LiveCampus

Apprentissage connecté

Sommaire

LiveCampus

Apprentissage connecté

Sécurité dans les phases de conception : Waterfall vs Agile vs DevSecOps

La sécurité est un enjeu crucial dans tout projet informatique. Toutefois, son intégration varie fortement selon la méthode de gestion de projet utilisée. Voici une analyse comparative de la sécurité dans les méthodologies Waterfall (cycle en V), Agile, et DevSecOps.

LiveCampus

Apprentissage connecté

Waterfall (Cycle en V) – Sécurité dans la phase de conception

Caractéristique	Description
Méthodologie	Séquentielle : analyse → conception → développement → tests → livraison
Sécurité intégrée	À la fin du cycle, lors de la phase de test
Responsabilité de la sécurité	Repos sur une équipe spécialisée (souvent externe au développement)
Modélisation des menaces	Ponctuelle, parfois absente
Tests de sécurité	Réalisés après le développement (ex : audit, pentest)
Réactivité face aux failles	Lente, coûteuse (retour en arrière complexe)
Outils de sécurité	Manuels, peu intégrés
Culture de sécurité	Centralisée, cloisonnée
Adaptabilité	Faible : modifications complexes et longues
Formation à la sécurité	Réservée aux experts

LiveCampus

Apprentissage connecté

Agile – Sécurité dans la phase de conception

Caractéristique	Description
Méthodologie	Itérative et incrémentale : cycles courts (sprints)
Sécurité intégrée	Parfois prévue dès le backlog, sinon intégrée en sprint spécifique
Responsabilité de la sécurité	Partagée, mais souvent floue ou négligée
Modélisation des menaces	Variable, dépend de l'équipe et du contexte
Tests de sécurité	Manuels ou semi-automatisés, planifiés selon les priorités
Réactivité face aux failles	Moyenne, selon le rythme des sprints
Outils de sécurité	Peu intégrés, souvent ponctuels
Culture de sécurité	En construction ou portée par certains membres sensibles au sujet
Adaptabilité	Moyenne à bonne : intégration possible de nouvelles exigences en sprint
Formation à la sécurité	Occasionnelle, dépend de l'organisation

LiveCampus

Apprentissage connecté

DevSecOps – Sécurité dans la phase de conception

Caractéristique	Description
Méthodologie	Intégrée et continue : Dev + Sec + Ops dans un pipeline CI/CD
Sécurité intégrée	Dès la conception, en continu jusqu'au déploiement
Responsabilité de la sécurité	Partagée entre développeurs, ops et experts sécurité
Modélisation des menaces	Systématique et outillée (ex : Threat Modeling intégré dans les pipelines)
Tests de sécurité	Automatisés : SAST, DAST, scans de dépendances, IAST
Réactivité face aux failles	Très rapide grâce à l'automatisation
Outils de sécurité	Fortement intégrés (DevSecOps Toolchain)
Culture de sécurité	Transversale, participative, intégrée à chaque étape
Adaptabilité	Très forte : feedbacks et corrections rapides
Formation à la sécurité	Continue, pour tous les membres de l'équipe

LiveCampus

Apprentissage connecté

Comparatif Sécurité : Waterfall vs Agile vs DevSecOps

Critère	Waterfall	Agile	DevSecOps
Méthodologie	Séquentielle (cycle en V)	Itérative et incrémentale	Continue, collaborative et automatisée
Moment d'intégration de la sécurité	Tardif, principalement en phase de test	Variable, souvent après plusieurs sprints	Dès le début, intégrée dans toutes les phases
Responsabilité de la sécurité	Équipe sécurité dédiée	Parfois absente ou limitée à certains rôles	Partagée entre développement, opérations et sécurité
Outils de sécurité	Outils manuels, audits ponctuels	Peu utilisés ou ajoutés en sprint spécifique	Automatisation avec CI/CD, scans, tests SAST/DAST
Modélisation des menaces	Ponctuelle, souvent formelle	Occasionnelle, selon les priorités des sprints	Continue, systématique avec outils dédiés
Tests de sécurité	En fin de cycle, peu fréquents	Manuels ou semi-automatisés selon les cycles	Automatisés et intégrés dans les pipelines DevOps
Réactivité aux failles	Lente, corrections coûteuses	Moyenne, dépend de la planification des sprints	Rapide, intégrée au cycle de livraison
Culture sécurité	Cloisonnée, externalisée	Fragmentée ou en cours d'intégration	Culture Dev+Sec+Ops, sécurité comme valeur commune
Formation à la sécurité	Réservée aux experts	Parfois proposée, selon les équipes	Généralisée à toute l'équipe
Adaptation aux menaces évolutives	Faible, architecture figée	Moyenne, possibilité d'ajustement par sprint	Forte, architecture et sécurité adaptables en temps réel
Investissement initial	Faible, mais coûts élevés si problèmes tardifs	Moyen, dépend de l'organisation	Plus élevé au départ, mais réduction globale des coûts à long terme
Environnement idéal	Projets à exigences stables et réglementées	Projets souples, en évolution rapide	Environnements critiques, cloud, DevOps, CI/CD

LiveCampus

Apprentissage connecté

Établissement des exigences de sécurité

Définition

L'établissement des exigences de sécurité consiste à **identifier, formaliser et documenter** les besoins de protection d'un système ou d'une application **dès la phase de conception**. Ces exigences visent à :

LiveCampus

Apprentissage connecté

- Protéger les données sensibles
- Prévenir les menaces
- Respecter les règlementations
- Garantir confidentialité / intégrité et disponibilité
- Anticiper mes futures évolutions de sécurité

LiveCampus

Apprentissage connecté

Les exigences pour Waterfall

Caractéristique	Détails
Moment de définition	En tout début de projet (phase d'analyse)
Méthode	Documentation formelle et exhaustive des exigences
Acteurs impliqués	Analystes métier, architectes, équipe sécurité
Nature des exigences	Techniques et réglementaires, souvent orientées infrastructure
Avantage	Vision claire dès le début
Limite	Difficile à adapter aux changements ; modifications coûteuses

Exemple :"Toutes les données utilisateurs doivent être chiffrées en base de données avec AES-256."

LiveCampus

Apprentissage connecté

Les exigences pour Agile

Caractéristique	Détails
Moment de définition	En continu, par incréments (durant les sprints)
Méthode	Exigences intégrées dans le backlog (user stories, security stories)
Acteurs impliqués	PO, développeurs, parfois expert sécurité
Nature des exigences	Contextuelles, selon les priorités du sprint
Avantage	Adaptabilité, intégration progressive
Limite	Risque d'oublier certaines exigences globales

Exemple (sous forme de User Story de sécurité) :"En tant qu'utilisateur, je veux que ma session expire après 10 minutes d'inactivité pour éviter un accès non autorisé."

LiveCampus

Apprentissage connecté

Les exigences pour Devsecops

Caractéristique	Détails
Moment de définition	Continu et intégré dans le pipeline DevOps
Méthode	Automatisation + exigences codifiées (Security as Code)
Acteurs impliqués	Dev + Ops + Sec (équipe DevSecOps)
Nature des exigences	Techniques, dynamiques, évolutives
Avantage	Sécurité proactive, ajustable en temps réel
Limite	Nécessite des compétences croisées et des outils adaptés

Exemple :Une règle dans le pipeline CI/CD vérifie automatiquement que toutes les dépendances sont exemptes de vulnérabilités connues (via un scanner type Trivy ou Snyk).

LiveCampus

Apprentissage connecté

Bonnes pratiques communes

Pratique	Description
Modélisation des menaces	Identifier les risques potentiels dès la conception (ex. STRIDE, LINDDUN)
Classification des données	Déterminer les niveaux de sensibilité et adapter les exigences de sécurité
Implication multi-acteurs	Intégrer les parties prenantes (métier, sécurité, IT, juridique)
Conformité réglementaire	Aligner les exigences avec les normes en vigueur
Traçabilité des exigences	Lier chaque exigence à une fonctionnalité ou un test pour suivi dans le temps

LiveCampus

Apprentissage connecté

Qu’est-ce que la modélisation des menaces ?

Élément	Description
Définition	Processus d’analyse permettant d’identifier, hiérarchiser et documenter les menaces potentielles contre un système
But	Protéger les actifs, orienter les choix techniques, anticiper les attaques
Outils fréquents	STRIDE (Microsoft), LINDDUN (vie privée), DREAD, Attack Trees, PASTA, MITRE ATT&CK
Acteurs impliqués	Développeurs, architectes, RSSI, DPO, DevSecOps
Moment optimal	Dès la phase de conception, puis à chaque changement d’architecture

LiveCampus

Apprentissage connecté

Qu’est-ce qu’une architecture sécurisée ?

Élément	Description
Définition	Structure technique conçue pour **réduire la surface d’attaque** et **contrôler les accès** aux composants sensibles
Principes clés	Segmentation, chiffrement, gestion des identités, principes du moindre privilège
Outils associés	Diagrammes d’architecture, firewall, IAM, MFA, proxy, bastions, zero trust
But	Assurer **confidentialité, intégrité et disponibilité (CIA)**
Lien avec la modélisation	Les résultats de la modélisation des menaces alimentent directement l’architecture sécurisée

LiveCampus

Apprentissage connecté

Outils et techniques utiles (toutes approches)

Outil / Technique	Utilité
STRIDE	Identifier les types de menaces : Spoofing, Tampering, Repudiation, etc.
Diagrammes DFD	Visualiser les flux de données entre composants
Zero Trust	Ne jamais faire confiance par défaut, toujours vérifier
Architecture en couches	Isoler les composants critiques (front, back, base de données, réseau)
Automatisation	Intégrer les vérifications de sécurité dans le pipeline CI/CD
Checklists OWASP	OWASP ASVS (Application Security Verification Standard) pour les exigences

LiveCampus

Apprentissage connecté

Qu’est-ce qu’un DFD (Data Flow Diagram) ?

Le DFD (ou Diagramme de flux de données) est un outil visuel utilisé pour représenter les flux d’informations entre les acteurs, les processus, les stockages de données et les systèmes externes.

Il est particulièrement utile pour :Comprendre comment les données circulent dans un système.

Identifier les zones sensibles pour la modélisation des menaces (STRIDE, etc.).Concevoir une architecture sécurisée autour des points critiques.

LiveCampus

Apprentissage connecté

Symboles principaux d'un DFD

Symbole	Nom	Description
	**Processus**	Traitement effectué sur les données (ex : authentification)
	**Stockage de données**	Lieu de conservation (base de données, fichier)
	**Entité externe (acteur)**	Source ou destination des données (utilisateur, API externe)
	**Flux de données**	Données échangées entre processus/entités

LiveCampus

Apprentissage connecté

Niveaux de DFD
Niveau 0 :
Vue d'ensemble du système.

Niveau 1 et suivants :
Détails internes des processus (décomposition).

LiveCampus

Apprentissage connecté

Utilisation en modélisation de menaces (ex: STRIDE)

Élément analysé	Exemple de menace STRIDE	Contre-mesure possible
Flux ➡️ entre l'utilisateur et le formulaire	**Spoofing** ou **Tampering** (falsification)	Chiffrement TLS, CAPTCHA, validation côté serveur
Processus "Vérifier les identifiants"	**Repudiation** (refus d'action)	Journalisation sécurisée, horodatage
Stockage "BDD utilisateurs"	**Information Disclosure** (exfiltration)	Hash des mots de passe, chiffrement de la BDD
Requête SQL vers la BDD	**Tampering** (Injection SQL)	Requêtes préparées, validation des entrées

LiveCampus

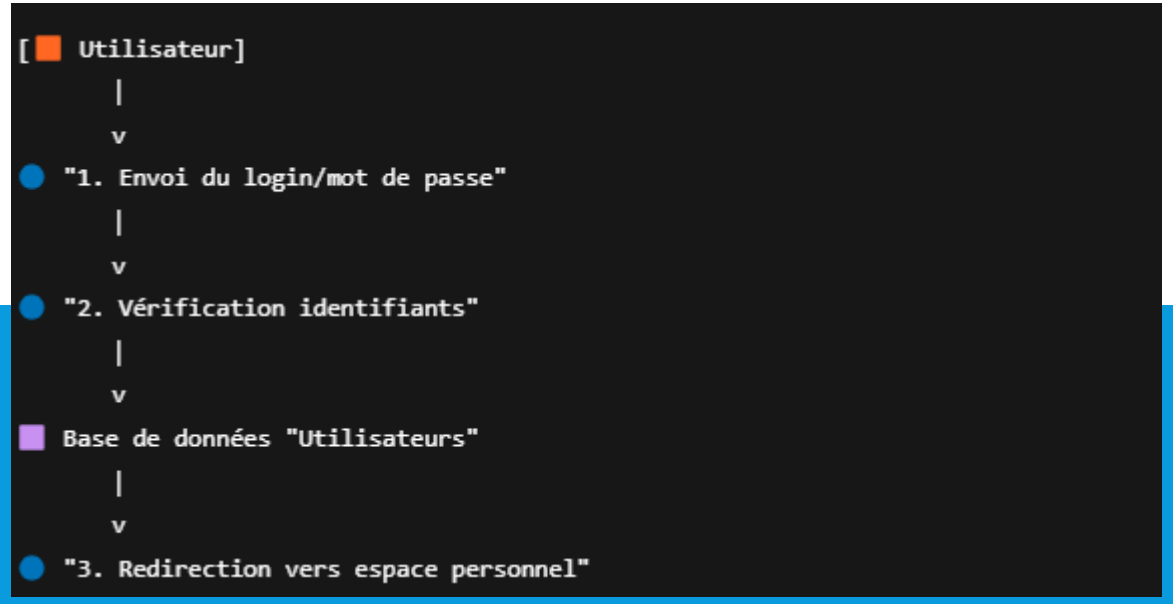
Apprentissage connecté

Pratique	Explication
Définir les **limites de confiance**	Délimiter où les données entrent/sortent du système
Identifier les **acteurs**	Internes vs externes (utilisateur, admin, service tiers)
Cartographier **chaque flux de données**	Quelles données transitent ? (email, mdp, fichiers, tokens…)
Associer chaque **flux à un protocole**	HTTP, HTTPS, SSH, etc. — pour l'analyse des failles réseau
Coupler à **STRIDE** ou **LINDDUN**	Pour identifier et catégoriser les menaces de chaque flux

LiveCampus

Apprentissage connecté

Exemple simplifié



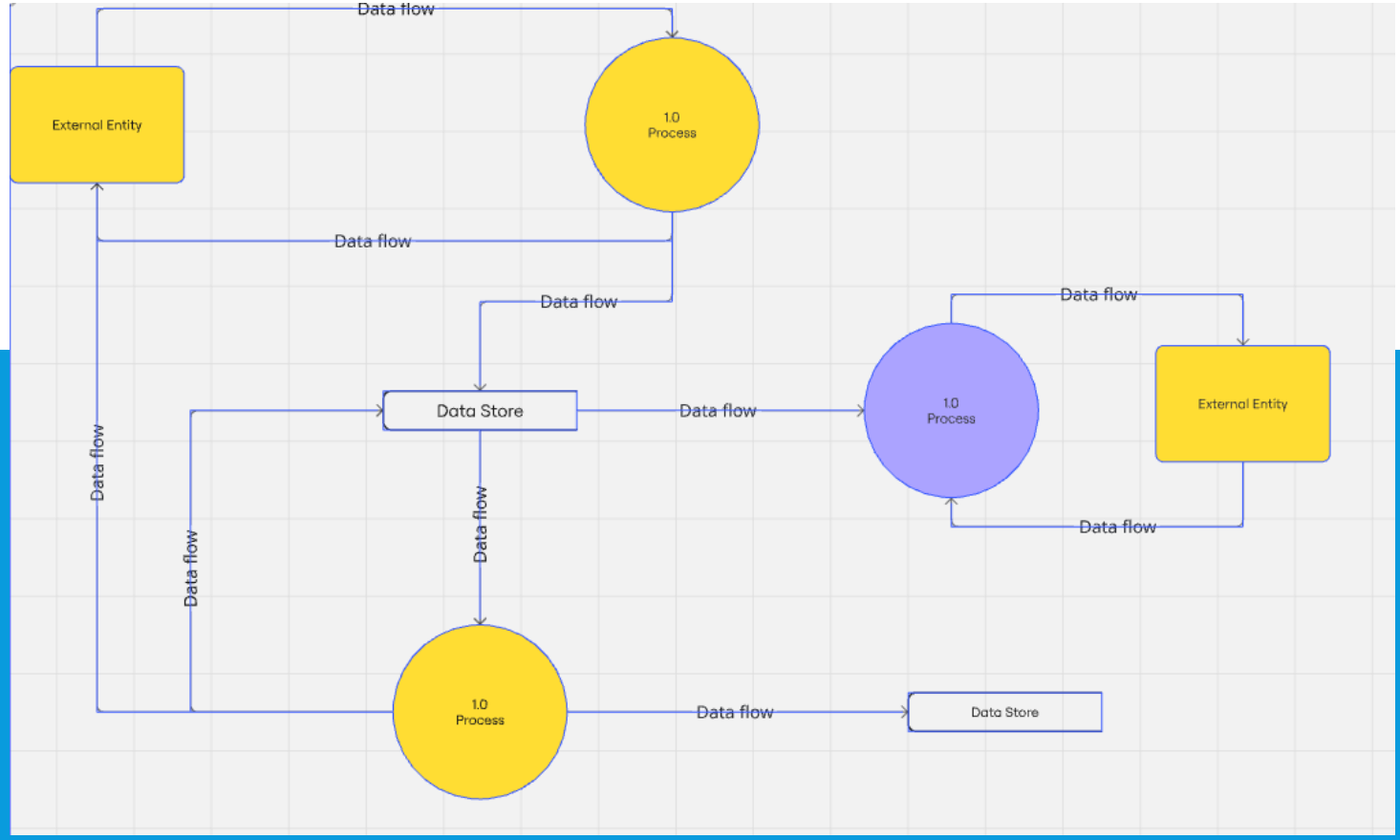
Intérêt pour la sécurité

- Visibilité globale sur le système.
- Repérage clair des zones à sécuriser.
- Facilite le dialogue entre développeurs, DevOps, sécurité, clients.

LiveCampus

Apprentissage connecté

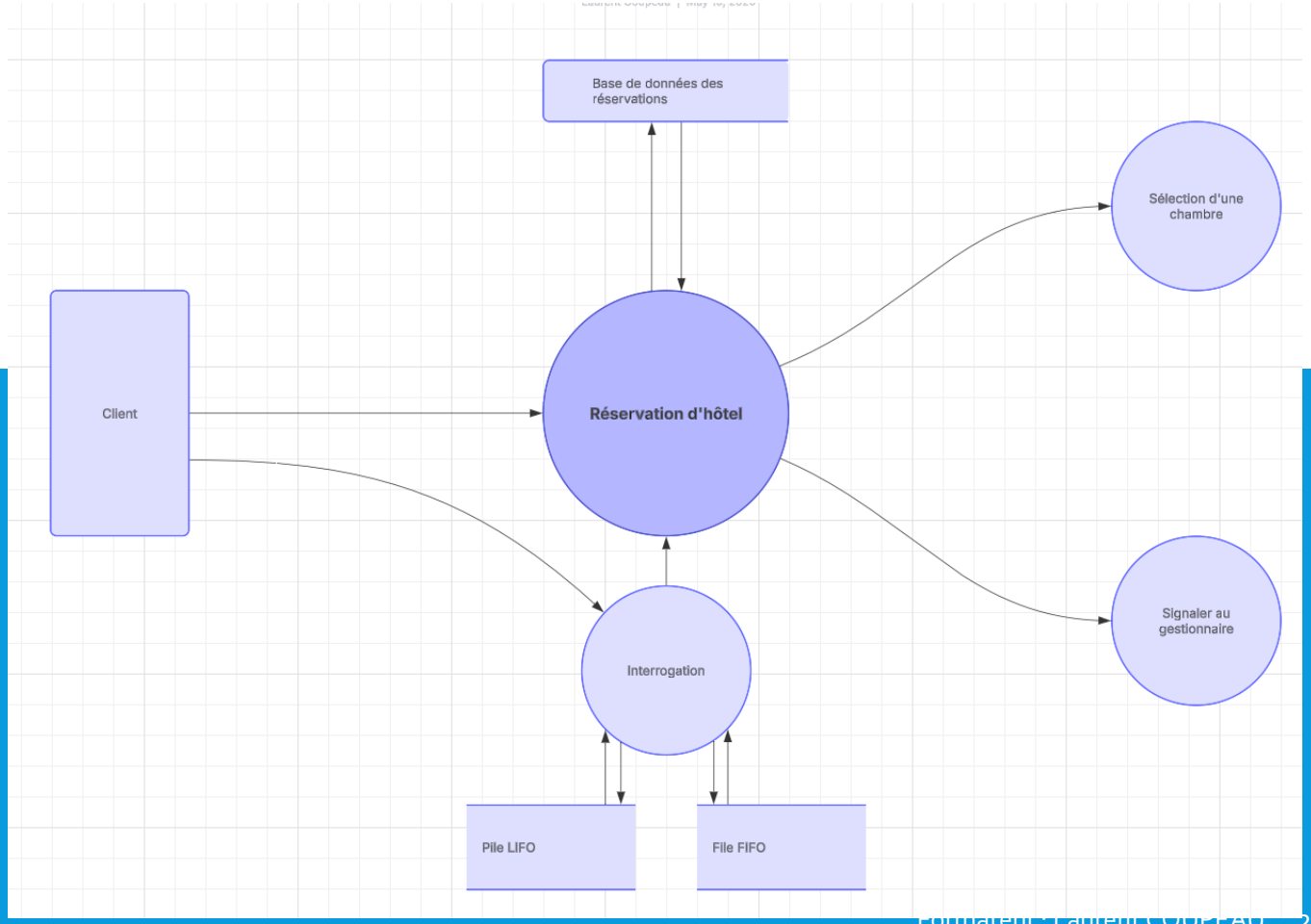
Exemples plus complexes



LiveCampus

Apprentissage connecté

Exemples plus complexes



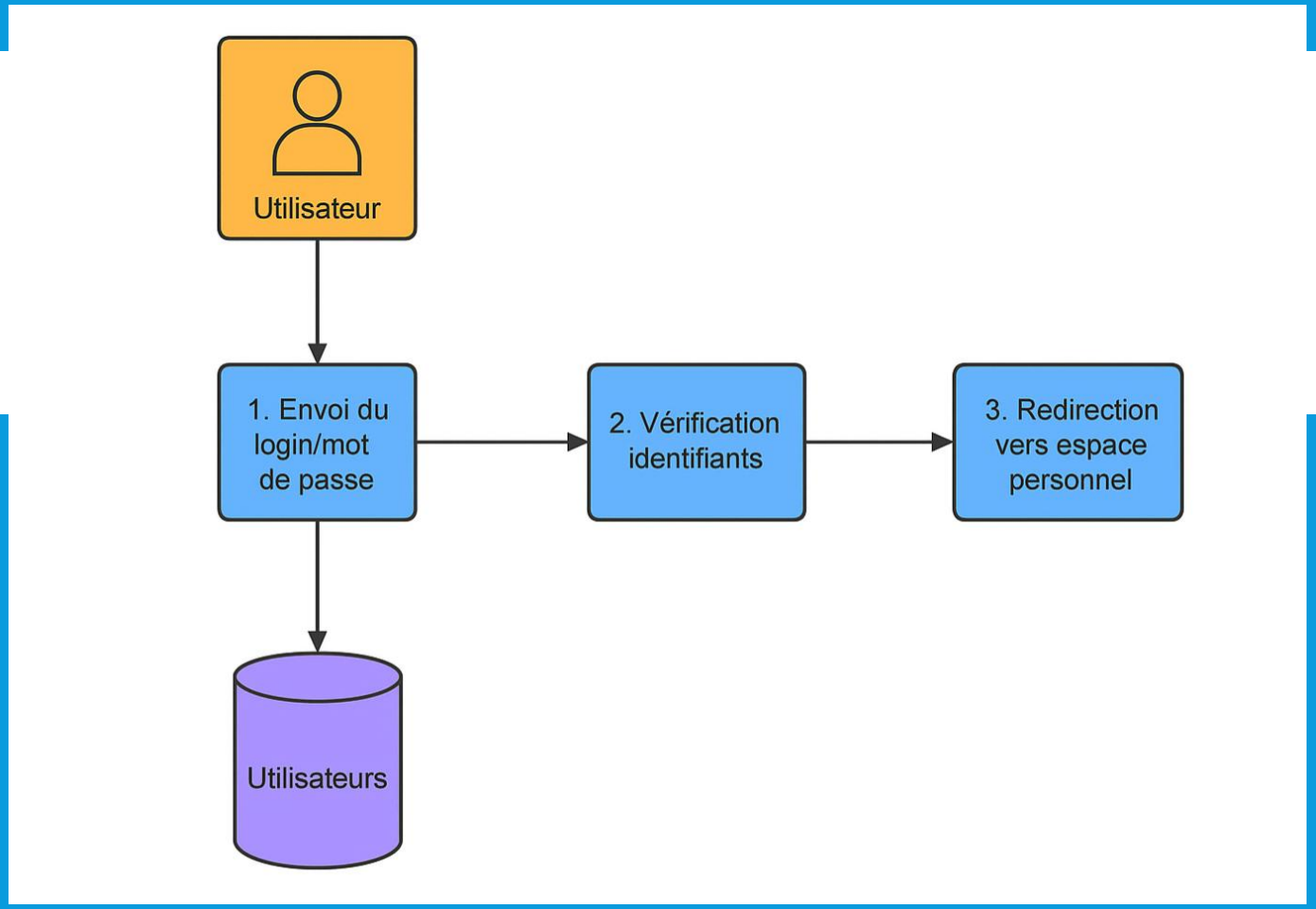
LiveCampus

Apprentissage connecté

Ce que fait un DFD	Ce qu'un DFD ne fait pas
Représente le **flux des données**	Ne montre pas l'interface utilisateur
Identifie les **interactions clés**	Ne décrit pas le code ou les algorithmes
Sert de **base à la modélisation des menaces**	N'analyse pas automatiquement les risques
Donne un **langage commun** entre équipes	Ne remplace pas les tests ou le monitoring

LiveCampus

Apprentissage connecté



LiveCampus

Apprentissage connecté

Analyse des menaces

L'analyse des menaces est une étape cruciale qui consiste à identifier, évaluer, et prioriser les risques liés à la sécurité d'un système, d'une application ou d'un processus. Elle permet d'anticiper les vulnérabilités potentielles et d'adapter les mesures de protection pour réduire les risques.

Objectifs de l'analyse des menaces

- **Comprendre** les risques qui pèsent sur un système.
- **Identifier** les points faibles susceptibles d'être exploités.
- **Définir** des priorités en matière de sécurité.
- **Soutenir** la prise de décision dans la conception et le développement.
- **Adapter** les contrôles de sécurité de manière ciblée et efficace.

LiveCampus

Apprentissage connecté

Étapes de l'analyse des menaces

Étape	Description
a) Identification des actifs	Repérer ce qui doit être protégé (données, services, matériel, processus).
b) Identification des menaces	Lister toutes les menaces possibles : attaques externes, erreurs internes, défaillances.
c) Identification des vulnérabilités	Identifier les failles ou points faibles qui pourraient être exploités par une menace.
d) Évaluation du risque	Combiner la probabilité d'occurrence et l'impact pour estimer le niveau de risque.
e) Priorisation	Classer les risques pour traiter en priorité les plus critiques.
f) Proposition de mesures	Définir les contrôles et contre-mesures adaptés (techniques, organisationnelles, humaines).

LiveCampus

Apprentissage connecté

Méthodes et cadres d’analyse

STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) : méthode classique de Microsoft pour catégoriser les menaces.
PASTA (Process for Attack Simulation and Threat Analysis) : approche basée sur des scénarios d’attaque.
LINDDUN : focalisée sur la confidentialité et la vie privée.
OWASP Threat Dragon : outil open-source pour modéliser les menaces graphiquement.

LiveCampus

Apprentissage connecté

Utilisation :
STRIDE est souvent utilisée dans la modélisation des menaces, par exemple lors de la création d'un diagramme de flux de données (DFD) pour identifier où chaque type de menace pourrait survenir.

STRIDE

Origine : Méthode développée par Microsoft pour classer les menaces en fonction de six catégories distinctes.
Objectif : Aider à identifier les types spécifiques de menaces auxquelles un système peut être exposé.

Acronyme	Signification	Description	Exemple
S	**Spoofing**	Usurpation d'identité : un attaquant se fait passer pour une autre entité (utilisateur, système).	Vol d'identifiants pour se connecter à un compte utilisateur.
T	**Tampering**	Altération non autorisée de données ou de code.	Modification frauduleuse du contenu d'une base de données.
R	**Repudiation**	Refus d'une action effectuée, empêchant la preuve d'une opération.	Un utilisateur nie avoir effectué une transaction.
I	**Information Disclosure**	Divulgation non autorisée d'informations sensibles.	Fuite de données personnelles via une faille dans une API.
D	**Denial of Service**	Rendre un service indisponible, souvent par saturation.	Attaque DDoS qui bloque l'accès au site web.
E	**Elevation of Privilege**	Un utilisateur obtient des privilèges supérieurs à ceux autorisés.	Exploitation d'une faille pour obtenir un accès administrateur.

LiveCampus

Apprentissage connecté

PASTA (Process for Attack Simulation and Threat Analysis)

Origine : Méthode d'analyse des menaces centrée sur les attaques, combinant aspects techniques et métier.
Objectif : Simuler des attaques réelles et analyser leurs impacts pour mieux protéger le système.

Étape	Description
1. Définition du périmètre métier et technique	Comprendre le contexte, les actifs et les objectifs.
2. Définition des objectifs de sécurité	Identifier les exigences en termes de confidentialité, intégrité, etc.
3. Analyse technique et modélisation	Modéliser les flux, identifier les points faibles.
4. Simulation d'attaques	Imaginer des scénarios d'attaque réalistes.
5. Évaluation des risques	Analyser la probabilité et l'impact des attaques.
6. Proposition de mesures de sécurité	Définir des contrôles pour réduire les risques.

Exemple :Pour une application bancaire, PASTA va modéliser des attaques comme l'injection SQL ou le phishing, puis analyser comment ces attaques pourraient compromettre les actifs et proposer des contre-mesures spécifiques.

LiveCampus

Apprentissage connecté

Utilisation :
Particulièrement utile dans les projets manipulant des données personnelles (RGPD) pour s'assurer que la conception respecte la confidentialité.

LINDDUN
Origine : Méthode axée sur la protection de la vie privée et la confidentialité, développée dans le cadre de la privacy engineering.Objectif : Identifier les menaces liées à la confidentialité, comme la collecte, le traitement ou la divulgation non autorisée de données personnelles.

Lettre	Menace associée	Description
L	Linkability	Capacité à relier deux actions ou données à une même entité.
I	Identifiability	Possibilité d'identifier une personne à partir de données.
N	Non-repudiation (absence)	Incapacité à prouver qu'une action n'a pas eu lieu.
D	Detectability	Possibilité de détecter l'existence d'une donnée ou d'un événement.
D	Disclosure of information	Divulgation non autorisée d'informations sensibles.
U	Unawareness	Manque de connaissance des utilisateurs sur la gestion des données.
N	Non-compliance	Non-respect des règles et normes relatives à la vie privée.

LiveCampus

Apprentissage connecté

OWASP Threat Dragon

Origine : Outil open-source de modélisation des menaces développé par OWASP.

Objectif : Permettre aux équipes de sécurité et développement de créer, visualiser et gérer les modèles de menaces facilement, collaborativement et graphiquement.

Fonctionnalités :

- Création intuitive de diagrammes DFD.
- Assignation des menaces selon STRIDE.
- Collaboration en temps réel.
- Export des modèles au format JSON, SVG, PNG.
- Intégration facile dans le cycle de développement Agile ou DevSecOps.

Exemple d'usage :Une équipe utilise Threat Dragon pour modéliser les flux de données d'une application mobile, identifier les menaces STRIDE, et générer des rapports de risque à partager avec les développeurs.

LiveCampus

Apprentissage connecté

Méthode / Outil	Focus principal	Utilisation clé	Exemple d'application
STRIDE	Typologie des menaces	Identification rapide et catégorisation	Modélisation des menaces sur un DFD
PASTA	Simulation d'attaques	Analyse approfondie et orientée métier	Scénarios d'attaque pour une application bancaire
LINDDUN	Confidentialité et vie privée	Analyse des risques liés à la protection des données personnelles	Protection RGPD sur une plateforme web
OWASP Threat Dragon	Outil de modélisation graphique	Collaboration, documentation et suivi des menaces	Création de modèles de menaces en équipe

LiveCampus

Apprentissage connecté

Bonnes pratiques d'architecture sécurisée

Catégorie	Bonnes pratiques	Description / Exemple
Séparation des responsabilités	Isoler les composants critiques pour limiter l'impact d'une faille	Séparer bases de données, serveurs d'applications, front-end et back-end
Principe du moindre privilège	Attribuer aux utilisateurs et services uniquement les permissions nécessaires	Les services n'ont accès qu'aux ressources dont ils ont besoin
Authentification forte	Utiliser l'authentification multi-facteurs (MFA) et politiques robustes	Mots de passe complexes, MFA par OTP ou certificat client
Chiffrement des données	Chiffrer les données en transit (TLS) et au repos (AES 256 bits)	Utiliser HTTPS, chiffrement des bases et des backups
Contrôle d'accès granulaire	Mettre en place des contrôles d'accès basés sur les rôles (RBAC) ou attributs (ABAC)	Les utilisateurs ne peuvent accéder qu'à ce qui est autorisé
Validation et filtrage des entrées	Valider toutes les entrées utilisateur côté serveur pour éviter injections et attaques XSS	Validation côté backend et frontend
Journalisation et surveillance	Collecter les logs et analyser les événements pour détecter les comportements suspects	Centralisation des logs avec outils SIEM
Conception modulaire	Développer en modules indépendants pour faciliter les mises à jour et le patching	Microservices, APIs bien définies
Redondance et haute disponibilité	Prévoir des mécanismes de basculement et de reprise après sinistre	Load balancers, clusters, sauvegardes régulières
Sécurité réseau	Utiliser des pare-feux, segmentation réseau, VPN pour isoler et protéger les flux	VLANs, règles firewall, chiffrement VPN
Mise à jour et patching	Mettre régulièrement à jour les logiciels et composants pour corriger les vulnérabilités	Automatisation des patchs via CI/CD
Test et audit de sécurité	Effectuer des tests d'intrusion réguliers et des audits de code	Pentests, analyse statique et dynamique du code

LiveCampus

Apprentissage connecté

Glossaire

A

AES : Advanced Encryption Standard, algorithme de chiffrement symétrique utilisé pour protéger les données.

ABAC : Attribute-Based Access Control, contrôle d'accès basé sur des attributs comme rôle, heure, etc.

API : Application Programming Interface, interface de programmation permettant d'interagir avec un service.

Audit : Processus d'évaluation de la sécurité et de la conformité d'un système ou d'une application.

LiveCampus

Apprentissage connecté

Glossaire

B

Backup : Sauvegarde des données pour récupération en cas de perte ou d'incident.
Base de données : Ensemble organisé de données stockées pour faciliter l'accès et la gestion.

C

CI/CD : Continuous Integration / Continuous Deployment, méthodes d'automatisation des tests et des déploiements logiciels.
Chiffrement : Processus de transformation des données en une forme codée pour empêcher l'accès non autorisé.

LiveCampus

Apprentissage connecté

Glossaire

D

DoS : Denial of Service, attaque visant à rendre un service indisponible en le surchargeant.

Déni de service distribué (DDoS) : Variante du DoS utilisant plusieurs sources pour l'attaque.

I

IDS : Intrusion Detection System, système détectant les intrusions dans un réseau ou système.

IPS : Intrusion Prevention System, système capable de bloquer automatiquement les intrusions détectées.

Injection SQL : Technique d'attaque qui injecte du code SQL malveillant via des entrées non sécurisées.

LiveCampus

Apprentissage connecté

Glossaire

J

JWT : JSON Web Token, standard ouvert pour représenter des informations d'identification et d'autorisation de manière sécurisée.

L

LVM : Logical Volume Manager, technologie pour gérer le stockage de disques en volumes logiques flexibles.

LDAP : Lightweight Directory Access Protocol, protocole pour accéder et gérer des services d'annuaire.

LiveCampus

Apprentissage connecté

Glossaire

M

MFA : Multi-Factor Authentication, méthode d'authentification combinant plusieurs facteurs (ex : mot de passe + code SMS).

MitM (Man-in-the-Middle) : Attaque où un tiers intercepte et modifie la communication entre deux parties sans qu'elles le sachent.

N

NAT : Network Address Translation, technique de traduction d'adresses IP entre réseau privé et public.

LiveCampus

Apprentissage connecté

Glossaire

O

OAuth : Protocole d'autorisation permettant à des applications d'agir au nom d'un utilisateur sans partager son mot de passe.

R

RBAC : Role-Based Access Control, contrôle d'accès basé sur les rôles assignés aux utilisateurs.

Ransomware : Logiciel malveillant qui chiffre les données d'une victime et demande une rançon pour les débloquer.

LiveCampus

Apprentissage connecté

Glossaire

S

SIEM : Security Information and Event Management, outil centralisant et analysant les journaux (logs) de sécurité.

Spoofing : Technique d'usurpation d'identité dans un réseau pour accéder à des ressources protégées.

SQL Injection : Voir Injection SQL.

SSL/TLS : Protocoles pour sécuriser les communications sur Internet via le chiffrement.

LiveCampus

Apprentissage connecté

Glossaire

T

TLS : Transport Layer Security, protocole de chiffrement pour sécuriser les communications sur Internet, successeur de SSL.

V

VLAN : Virtual Local Area Network, segmentation logique d'un réseau physique pour isoler des groupes d'utilisateurs.

VPN : Virtual Private Network, réseau privé virtuel permettant une communication sécurisée sur un réseau public.

LiveCampus

Apprentissage connecté

Glossaire

X

XSS (Cross-Site Scripting) : Vulnérabilité web permettant d'injecter du code malveillant dans une page web consultée par d'autres utilisateurs.