Incident Report: Privacy Incident Report: Privacy Concerns and Consent Issues with Moq(version 4.2) Integration

Introduction

An alarming privacy issue has come to light involving the integration of Moq (version 4.2), a well-known mocking library for .NET projects, with SponsorLink, another project by the same creator. This problem has caused worries about user privacy and agreement.

Reportedly, this combination involves **looking through users' Git repositories to find email addresses, which are then connected to SponsorLink profiles.** Originally, this connection was meant to identify GitHub sponsors and offer them extra features. However, this has led to concerns about the possible privacy consequences linked to collecting emails.

The Nature and Goal of SponsorLink

SponsorLink, made by the same creator as Moq, aims to create a direct link between GitHub sponsors and creators of open-source libraries. Its main purpose is to give real benefits to sponsors, like unlocking advanced features or showing gratitude messages in development environments during the building process.

Privacy Concerns

The integration of SponsorLink into Moq has caused strong criticism due to worries about user privacy. The main issue here is whether the process of collecting emails respects user consent. Apparently, users need to install the SponsorLink GitHub app to prevent email collection. But this approach has been met with doubt in the developer community. Critics argue that these methods might not gather clear permission from users and could unintentionally gather sensitive information.

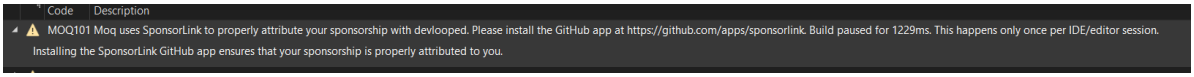
Solution

Considering Other Options: In response to this controversy, developers are actively looking into other mocking libraries, like "NSubstitute." These alternatives offer similar features without the perceived privacy worries. Thinking about these alternative libraries highlights how much trust impacts developers' choices of tools, which has broader effects for library creators.

This incident shows the importance of being clear and careful about user privacy, not just in this situation, but across the wider field of software development.

The flow sponsor link get email

1. **Library Installation**:
    - A user installs the Moq library (version 4.2.0) into their project using NuGet packages.
2. **Build Process**:
    - The user initiates the build process for their project.
    - During the build, the Moq library includes the Sponsor Link functionality, which aims to associate GitHub sponsors with project dependencies.



3. **Checking Git Email**:
    - The Sponsor Link functionality checks the user's Git configuration to extract the email address associated with the Git commits.
4. **Hashing Email Address**:
    - The extracted email address is hashed to create a unique identifier.
    - This hashed identifier is sent to a remote server as part of an HTTP request.

Problem here: sponsor link is closed source, so we don't know if it encrypted or salted the email after it did SHA256?. Since it's closed source, we can't know for sure everything it's doing.

refer link: (2) Does Moq in it's latest version extract and send my email to the cloud via SponsorLink? : dotnet (reddit.com)

5. **Remote Server Interaction**:
    - The remote server receives the hashed identifier from the Sponsor Link functionality.
    - The server checks whether the hashed identifier corresponds to a GitHub account that is sponsoring the project.
6. **Response and User Interaction**:
    - The remote server sends back a response based on the check:
        - If no matching sponsor account is found, the Sponsor Link functionality prompts the user to sign up with their sponsor-linking service.
        - If the user has an account but isn't sponsoring the dependency, the user is encouraged to sponsor the dependency/project.
        - If the user is already sponsoring, a congratulatory message is displayed.

Note: **Network Traffic and Firewall Detection**: The interaction with the remote server generates network traffic, which could be detected by firewalls in organizations with strict network security policies.

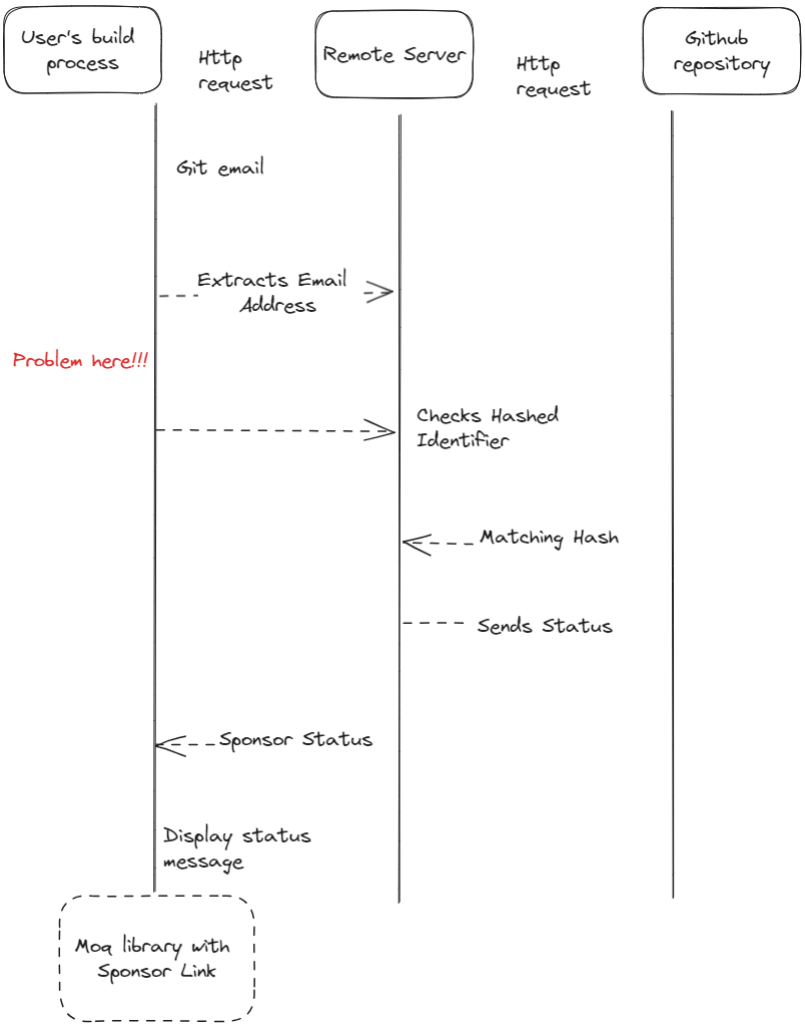


Diagram: flow sponsor working

- "Git Email" refers to the email address in the user's Git configuration.
- "Sponsor Status Message" refers to the response from the remote server, indicating sponsorship status.

Problem here!!! Sponsor link is closed source, so we don't know if it encrypted or salted the email after it did SHA256

Refer document

- Remove Moq From Your .NET Projects Right NOW! - YouTube
- Moq Uses Your Personal Information!!! - YouTube
- Warnings with latest version from SponsorLink · Issue #1370 · moq/moq (github.com)
- (2) Does Moq in it's latest version extract and send my email to the cloud via SponsorLink? : dotnet (reddit.com)
- SponsorLink bây giờ cũng là OSS · Vấn đề #1384 · MOQ / MOQ (github.com)