Web Security Shorts
Securing your GitHub code repository

UNIVERSITY of WASHINGTON
Today’s topics

- **git**, GitHub, and UW GitHub enterprise
- why secure repositories?
- managing access in GitHub
Future topics

- repository security policies
- code scanning
- secret scanning
- dependency management
git, GitHub and UW GitHub
git

- version control system
  - tracks content and history of changes
  - mostly used for source code, but also for other content like books
- allows for coordination and collaboration between contributors working on the same code base
Local and distributed

- every computer used for the project has a full-fledged instance of the repository with a complete history and version tracking
- almost all operations are local, most commonly via the command line
- provides a variety of workflows that give developers flexibility on how they work with each other
Centralized workflow

Base diagram from Git - Distributed Workflows
GitHub
- cloud-based app that hosts shared git repositories
  - other git hosts exist, like BitBucket
- provides a web-based user interface
Centralized workflow with GitHub

Base diagram from Git - Distributed Workflows
UW GitHub Enterprise service

University departments or entities can:

- obtain fully-featured GitHub organization accounts
- with unlimited private repositories
- that are free of charge

also includes features of GitHub Classroom capability
UW GitHub and GitHub accounts

- users of the service create or reuse their own personal GitHub accounts
- the service does NOT provision GitHub accounts
- your NetID will be linked to your GitHub account when you join an organization in UW’s enterprise
- the service is provided through github.com
  - but support provided via help@uw.edu
UW GitHub Enterprise

Outside collaborators: added directly to repos

Organization:
Libraries
repo

Organization:
Front-end Tech CoP
repo

Enterprise members: associated with an organization
UW GitHub access

- **the UW enterprise**
  - access to the enterprise is allowed for current faculty, staff and students
  - shared and sponsored NetIDs can be permitted on request, email help@uw.edu

- **organizations within the enterprise**
  - can be requested by faculty and staff
  - training required for org administrators
Why secure your code repository?
Why?

- Secure management of git repositories can lower the risk of exposing vulnerable code and secrets.
- Modern git hosts are providing more and more tools to help you harden your code, e.g.
  - Dependency management
  - Code scanning
  - Secret scanning and management
Managing access
Visibility of your repository

- **public**
  - everyone on the internet has access

- **internal**
  - anyone in your enterprise has access

- **private**
  - only those that you choose share access
Visibility: public

- accessible to everyone on the internet
- good for open-source projects
- example: AblePlayer, a fully accessible media player
AblePlayer

https://github.com/ableplayer/ableplayer/graphs/code-frequency
Visibility: internal

- based on the concept of “Inner Source”
  - open source within the confines of an enterprise
- only available for GitHub enterprise organizations
- anyone belonging to an organization within the UW enterprise can see any internal repository
- example: UW Storytelling Modules
UW storytelling modules
Internal - who’s looking

UW GitHub Enterprise

Enterprise members: associated with an organization

Organization: Libraries

repo

Organization: Front-end Tech CoP

internal repo
Visibility: private (1 of 2)

- for proprietary and/or code repositories not meant to be public or internal
- unlimited private repositories available with the UW GitHub and the GitHub free version
Visibility: private (2 of 2)

- in UW GitHub, private repositories can only be accessed by members of your organization or specifically invited outside collaborators
- if you are an owner of an org you can set newly created repos to be private by default
Teams and individuals

- in UW GitHub you can fine-grain access to repos within an organization to teams, particular org members, or outside contributors
- when selecting a team or individuals verify you have selected the correct one
  - remember users use their own personal GitHub accounts so may not be easily identifiable
Adding users to repos

Add people to js-session-3
24248 seats left — Buy more

- HappyDays
  Invite outside collaborator
- HappydaysCode
  HappydaysCode • Invite outside collaborator
- HappyDays
  HappyDaysIF • Invite outside collaborator
- happydays1
  Invite outside collaborator
- HappyDays2004
  Invite outside collaborator
Takeaways for managing access

- review your repository visibility settings
  - set visibility as private unless there is a compelling reason to be public or internal
- regularly review teams and individuals assigned to your repos and organizations
More reading

- Securing your repository
Thank you! Questions?

Peter Giles
gilesp@uw.edu

Pete Graff
pgraff@uw.edu

Jeane Marty
jeanem@uw.edu