Tzuf Stokelman (Tabnine) <support@tabninedesk.zendesk.com>
to me

Jan 29, 2024, 10:31PM (11 days ago)

Your request (29376) has been updated. To add additional comments, reply to this email.

Tzuf Stokelman (Tabnine)
Jan 29, 2024, 13:31 GMT+2

Hi

Thanks for contacting us!

We are looking into it.

Best regards,

Tzuf Stokelman
Tabnine Support
tabnine

Daniel Ávila
to me

Hi

Apologies for the delay in my response. I was in the midst of a trip.

What you've shared is very interesting. It's incredible how they managed to obtain data from the tool by installing a fake extension.

We'll continue to enhance the CodeGPT code to make it increasingly secure.

I have a question, did you create the fake extension?

Thank you very much for providing this information. I've already forwarded it to the development team, and we're working on improving the upcoming versions.

Best regards,
Dani

--
Daniel Ávila Arias

Wimer Hazenberg
to me

1:35 AM

Hi

Thanks for your message. While you're correct that I store the license key and email in the global state, I don't store chat history. the SecretStorage could be a solution, thanks. Also, I don't see any video?

Thanks,
Wimer.

Nadav Cohen <nc.amaa@gmail.com>
to me

Dear

Thank you for your detailed analysis and for bringing these potential security issues to our attention. We appreciate the efforts of your research team in enhancing the security of the VSCode ecosystem.

We understand the importance of securely handling credentials within our extension, CodamaSoftware.ai-docs-and-comments, and are committed to implementing the necessary measures to protect our users' data. To effectively address the concerns you've raised, could you please provide us with more specific instructions and, if possible, code examples highlighting the best practices for mitigating these vulnerabilities? This information will be invaluable in guiding our development team to promptly and accurately rectify any security flaws.

In particular, we are interested in:
- Examples of secure methods for storing and retrieving API keys within the VSCode extension environment.
- Recommended practices for handling user inputs that involve sensitive information.
- Any suggestions on libraries or tools that could facilitate the secure management of credentials.

We are eager to review and improve our extension's security features and look forward to your expert advice on these matters.

Please let us know if you need any additional information from our side or if there are any specific areas within our extension that we should focus on. Your guidance is crucial for us to ensure the highest level of security and user trust.

Thank you once again for your valuable feedback and support.

Best regards,

Nadav Cohen
CEO, Codama.dev

Maik Michel
to me

Thank you for analysing, I will investigate further.
Most of the bullets seem irrelevant, but I think we check the one pointing to the use of Global State when persisting secrets.

Best regards,
Maik Michel

•••

Yue Huang
to me

Hi

Thanks. I'll take a look at it.

Is there any recommended practice for storing OpenAI key for vscode extensions now?

•••

John Iwasz
to me

Thanks for your analysis and including this extension in your research scope.

The AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY environment variables are used by the AWS CLI per this document:

Environment variables to configure the AWS CLI - AWS Command Line Interface (amazon.com)

If an AWS user opts for these environment settings as is often the case on development environments, then the extension automatically obtains the credentials from the same environment variables used by the AWS CLI.

You do raise a fair point about encrypting entered user credentials and that's something to consider for a future version. Since the credentials need to be supplied as plain text to an AWS REST call, the extension can't hash the credentials.

Thanks,
John

Albert Hendriks
to me

Dear

Thanks for reaching out. You are correct that this is (technically) a security flaw. Please allow me to give some background information.
The extension functions by making calls to our WordPress backend. To secure our API, our users need to create an application password, which is indeed stored in vscode in plain text. We're using application passwords in a similar way as API keys (maybe there isn't even much difference between these two generally). This issue is not unique to our extension; it's a challenge faced industry-wide, as evidenced by common practices around API key storage. However, simply encrypting these credentials leads to the question of secure encryption key management, which is equally challenging.
Our business focuses on software quality, so if possible we'd like to do this more securely. But at the same time, we want to avoid barriers to our first users, as we're a new company. If you have more recommendations or insights, I would greatly appreciate your input.

Best regards,
Albert Hendriks, MSc
Soliditai.com

•••

Marcello Urbani
to me

Thanks for contacting me
I might have erred on the way of ergonomics, but:

- storing passwords in configuration is optional - and not recommended. I always assumed this was clear to users
- the extension does store passwords, but uses the vscode provided interface to the os password safe. I doubt I can do better without asking users to type their pwd every time
- what's the issue with input box? How else could I prompt users for passwords?
- pretty sure I don't store any sensitive information in global state

Cheers
Marcello

Aram Becker Thu, 8 Feb,
to me

Thank you for the heads up! This was initially an internally used extension, but I decided to publish it after we faded out the use of it.

To address the security concerns, I have:

- marked the token configuration as deprecated and added a warning message if it's set
- moved the token storage into SecureStorage
- completely removed a private token inside the settings.json throughout the git history
- updated all dependencies to the newest version matching the used vscode engine

I could not remove the input box, as this is required to initialize the token.

Just in case email responses do get evaluated, the corresponding commit is https://github.com/1nVitr0/plugin-vscode-daily-timelog/pull/13/commits/04b0ce969a19a3c00aa24825325d51acc80a04cb

Thanks again, glad you exist!

Aram Becker

1Password Support
to me

Thank you for informing us about the potential security risk in 1Password VS code extension. Upon receiving your email, I informed our Security team regarding your finding for investigation, and someone will be with you as soon as possible.

Best regards,

Nhat Nguyen.
https://support.1password.com/

Nikita Fedkin
to me

Hello,

Thanks for your findings. What is your suggestion to improve the configuration?

Kind regards,
Nikita Fedkin.

ср, 7 февр. 2024 г., 20:11 Yue <report2@vsanalysis.net>:

Martin Picard
to me

Dear

Thank you for bringing this to my attention, I was not aware of the issue, as I did not write most of this code, it is a fork of another repository https://github.com/42Crunch/vscode-openapi, they might also have this security issue. I will look into refactoring the code ASAP.

It looks like VSCode already has an API for storing extension secrets https://dev.to/kompotkot/how-to-use-secretstorage-in-your-vscode-extensions-2hco

Best regards,

Martin Picard
Lead Consultant

一ノ瀬裕斗
to me

Thank you very much.

We have taken steps to make this extension private.

Next, I would like to change the API key so that it is entered each time the process is executed instead of storing it in the settings screen, is this a problem?

Thank you in advance.

Microsoft Security Response Center
to Microsoft, me

Thu, Feb 1, 5:52 AM (8 days ago)

Hello,

Thank you again for submitting this issue to Microsoft. Currently, MSRC prioritizes vulnerabilities that are assessed as "Important" or "Critical' severities for immediate servicing. After careful investigation, this does not meet MSRC's bar for immediate servicing due to this not representing breaches of security. There is no security sandbox for running extensions, they are available to use as desired as long as the OS user permissions allow them to. However, we have shared the report with the team responsible for maintaining the product or service.

Here is some information on Microsoft's security vulnerability servicing criteria that may help you in your future research:

- For Online Services Vulnerability Research: Microsoft Vulnerability Severity Classification for Online Services
- For Windows Vulnerability Research: Microsoft Security Servicing Criteria for Windows
- For Bounty information and current programs: Microsoft Bounty Programs | MSRC

Thank you for your submission. We appreciate your partnership to help secure our customers.

Regards,
Alexus
MSRC