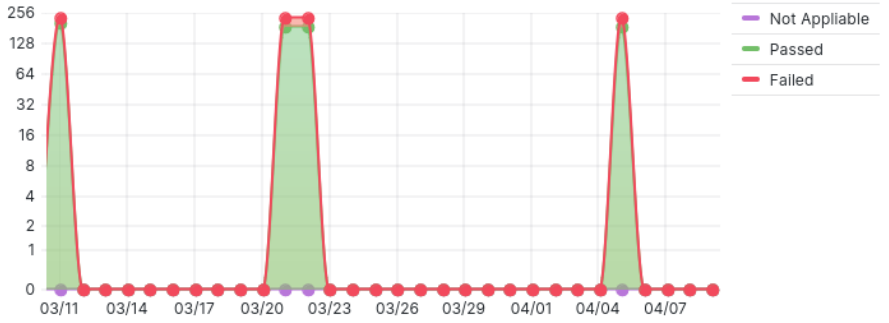
Stream A – Incident Detection	Stream B – Incident Response
Livello di maturità 1: Rilevamento e gestione degli incidenti con il massimo impegno ma senza un processo formale.	
Utilizzare i dati di log disponibili per effettuare il rilevamento con il massimo impegno di possibili incidenti di sicurezza.	Identificare ruoli e responsabilità per la risposta agli incidenti.
Livello di maturità 2: Processo formale di gestione degli incidenti istituito.	
Seguire un processo stabilito e ben documentato per il rilevamento degli incidenti, con enfasi sull'valutazione automatizzata dei log.	Stabilire un processo formale di risposta agli incidenti e garantire che il personale sia adeguatamente formato per svolgere i propri ruoli.
Livello di maturità 3: Gestione degli incidenti matura.	
Utilizzare un processo gestito in modo proattivo per il rilevamento degli incidenti.	Impiegare un team dedicato e ben addestrato per la risposta agli incidenti.

INCIDENT MANAGEMENT

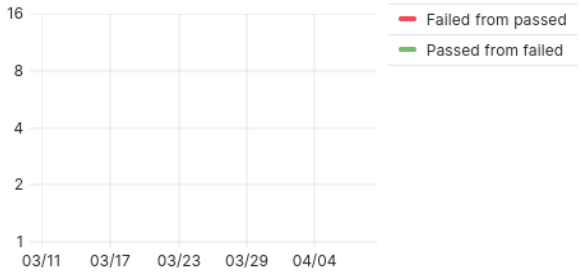
Compliance distribution



Compliance trend

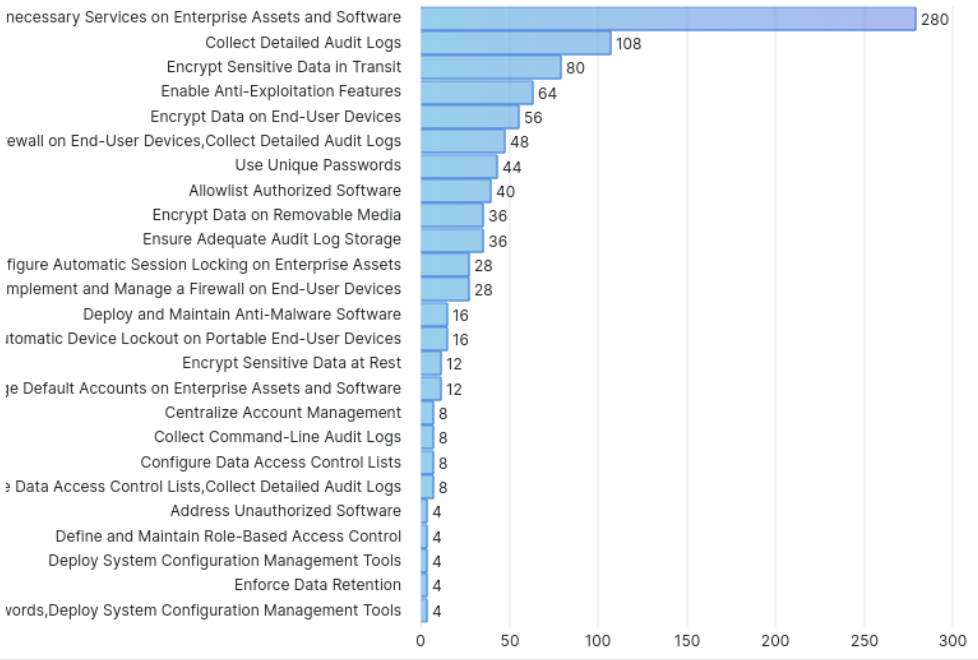


Checks status changes



No status change to failure in selected range

Compliance checks by category



Compliance summary

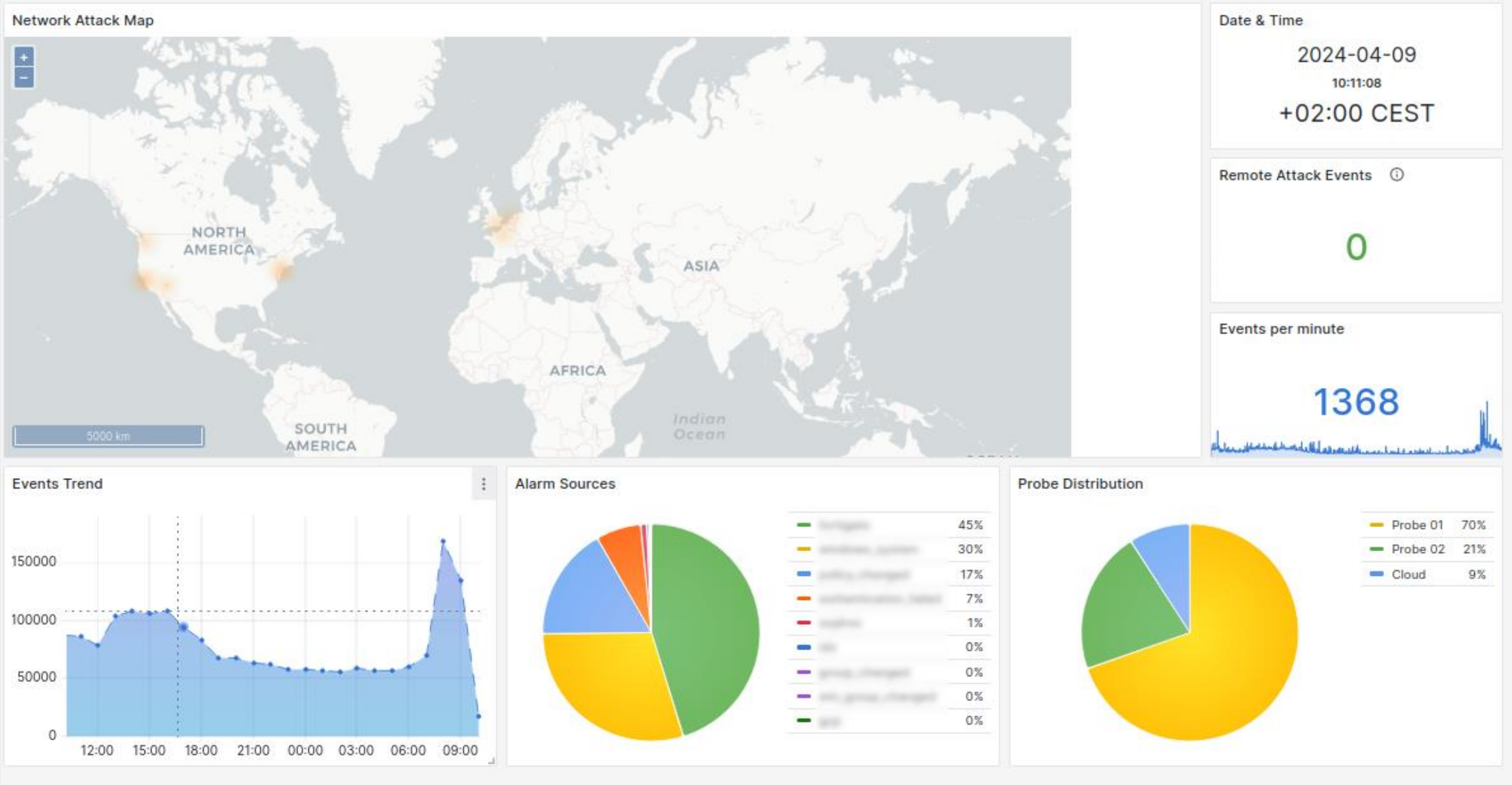
Endpoint	Failed check	Passed check	Invalid c
[redacted]	40	187	
[redacted]	39	188	
[redacted]	21	206	

Compliance checks failure details

Description	Rationale	Remediation	CIS ID	Endpoint with fail
Ensure 'Interactive logon: Machine account loc...	If a machine is lost or stolen, or if an insider th...	To establish the recommended configuration v...	2.3.7.3	3
Ensure 'Configure Windows Defender SmartSc...	Windows Defender SmartScreen helps keep P...	To establish the recommended configuration v...	18.10.76.2.1	3
Ensure 'Prevent installation of devices using d...	A BitLocker-protected computer may be vulne...	To establish the recommended configuration v...	18.9.7.1.6	3

CYBERLOOP

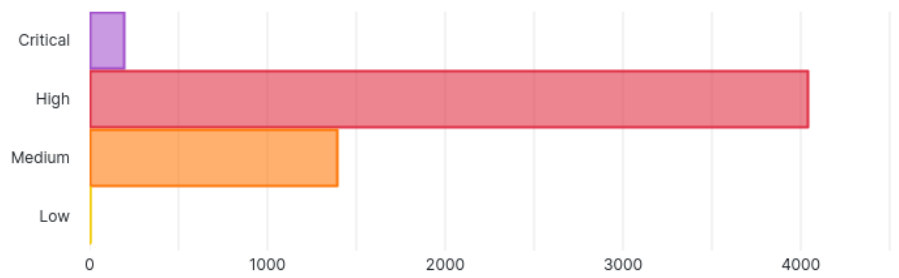
INCIDENT MANAGEMENT



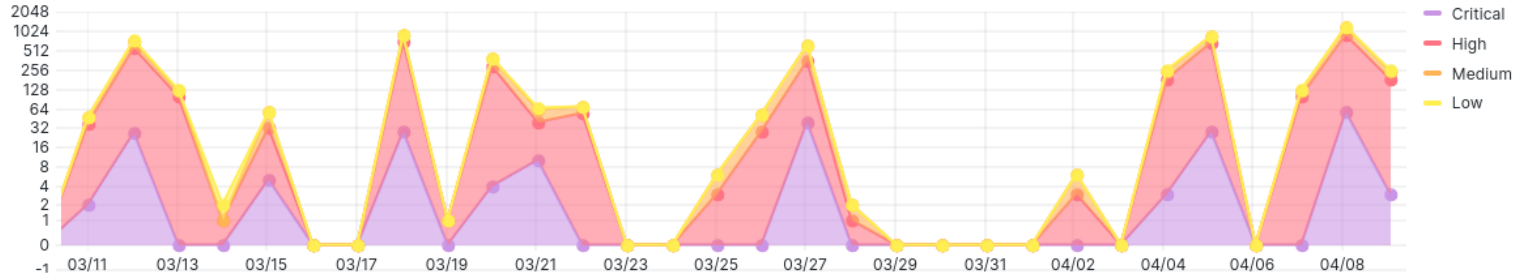
CYBERLOOP

INCIDENT MANAGEMENT

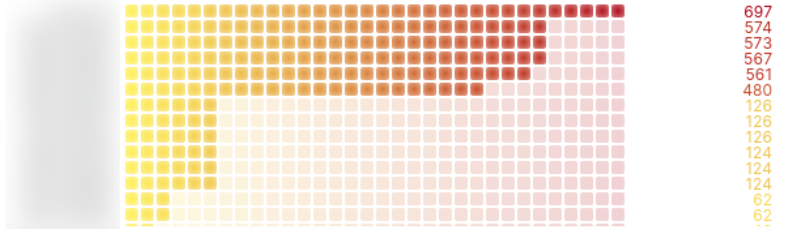
Vulnerability by Severity



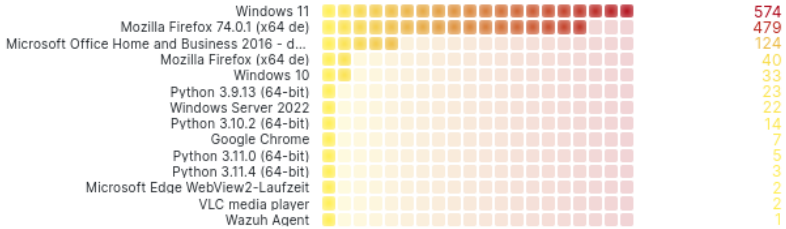
Vulnerability Trend



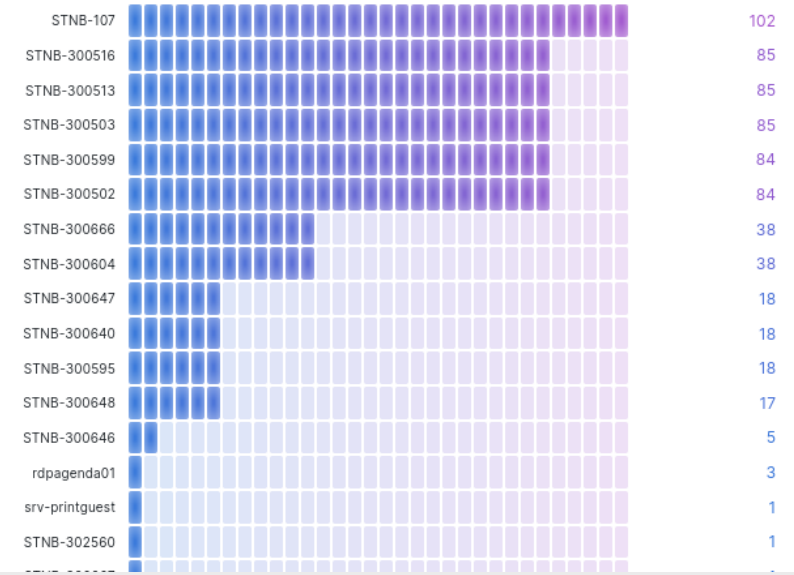
CVE detected by Hosts



CVE detected by Software



Missing Microsoft Security KB



Vulnerability Mitigation by Hosts

Hosts	Impacted Software	Mitigation Impact
[redacted]	1	
[redacted]	1	
[redacted]	1	
[redacted]	1	
[redacted]	2	

Vulnerability Mitigation by Software

Software	Impacted Hosts	Mitigation Impact
Mozilla Firefox 74.0.1 (x64...	1	
Windows 11	11	
Microsoft Office Home an...	8	
Python 3.9.13 (64-bit)	1	
Python 3.10.2 (64-bit)	1	

CYBERLOOP

ENVIROMENT MANAGEMENT

Stream A – Configuration Hardening

Le attività in questo stream si concentrano sulla gestione da parte dell'organizzazione delle configurazioni legate alla sicurezza in tutti gli elementi dello stack tecnologico.

L'accento è posto su quegli elementi (ad esempio, sistemi operativi, container, framework, servizi, dispositivi e librerie) ottenuti da terze parti, poiché la loro architettura e design non sono sotto il controllo dell'organizzazione.

Stream B – Patching & Updating

Le attività in questo stream si concentrano sulla gestione da parte dell'organizzazione di patch e aggiornamenti per tutti gli elementi dello stack tecnologico. Per il software sviluppato dall'organizzazione, queste attività riguardano la consegna di patch e aggiornamenti ai clienti, nonché la loro applicazione alle soluzioni gestite dall'organizzazione (ad esempio, software come servizio). Per gli elementi di terze parti, queste attività riguardano l'applicazione tempestiva di aggiornamenti e patch ricevuti.

ENVIROMENT MANAGEMENT

Stream A	Stream B
Livello di maturità 1: patch e hardening a best-effort.	
Effettuare hardening delle configurazioni con il massimo impegno, basato sulle informazioni prontamente disponibili.	Effettuare il patching dei componenti di sistema e delle applicazioni con il massimo impegno disponibile.
Livello di maturità 2: Processo formale con basi stabilite.	
Effettuare un hardening delle configurazioni, coerentemente con le linee guida e con gli standard stabiliti.	Effettuare regolarmente il patching dei componenti di sistema e delle applicazioni, su tutto lo stack. Garantire la consegna tempestiva delle patch ai clienti.
Livello di maturità 3: Conformità con un processo di miglioramento continuo, applicato con rigore.	
Monitorare attivamente le configurazioni per individuare eventuali non conformità e gestire le occorrenze rilevate come difetti di sicurezza.	Monitorare attivamente lo stato degli aggiornamenti e gestire i patch mancanti come difetti di sicurezza. Ottenere proattivamente informazioni sulle vulnerabilità e sugli aggiornamenti per i componenti.

OPERATIONAL MANAGEMENT

Stream A – Data Protection

Le attività in questo stream si concentrano sul garantire che l'organizzazione protegga adeguatamente i dati in tutti gli aspetti della loro creazione, gestione, archiviazione e elaborazione.

Stream B – Legacy Management

Le attività in questo stream si concentrano sull'identificazione, gestione e tracciamento di sistemi, applicazioni, dipendenze e servizi che non sono più utilizzati, hanno raggiunto la fine del loro ciclo di vita o non sono più sviluppati o supportati attivamente.

La rimozione di sistemi e servizi inutilizzati migliora la gestibilità dell'ambiente e riduce la superficie di attacco dell'organizzazione, consentendo risparmi diretti e indiretti (ad esempio, riduzione del numero di licenze, riduzione del volume di logging o riduzione dello sforzo degli analisti).

OPERATIONAL MANAGEMENT

Stream A – Data Protection	Stream B – Legacy Management
Livello di maturità 1: Pratiche Fondamentali.	
Implementare pratiche di protezione dei dati di base.	De-commissionare le applicazioni e i servizi non utilizzati. Gestire gli aggiornamenti/migrazioni dei clienti, seppur individualmente.
Livello di maturità 2: Processi Gestiti e Reattivi	
Sviluppare un catalogo dei dati e stabilire una politica di protezione dei dati sulla base del loro livello di riservatezza.	Sviluppare processi di dismissione ripetibili per sistemi/servizi non utilizzati e per la migrazione dalle dipendenze legacy. Gestire le roadmap di migrazione legacy per i clienti.
Livello di maturità 3: Monitoraggio Attivo e Risposta repentina	
Automatizzare il rilevamento della non conformità alle politiche e verificare periodicamente la conformità. Rivedere e aggiornare regolarmente il catalogo dei dati e la politica di protezione dei dati.	Gestire proattivamente le roadmap di migrazione, sia per le dipendenze non supportate in fase di fine vita, sia per le versioni legacy del software fornito.

LABORATORIO DI SICUREZZA DEI SISTEMI E PRIVACY
CICLO DI SVILUPPO SICURO DEL CODICE

Marco Canducci • V1 • 09/05/2025

https://cyberloop.it
info@cyberloop.it
P.IVA: IT04502170402

CYBERLOOP

QUALITÀ ISO 9001
SICUREZZA ISO 27001

CLASSIFICAZIONE

TLP:GREEN