Durée :	1 jour soit 7 heures	Référence :	GH-500T00

Public visé :

Ce cours s'adresse aux étudiants qui souhaitent comprendre et mettre en œuvre des pratiques de sécurité avancées à l'aide de GitHub Advanced Security (GHAS). Ils apprendront à renforcer significativement les processus de développement logiciel et à construire un écosystème de développement plus résilient et plus sécurisé grâce à des solutions pensées pour les développeurs, permettant de protéger le code, la chaîne d'approvisionnement logicielle (*software supply chain*) et les secrets avant toute mise en production.

Ils découvriront également comment GHAS offre aux équipes de sécurité une visibilité globale sur la posture de sécurité et la chaîne d'approvisionnement logicielle à l'échelle de l'organisation, tout en leur donnant accès à une veille de sécurité de premier plan, alimentée par des millions de développeurs et de chercheurs en sécurité à travers le monde.

Pré-requis :

Prérequis nécessaires :

- Connaissance de base de Git et GitHub (dépôts, commits, pull requests, workflow collaboratif)
- Notions générales de développement logiciel (cycle de vie, CI/CD)
- Aisance avec l'utilisation d'une ligne de commande

Prérequis non obligatoires mais recommandés :

- Expérience avec GitHub Actions (les workflows d'analyse de code s'appuient dessus)
- Sensibilisation générale aux enjeux de sécurité applicative (types de vulnérabilités courantes)
- Notions de base en administration d'organisation/dépôts GitHub (droits, rôles) pour tirer pleinement parti du module 7

Objectifs pédagogiques :

À l'issue de cette formation d'une journée, le stagiaire sera capable de :

- Comprendre le rôle de GitHub Advanced Security dans la sécurisation du cycle de développement logiciel
- Mettre en place une surveillance automatisée des dépendances vulnérables via Dependabot
- Détecter et traiter les secrets exposés dans un dépôt à l'aide du Secret Scanning
- Configurer et exploiter l'analyse de code, notamment via CodeQL, pour identifier des vulnérabilités dans une base de code
- Administrer les fonctionnalités GHAS et les droits d'accès associés au niveau organisation
- Définir des règles et stratégies de sécurité applicables aux dépôts (rulesets, gestion des données sensibles)

Compétences acquises à l'issue de la formation :

- Être capable de configurer et exploiter la veille des dépendances : mettre en place Dependabot (alertes, mises à jour de sécurité automatisées, notifications, rapports) et effectuer une revue des dépendances d'un dépôt.
- Savoir détecter et gérer les fuites de secrets dans un dépôt GitHub : configurer et utiliser l'analyse des secrets (Secret Scanning) pour identifier des identifiants ou clés exposés avant qu'ils ne soient exploités.
- Être capable de mettre en œuvre l'analyse de code (Code Scanning) : configurer l'analyse de code, y compris avec des outils tiers, sur un dépôt GitHub.
- Maîtriser l'utilisation de CodeQL pour l'identification de vulnérabilités : préparer une base de données CodeQL, exécuter une analyse, interpréter les résultats et corriger les problèmes identifiés.
- Savoir personnaliser un workflow d'analyse de code avec CodeQL : adapter des requêtes CodeQL, utiliser la CLI CodeQL, et configurer une matrice de langages pour des projets multi-langages.
- Être en mesure d'administrer GitHub Advanced Security au niveau organisation : activer GHAS, gérer les droits d'accès et piloter les alertes et fonctionnalités associées.
- Savoir définir et appliquer des stratégies de sécurité au niveau des dépôts : créer des repository rulesets, gérer les données sensibles et exploiter les rapports/journalisation de sécurité.

Modalités pédagogiques :

Session dispensée en présentiel ou téléprésentiel, selon la modalité inter-entreprises ou intra-entreprises sur mesure. La formation

est animée par un(e) formateur(trice) durant toute la durée de la session et présentant une suite de modules théoriques clôturés par des ateliers pratiques validant l'acquisition des connaissances. Les ateliers peuvent être accompagnés de Quizz. L'animateur(trice) présente la partie théorique à l'aide de support de présentation, d'animation réalisée sur un environnement de démonstration. En présentiel comme en téléprésentiel, l'animateur(trice) accompagne les participants durant la réalisation des ateliers.

Note relative au programme : Les ateliers, démonstrations, travaux pratiques et scénarios présentés dans ce programme sont fournis à titre d'exemples illustratifs et ne sont pas exhaustifs. Afin de garantir une adéquation permanente avec les évolutions technologiques, le contenu pédagogique ainsi que les environnements de travaux pratiques (TP) sont susceptibles d'être adaptés, ajustés ou mis à jour au cours du temps.

Moyens et supports pédagogiques :

Cadre présentiel

Salles de formation équipées et accessibles aux personnes à mobilité réduite.

- Un poste de travail par participant
- Un support de cours numérique ou papier (au choix)
- Un bloc-notes + stylo
- Vidéoprojection
- Connexion Internet
- Accès extranet pour partage de documents et émargement électronique

Cadre téléprésentiel

Session dispensée via notre solution iClassRoom s'appuyant sur Microsoft Teams.

- Un compte Office 365 par participant
- Un poste virtuel par participant
- Un support numérique (PDF ou Web)
- Accès extranet pour partage de documents et émargement électronique

Modalités d'évaluation et de suivi :

Avant

Afin de valider le choix d'un programme de formation, une évaluation des prérequis est réalisée à l'aide d'un questionnaire en ligne ou lors d'un échange avec le formateur(trice), qui valide la base de connaissances nécessaires.

Les stagiaires disposent également, depuis leur extranet, d'un **outil d'auto-positionnement** leur permettant d'évaluer leur niveau de maîtrise des compétences visées par la formation. Cette auto-évaluation est réalisée avant le démarrage de la formation à l'aide d'un curseur de positionnement allant du niveau le plus faible au niveau le plus élevé.

Pendant

Après chaque module théorique, un ou plusieurs ateliers pratiques permettent de valider progressivement l'acquisition des connaissances. Un quiz peut accompagner les ateliers pratiques afin de mesurer la compréhension des notions abordées.

Après

À l'issue de la formation, les stagiaires réalisent une nouvelle **auto-évaluation** sur les mêmes compétences que celles évaluées avant la formation. Cette comparaison permet de mesurer leur perception de la progression réalisée et de visualiser les compétences développées au cours de la formation.

Lorsque le programme de formation le prévoit, un examen de certification est organisé conformément aux modalités définies par l'éditeur ou le centre de certification (TOSA, Pearson VUE, ENI, PeopleCert…).

Le formateur réalise également une **évaluation individuelle des compétences** de chaque stagiaire sur l'ensemble des compétences visées par la formation. Pour chacune d'elles, il indique si la compétence est :

- Acquise ;
- En cours d'acquisition ;
- Non acquise.

Lorsque la compétence est évaluée comme **en cours d'acquisition** ou **non acquise**, le formateur formule des observations précisant les difficultés rencontrées ainsi que les actions ou recommandations permettant au stagiaire de poursuivre sa montée en compétences.

Enfin
Un questionnaire de satisfaction permet au participant d'évaluer la qualité de la prestation de formation.

Informations sur l'accessibilité :

Nos formations sont, dans la mesure du possible, conçues pour être accessibles à toutes et à tous. Afin de garantir les meilleures conditions d'accueil et d'apprentissage pour les personnes en situation de handicap, nous vous invitons à contacter notre référente handicap certifiée :
Céline SOLATGES – 05 61 34 39 80 – csolatges@iform.fr
Nous vous remercions de bien vouloir nous communiquer toute information utile à ce sujet en amont de la formation, afin de mettre en place les adaptations nécessaires et d'assurer un accompagnement optimal.
Pour en savoir plus sur les dispositifs d'accompagnement existants, vous pouvez consulter les sites suivants :

- AGEFIPH
- FIPHFP
- MDPH de votre département

Description / Contenu

GitHub Advanced Security (GHAS) joue un rôle essentiel dans le renforcement de la sécurité des projets de développement logiciel hébergés sur GitHub. Il fournit un ensemble complet d'outils et de fonctionnalités conçus pour identifier et corriger les vulnérabilités de sécurité tout au long du cycle de vie du développement. En intégrant directement la sécurité au processus de développement grâce à GHAS, votre équipe peut concevoir des logiciels plus sûrs et plus fiables. Ce cours explique comment exploiter GitHub Advanced Security afin d'optimiser la sécurité des développements, tout en comprenant son fonctionnement et son rôle dans l'écosystème de la sécurité.

Module 1 : Introduction à GitHub Advanced Security (GHAS)

- Définir GHAS et l'importance de ses fonctionnalités principales
- Comment utiliser GHAS pour maximiser son impact
- Comprendre GHAS et son rôle dans l'écosystème de la sécurité

Module 2 : Configurer les mises à jour de sécurité Dependabot sur votre dépôt GitHub

- Gérer vos dépendances dans GitHub
- Alertes Dependabot
- Mises à jour de sécurité Dependabot
- Gérer les notifications et les rapports Dependabot
- Révision des dépendances
- Exercice - Configurer les mises à jour de sécurité Dependabot

Module 3 : Configurer et utiliser l'analyse des secrets (*secret scanning*) dans votre dépôt GitHub

- Qu'est-ce que l'analyse des secrets (*Secret Scanning*) ?
- Configurer l'analyse des secrets (*Secret Scanning*)
- Utiliser l'analyse des secrets (*Secret Scanning*)

Module 4 : Configurer l'analyse du code (*Code Scanning*) dans GitHub

- Qu'est-ce que l'analyse du code (*Code Scanning*) ?
- Activer l'analyse du code avec des outils tiers
- Configurer l'analyse du code (*Code Scanning*)
- Exercice - Configurer l'analyse du code (*Code Scanning*)

Module 5 : Identifier les vulnérabilités de sécurité dans votre base de code à l'aide de CodeQL

- Préparer une base de données pour CodeQL
- Exécuter CodeQL dans une base de données
- Comprendre les résultats de CodeQL
- Résoudre les problèmes liés aux résultats de CodeQL

Module 6 : Analyse du code avec GitHub CodeQL

- Qu'est-ce que CodeQL ?
- Comment CodeQL analyse-t-il le code ?
- Qu'est-ce que le langage QL ?
- Analyse du code (*Code Scanning*) et CodeQL
- Personnaliser votre workflow d'analyse du code avec CodeQL - Partie 1
- Exercice - Référencer une requête CodeQL
- Personnaliser votre workflow d'analyse du code avec CodeQL - Partie 2

- Utiliser l'interface en ligne de commande (CLI) CodeQL
- Personnaliser les langages et les builds pour l'analyse du code
- Exercice - Configurer une matrice de langages CodeQL

Module 7 : Administration GitHub pour GitHub Advanced Security (GHAS)

- Qu'est-ce que GitHub Advanced Security ?
- Activer GitHub Advanced Security
- Gérer l'accès à GitHub Advanced Security
- Gérer les fonctionnalités et les alertes de GitHub Advanced Security

Module 8 : Gérer les données sensibles et les stratégies de sécurité dans GitHub

- Définir les stratégies de sécurité
- Créer et gérer les ensembles de règles de dépôt (*repository rulesets*)
- Rapports et journalisation