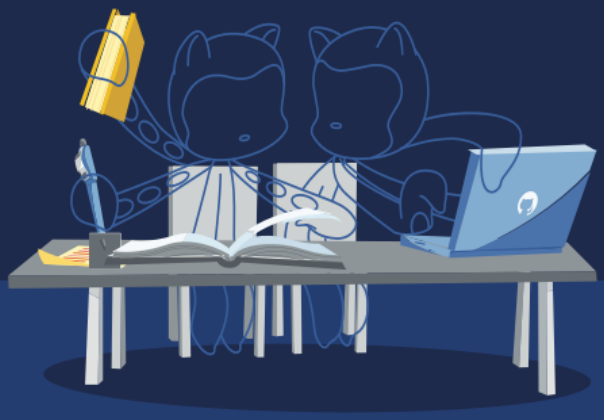
GitHub

Guia de estudo
GitHub Advanced Security



Prepare-se para o exame da certificação do GitHub Advanced Security com nosso abrangente guia de estudo. Nós consolidamos as atividades de aprendizado e os recursos essenciais para melhor preparar você para o exame e impulsionar suas chances de sucesso.

Perfil do público

Este exame foi concebido para profissionais experientes na área de segurança e desenvolvimento de software. Esta certificação foi concebida para indivíduos que têm um profundo conhecimento do GitHub e de seus recursos de segurança, bem como experiência prática em fluxos de trabalho de desenvolvimento de software de segurança.

Domínios de objetivos

Um domínio de objetivo para um exame de certificação, geralmente denominado como “domínio” ou “domínio do exame”, é uma estrutura ou um resumo estruturado que define tópicos, habilidades e conhecimentos específicos que o exame da certificação vai abranger. Ele fornece um roteiro claro para que os candidatos saibam o que poderão encontrar no exame e o que precisam estudar para se preparar.

Os domínios fornecidos neste guia de estudo têm como objetivo fornecer insights sobre as categorias dos tópicos abordados no exame do GitHub Advanced Security, junto com o objetivo de aprendizado em cada domínio.

Discriminação dos domínios	Porcentagens dos exames
Domínio 1: Descrever as funcionalidades e os recursos de segurança do GHAS	10%
Domínio 2: Configurar e usar a verificação de segredo	10%
Domínio 3: Configurar e usar o gerenciamento de dependência	15%
Domínio 4: Configurar e usar a varredura de código	15%
Domínio 5: Usar varredura de código com o CodeQL	20%
Domínio 6: Descrever as melhores práticas do GitHub Advanced Security	20%
Domínio 7: Configurar as ferramentas do GitHub Advanced Security no GitHub Enterprise	10%

Recomendações e as melhores práticas para o sucesso

Para aumentar as chances de obter sucesso no exame do GitHub Advanced Security, os candidatos devem ter um profundo conhecimento do GitHub e de seus recursos de segurança, bem como experiência prática em fluxos de trabalho de desenvolvimento de software de segurança. Os caminhos de aprendizado recomendados para esse exame proporcionam um estudo aprofundado do conteúdo de aprendizado, seguido de exercícios práticos e perguntas preparatórias para a avaliação que foram criados para possibilitar uma preparação e um conhecimento aperfeiçoados para o exame da certificação.

Recursos de conteúdo

Os recursos a seguir foram criados em colaboração com o GitHub como um conteúdo recomendado que abrange os objetivos de aprendizado em cada domínio para o exame do GitHub Advanced Security. Tanto o Microsoft Learn como o LinkedIn Learning fornecem um caminho de aprendizado completo para o exame, mas oferecem uma experiência de aprendizado diferente.

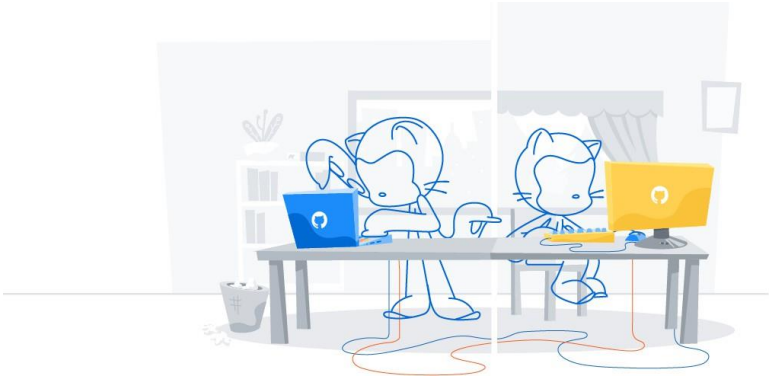
Microsoft Learn

O caminho de aprendizado do GitHub Advanced Security no MS Learn fornece uma coleção robusta de módulos de aprendizado elaborados para preparar você para o exame do GitHub Advanced Security. Saiba como proteger seu código com recursos de segurança avançados em todas as fases do ciclo de vida do desenvolvimento. O GitHub Advanced Security é um complemento do GitHub Enterprise que permite usar recursos de segurança, como verificação de segredo, varredura de código e gerenciamento de dependência nos repositórios privados. Os módulos a seguir o guiarão pelos recursos do GitHub Advanced Security e fornecerão as habilidades necessárias para que você possa reconhecer, aplicar e avaliar esses recursos no seu próprio ambiente do GitHub.

LinkedIn Learning

LinkedIn Learning

Descubra a arte de proteger seu código usando as técnicas avançadas de segurança por meio de todo o processo de desenvolvimento de software percorrendo o caminho de aprendizado Preparar-se para a certificação do GitHub Advanced Security no LinkedIn Learning. Com o GitHub Advanced Security, você obtém acesso a um pacote de ferramentas robustas de segurança, incluindo verificação de segredo e gerenciamento de dependência, feitas sob medida para seus repositórios privados. Este caminho de aprendizado baseado em vídeos consiste em uma série de cursos envolventes que guiarão você pelas complexidades das funcionalidades do GitHub Advanced Security. Ao finalizar o caminho de aprendizado, você estará bem preparado com bastante conhecimento e experiência para aplicar, avaliar e maximizar facilmente esses recursos de segurança no seu ambiente do GitHub.



Domínio 1: Descrever as funcionalidades e os recursos de segurança do GHAS

Comparar os recursos do GHAS e sua função no ecossistema de segurança
Diferenciar os recursos de segurança que chegam automaticamente para projetos de código aberto e quais recursos estão disponíveis quando o GHAS está emparelhado com o GHEC ou o GHES
Descrever os recursos e benefícios da visão geral da segurança
Descrever as diferenças entre verificação de segredo e varredura de código
Descrever como a verificação de segredo, a varredura de código e o Dependabot criam um ciclo de vida de desenvolvimento de software mais seguro
Comparar um cenário de segurança com uma revisão de segurança isolada e um cenário avançado, com a segurança integrada em cada etapa do ciclo de vida de desenvolvimento de software

Explicar e usar os recursos do GHAS
Descrever como as dependências vulneráveis são identificadas (olhando nos arquivos de manifesto e comparando com bancos de dados de vulnerabilidades conhecidas)
Explicar como agir com alertas do GHAS
Explicar as implicações de ignorar um alerta
Explicar a função de um desenvolvedor ao descobrir um alerta de segurança
Descrever as diferenças no gerenciamento de acesso para visualizar alertas para diferentes recursos de segurança
Descrever uma política de segurança em um repositório do GitHub
Identificar onde usar alertas de Dependabot no ciclo de vida de desenvolvimento de software

Domínio 2: Configurar e usar a verificação de segredo

Habilitar e usar a verificação de segredo
Descrever a verificação de segredo
Escolher quando a verificação de segredo ocorrerá
Comparar a disponibilidade da verificação de segredo para repositórios privados e públicos
Habilitar a verificação de segredo para repositórios privados
Habilitar a verificação de segredo para uma organização
Explicar como selecionar uma resposta apropriada para um alerta de verificação de segredo
Determinar se um alerta será gerado para um determinado segredo, padrão ou provedor de serviços
Determinar se uma determinada função de usuário verá alertas de verificação de segredo

Personalizar o comportamento padrão da verificação de segredo
Configurar os destinatários de um alerta de verificação de segredo (também inclui como fornecer acesso para membros e equipes que não sejam administradores)
Descrever como excluir determinados arquivos para que não sejam verificados quanto a segredos
Explicar como habilitar a verificação de segredo personalizada para um repositório
Explicar como habilitar a verificação de segredo personalizada para uma organização

Domínio 3: Configurar e usar o gerenciamento de dependência

Descrever as ferramentas para o gerenciamento de vulnerabilidades nas dependências
Definir uma vulnerabilidade
Descrever alertas do Dependabot
Descrever atualizações de segurança do Dependabot
Definir o gráfico de dependência
Descrever como o gráfico de dependência é gerado
Descrever como alertas são gerados para dependências vulneráveis (acionados do gráfico de dependência, originados do Banco de Dados Consultivo - GitHub Advisory Database e do WhiteSource)

Habilitar e configurar ferramentas para o gerenciamento de dependências vulneráveis
Identificar as configurações padrão dos alertas do Dependabot em repositórios privados e públicos
Identificar as permissões e funções necessárias para habilitar alertas do Dependabot
Identificar as permissões e funções necessárias para visualizar alertas do Dependabot
Habilitar alertas do Dependabot para repositórios privados
Habilitar alertas do Dependabot para orgnaizações
Criar um arquivo de configuração válido do Dependabot
Configurar as notificações para as dependências vulneráveis

Identificar e corrigir as dependências vulneráveis
Identificar uma dependência vulnerável de um alerta do Dependabot
Identificar dependências vulneráveis de um pull request
Habilitar as atualizações de segurança do Dependabot
Corrigir uma vulnerabilidade de uma alerta do Dependabot na guia Segurança (pode incluir atualizar ou remover a dependência)
Corrigir uma vulnerabilidade de uma alerta do Dependabot no contexto de um pull request (pode incluir atualizar ou remover a dependência)
Tomar medidas com relação a alertas do Dependabot testando e fazendo o merge de pull requests

Domínio 4: Configurar e usar a varredura de código

Descrever e habilitar a varredura de código
Descrever a varredura de código
Listar as etapas para habilitar a varredura de código em um repositório usando o GitHub Actions (por exemplo, guia Segurança e clicar em “definir a varredura de código”, definir o fluxo de trabalho do GitHub Actions e fazer as modificações necessárias no fluxo de trabalho)
Habilitar a varredura de código para uso com um fluxo de trabalho de análise do CodeQL
Descrever como a varredura de código relata o consumo do GitHub Actions

Usar a varredura de código com ferramentas de terceiros
Habilitar a varredura de código para uso com uma análise de terceiros
Comparar as etapas para usar o CodeQL versus análises de terceiros ao habilitar a varredura de código
Comparar como implementar a análise do CodeQL em um fluxo de trabalho do GitHub Actions versus uma ferramenta de CI de terceiros

Configurar a varredura de código
Descrever como a varredura de código se ajusta ao ciclo de vida de desenvolvimento de software
Comparar a frequência de fluxos de trabalho de varredura de código (programada versus acionada por eventos)
Escolher um evento acionador para um determinado padrão de desenvolvimento (por exemplo, em um pull request e para arquivos específicos)
Editar o modelo padrão para o fluxo de trabalho de Actions para se ajustar a um repositório de produção, de código aberto e ativo

Domínio 5: Usar varredura de código com o CodeQL

Explicar como o CodeQL habilita a varredura de código
Descrever o CodeQL
Definir um pacote QL, uma consulta de código e um pacote de códigos
Descrever os pacotes de consulta padrão do CodeQL
Descrever como o CodeQL analisa código e produz resultados, incluindo as diferenças entre a linguagem compilada e interpretada

Usar o CodeQL para varredura de código
Introduzir um fluxo de trabalho de análise do CodeQL em um repositório
Listar os locais em que as consultas do CodeQL podem ser especificadas para uso com a varredura de código
Configurar a matriz da linguagem em um fluxo de trabalho do CodeQL
Fazer referência a uma consulta do CodeQL de um repositório público em um fluxo de trabalho de varredura de código
Fazer referência a uma consulta do CodeQL de um repositório privado em um fluxo de trabalho de varredura de código
Fazer referência a uma consulta do CodeQL de um diretório local em um fluxo de trabalho de varredura de código
Fazer referência a um arquivo de configuração em um mesmo repositório
Fazer referência a um arquivo de configuração em um repositório público remoto
Executar a varredura de código com a interface de linha de comandos (CLI) do CodeQL, incluindo a criação do banco de dados do CodeQL, a análise desse banco de dados e a postagem dos resultados SARIF no GitHub
Comparar as etapas para executar a varredura de código no GitHub Actions versus CLI do CodeQL

Descrever como fazer triagem dos resultados da varredura de código da análise do CodeQL
Descrever como fazer a triagem dos resultados da varredura de código da análise do CodeQL
Solucionar os problemas de um fluxo de trabalho de varredura de código com falha usando o CodeQL, incluindo criar e alterar uma configuração personalizada no fluxo de trabalho do CodeQL
Seguir o fluxo de dados por meio de código usando a experiência de caminhos de demonstração
Explicar a razão para um alerta de varredura de código conforme a documentação vinculada do alerta
Determinar se e por que um alerta de varredura de código precisa ser dispensado
Descrever falhas potenciais no CodeQL via o modelo de compilação e o suporte de linguagem
Otimizar os tempos de execução de análise do CodeQL

Usar ferramentas de terceiros com varredura de código
Explicar como fazer upload de resultados SARIF de terceiros via endpoint SARIF
Explicar o propósito de definir uma categoria SARIF

Domínio 6: Descrever as melhores práticas, os resultados e como tomar medidas corretivas no GitHub Advanced Security

Descrever as melhores práticas, os resultados e como tomar medidas corretivas no GitHub Advanced Security
Usar o Common Vulnerabilities and Exposures (CVE) e o Common Weakness Enumeration (CWE) para descrever um alerta do GitHub Advanced Security e listar uma possível correção
Descrever o processo de tomada de decisão para encerrar e dispensar alertas de segurança (documentar a dispensa, tomar uma decisão com base em dados)
Determinar as funções e responsabilidade das equipes de segurança e desenvolvimento em um fluxo de trabalho de desenvolvimento de software
Explicar como definir uma cadência de revisões com as equipes de segurança, quando apropriado
Usar políticas de segurança para instruir todos os contribuidores como proteger melhor os repositórios
Comparar o alerta de varredura de código com a política de segurança do repositório (por exemplo, devemos bloquear merges sem vulnerabilidades de segurança corrigidas?)
Alinhar a configuração de proteção de branch de repositório com políticas de segurança escritas

Domínio 7: Administração do GitHub Advanced Security

Administração do GitHub Advanced Security
Explicar como os recursos do GitHub Advanced Security são habilitados no GitHub Enterprise Server
Explicar como os recursos do GitHub Advanced Security são habilitados para uma organização
Definir políticas de segurança para um repositório
Definir políticas de segurança para uma organização
Descrever como as permissões são interpretadas ao longo do fluxo de trabalho de segurança
Localizar os endpoints de APIs para recursos do GHAS, como verificação de segredo, varredura de código e o Dependabot
Listar os stakeholders que precisam estar envolvidos nos fluxos de trabalho de segurança habilitados pelo GHAS, incluindo suas funções no fluxo de trabalho
Configurar a varredura de código em um repositório ou organização usando a fluxo de trabalho do CodeQL padrão
Identificar as etapas de criação de personalização necessárias em um fluxo de trabalho do CodeQL