"How did that get into the repo?"

DISPATCH - IETF103, Bangkok

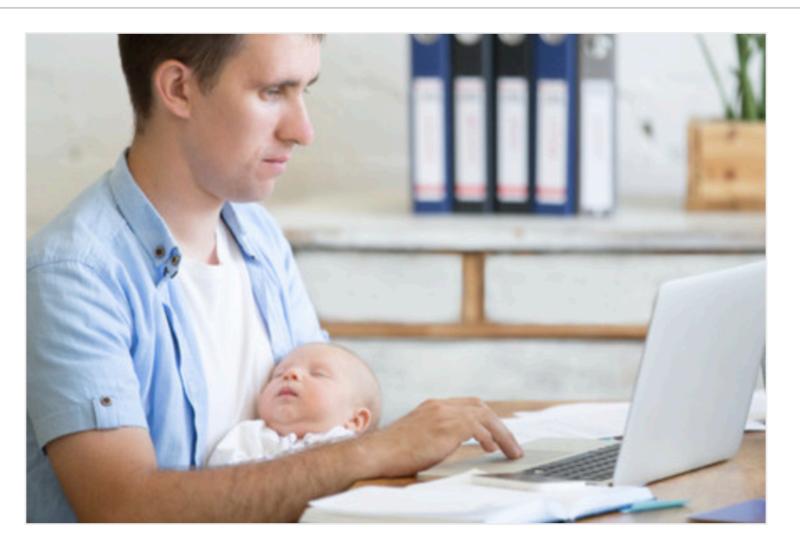
Software

Researcher found Homebrew GitHub token hidden in plain sight

'Kid, take a nap, I have a project to save'

By Richard Chirgwin 8 Aug 2018 at 09:38

11 🖵 SHARE ▼



The popular Homebrew macOS package installer has moved to plug a serious vulnerability – it accidentally left a GitHub token visible to the public. Luckily, a team member on paternity leave had a moment while their child napped to fix it. 2

Repositories	97	Showing 640,102 available commit results ③			Sort: Best n	natch -
Code	16M+					
Commits	640K+	Removed token. Not a good token, but better to delete.	Verified	Ê	f77d8a0	<>
Issues	112K	salsalabs committed to salsalabs/engage_api_csharp				
Marketplace		on Sep 20				
Topics						
Wikis	8K	[Feature #159965442] Remove the hardcoded token and get the tokens fr		Ê	878b12a	<>
Users		Quantum-35 committed to andela/ah-				
		aniellau415 committed to daniellau415/RandomFood on Sep 28				
		removed token		Ê	19ec102	<>
		aniellau415 committed to daniellau415/RandomFood on Sep 28				
		removed authentication token for user profile		Ê	19eddc3	
						<>
		McHardex authored and McHardex committed to McHardex/newsRestApi 20 days ago				\diamond

truffleHog

Searches through git repositories for secrets, digging deep into commit history and branches. This is effective at finding secrets accidentally committed.

NEW

truffleHog previously functioned by running entropy checks on git diffs. This functionality still exists, but high signal regex checks have been added, and the ability to surpress entropy checking has also been added.

These features help cut down on noise, and makes the tool easier to shove into a devops pipeline.

truffleHog --regex --entropy=False https://github.com/dxa4481/truffleHog.git

or

truffleHog file:///user/dxa4481/codeprojects/truffleHog/

```
Date: 2014-04-21 18:46:21
Branch: master
Commit: Removing aws keys
@@ -57,8 +57,8 @@ public class EurekaEVCacheTest extends AbstractEVCacheTest {
```

About Token Scanning

GitHub scans public repositories for known token formats, to prevent fraudulent use of credentials that were committed accidentally.

Note: Token Scanning is currently in beta and subject to change.

Token Scanning for public repositories

When you push commits to a public repository, or switch a private repository to public, GitHub scans the contents of the commits or repository for tokens issued by the following service providers:

- Amazon Web Services (AWS)
- Azure
- > GitHub
- Google Cloud
- Slack
- Stripe

When GitHub detects a set of credentials, we notify the service provider who issued the token. The service provider may revoke the token, issue a new token, or reach out to you directly.



Probably stupid idea: what if we created a new uri scheme for access tokens, and then GitHub refused to check in textual content that contains it?

 \sim

7:55 AM - 9 Aug 2018

Network Working Group Internet-Draft Intended status: Standards Track Expires: March 1, 2019

The secret-token URI Scheme

draft-nottingham-how-did-that-get-into-the-repo-01

Abstract

This document registers the "secret-token" URI scheme, to aid in the identification of authentication tokens.

Note to Readers

RFC EDITOR: please remove this section before publication

The issues list for this draft can be found at https://github.com/mnot/I-D/labels/how-did-that-get-into-the-repo.

The most recent (often, unpublished) draft is at https://mnot.github.io/I-D/how-did-that-get-into-the-repo/.

Recent changes are listed at https://github.com/mnot/I-D/commits/gh-pages/how-did-that-get-into-the-repo.

See also the draft's current status in the IETF datatracker, at https://datatracker.ietf.org/doc/draft-nottingham-how-did-that-get-into-the-repo/.

secret-token:E92FB7EB-D882-47A4-A265-A0B6135DC842

- Good enough to try?
- AD-Sponsored? WG? Independent Submission?
- Permanent? Provisional?