GH-500: GitHub Advanced Security

Duration: 2 days (16 hours)

Course description

GH-500: GitHub Advanced Security provides a comprehensive overview of GitHub Advanced Security (GHAS), focusing on its integral features such as secret scanning, code scanning, and Dependabot. It begins with an introduction to GHAS, explaining its importance in the security ecosystem and how it integrates into the development workflow. The 1-day course emphasizes the role of GHAS in identifying and mitigating security vulnerabilities early in the software development lifecycle, thereby enhancing the overall security posture of an organization.

The course modules delve into the specifics of each GHAS feature, detailing how secret scanning helps prevent the exposure of sensitive information like API keys and tokens, while code scanning analyzes source code for security vulnerabilities and coding errors using tools like CodeQL. Dependabot is highlighted for its ability to manage project dependencies by checking for updates and opening pull requests to ensure projects have the latest security patches. The course also covers the configuration and utilization of these features to maximize their security impact.

Additionally, the course explores the practical aspects of implementing GHAS, including configuring Dependabot alerts and security updates, enabling secret scanning, and setting up code scanning workflows. It provides actionable insights and best practices for integrating these security measures into the development process, ensuring that security becomes an integral part of the workflow. The goal is to equip security professionals and developers with the knowledge and tools needed to effectively use GHAS to protect their codebases and maintain a secure development environment.

The course is designed as a blended learning experience that combines instructor-led training with online materials on the Microsoft Learn platform (https://docs.microsoft.com/learn). Students are encouraged to use the content on Learn as reference materials to reinforce what they learn in class and to explore topics in more depth.

IMPORTANT: This course is designed to be delivered in one full day. The activities are approximately 60% instructional led and 40% student interactive exercises and/or instructor demos.

Learning objectives

After completing this course, students will be able to:

- **Define GitHub Advanced Security (GHAS):** Understand the importance of integral features such as Secret scanning, Code scanning, and Dependabot.
- **Utilize GHAS:** Learn how to maximize security impact using GHAS features.
- **Understand GHAS in the Security Ecosystem:** Recognize GHAS's role and its integration into the security workflow.
- **Configure Dependabot:** Learn to enable and configure Dependabot alerts and security updates.
- **Implement Secret Scanning:** Understand how to enable and use secret scanning to prevent secret leaks.

- **Configure Code Scanning:** Learn to implement and configure code scanning using CodeQL and other tools

Audience profile

Audience profile for this course is the following:

- A security professional or developer responsible for implementing and managing security measures within their organization.
- Possesses a foundational understanding of GitHub and its basic functionalities but are looking to deepen their knowledge of advanced security features.
- Desire to learn how to effectively use GitHub Advanced Security (GHAS) to enhance the security of their codebase and development workflow.

Audience prerequisites

The audience for this 1-day course consists of security professionals and developers who are responsible for implementing and managing GitHub security measures within their organizations.

NOTE: The exercise activities in this 1-day class are *Advanced* and require and intermediate knowledge of Git and GitHub functions and features.

Candidates should have the following:

- Experience in using and administering GitHub repositories.
- Experience of working with Microsoft Azure services.
- Technical skills in code scanning, dependency management, and secret scanning, and are familiar with tools like CodeQL and Dependabot.

Prerequisite knowledge to teach this course

To successfully teach these courses, instructors must have a working knowledge of Git, GitHub principles, and Azure Administration.

Learn Modules (8 modules, 6 hours)

There are 8 Learn modules to be covered in approximately 6 hours of teaching. The 8 MS Learn modules are divided into two learning paths:

- **Learning Path: GitHub Advanced Security Part 1 of 2**
- **Learning Path: GitHub Advanced Security Part 2 of 2**

Required materials to prepare for and teach this course

The exercise utilizes the following GitHub technologies:

1. **Secret Scanning**: This technology scans for sensitive information like API keys and tokens to prevent secret leaks.

2. **Code Scanning**: It analyzes source code for security vulnerabilities and coding errors using static analysis tools like CodeQL.

3. **Dependabot**: Manages project dependencies by checking for updates and opening pull requests to ensure projects have recent security patches.

4. **Dependency Graph**: Identifies all upstream dependencies and public downstream dependents of a repository or package.

5. **GitHub Actions**: Used for implementing and configuring code scanning workflows.

6. **SARIF (Static Analysis Results Interchange Format)**: Allows uploading of SARIF files generated outside GitHub to display code scanning alerts in the repository.

Learning Path: GitHub Advanced Security Part 1 of 2

Module 1: Introduction to GitHub Advanced Security

- Introduction
- Define GHAS and the importance of its integral features • How to utilize GHAS to get the most impact
- Understand GHAS and its role in the security ecosystem
- Knowledge check
- **Summary**

Module 2: Configure Dependabot security updates on your GitHub repo

- Introduction
- Manage your dependencies on GitHub
- Dependabot alerts
- Dependabot security updates
- Manage Dependabot notifications and reports
- Dependency review
- Exercise - Configure Dependabot security updates
- Knowledge check
- Summary

Module 3: Configure and use secret scanning in your GitHub repository

- Introduction
- What is secret scanning?

- Configure secret scanning
- Use secret scanning
- Exercise - Introduction to secret scanning
- Knowledge check
- Summary

Module 4: Configure code scanning on GitHub

- Introduction
- What is code scanning?
- Enable code scanning with third party tools
- Configure code scanning
- Exercise - Configure code scanning
- Knowledge check
- Summary

Learning Path: GitHub Advanced Security Part 2 of 2

Module 5: Identify security vulnerabilities in your codebase by using CodeQL

- Introduction
- Prepare a database for CodeQL
- Run CodeQL in a database
- Understand CodeQL results
- Troubleshoot CodeQL results
- Knowledge check
- Summary

Module 6: Code scanning with GitHub CodeQL

- Introduction
- What is CodeQL?
- How does CodeQL analyze code?
- What is QL?
- Code scanning and CodeQL
- Customize your code scanning workflow with CodeQL - Part 1
- Exercise - Reference a CodeQL query
- Customize your code scanning workflow with CodeQL - Part 2

- Use the CodeQL CLI
- Customize languages and builds for code scanning
- Exercise - Configure a CodeQL language matrix
- Knowledge check
- Summary

Module 7: GitHub administration for GitHub Advanced Security

- Introduction
- What is GitHub Advanced Security?
- Enable GitHub Advanced Security
- Manage access to GitHub Advanced Security
- Manage the GitHub Advanced Security features and alerts
- Knowledge check
- Summary

Module 8: Manage sensitive data and security policies within GitHub

- Introduction
- Setting security policies
- Create and manage repository rulesets
- Reporting and logging
- Exercise - Removing a commit from the git history
- Knowledge check
- Summary

Exercises and Demos (6 exercises, 2.5 hours)

Exercises are to be used as hands-on activities for individual students which are led by the instructor, or demonstrations led by the instructor. The decision to lead hands-on activities or perform demonstrations is the instructor's responsibility.

Module 2: Configure Dependabot security updates on your GitHub repo

- Exercise - Configure Dependabot security updates

Module 3: Configure and use secret scanning in your GitHub repository

- Exercise - Intro to secret scanning exercise

Module 4: Configure code scanning on GitHub

- Exercise - Configure code scanning exercise

Module 6: Code scanning with GitHub CodeQL

- Exercise - Reference a CodeQL query
- Exercise - Configure a CodeQL language matrix

Module 8: Manage sensitive data and security policies within GitHub

- Exercise - Removing a commit from git history

Course timing

The following agenda provides estimated times to complete each classroom activity. However, the estimated times may vary depending on the background of your students, which may affect whether you can move faster or slower through the course material.

Estimated times for each Module include the time to complete as part of a 2-day course:

- The module's PowerPoint slide deck presentation (60% of course timing)
- Exercises (40% of course timing)
- Optional: Pre-defined product demonstrations, determined by instructor
- Optional: Knowledge Check questions (see the section on Additional Timing Notes below)

You should adjust the agenda accordingly based on any classroom activities that you personally created or plan to deliver that are not included in the slides for this course. For example, if you plan to present:

- ad-hoc demonstrations
- review activities
- classroom games
- and so on…

Note: Each module activity for the following agenda is the slide deck presentation for that module.

Portal, Cloud Shell, PowerShell, and the CLI (when necessary for the courseware)

The exercise instructions are written to use the Cloud Shell. The Cloud Shell automatically connects to Azure and provides access to PowerShell and the CLI.

If you would rather have students use PowerShell or the CLI locally, you can use these links.

- Install Azure PowerShell on Windows with PowerShellGet
- Install Azure CLI on Windows

Azure subscriptions (when necessary for the courseware)

To complete the exercises and any additional demonstration exercises in this course, students will need an Azure Subscription.

The Azure pass effectively functions in the same way as the publicly available Microsoft Azure Trial Subscription. This means there are limitations on what you can do with the pass.