Developers can’t seem to stop exposing credentials in publicly accessible code

Victor De Schwanberg/Science Photo Library via Getty Images

reader comments

33 with

Despite more than a decade of reminding, prodding, and downright nagging, a surprising number of developers still can’t bring themselves to keep their code free of credentials that provide the keys to their kingdoms to anyone who takes the time to look for them.

The lapse stems from immature coding practices in which developers embed cryptographic keys, security tokens, passwords, and other forms of credentials directly into the source code they write. The credentials make it easy for the underlying program to access databases or cloud services necessary for it to work as intended. I published one such PSA in 2013 after discovering simple searches that turned up dozens of accounts that appeared to expose credentials securing computer-to-server SSH accounts. One of the credentials appeared to grant access to an account on Chromium.org, the repository that stores the source code for Google’s open source browser.

In 2015, Uber learned the hard way just how damaging the practice can be. One or more developers for the ride service had embedded a unique security key into code and then shared that code on a public GitHub page. Hackers then copied the key

and used it to access an internal Uber database and, from there, steal sensitive data belonging to 50,000 Uber drivers.

Uber lawyers argued at the time that “the contents of these internal database files are closely guarded by Uber,” but that contention is undermined by means the company took in safeguarding the data, which was no better than stashing a house key under a door mat.

The number of studies published since following the revelations underscored just how common the practice had been and remained in the years immediately following Uber’s cautionary tale. Sadly, the negligence continues even now.

Researchers from security firm GitGuardian this week reported finding almost 4,000 unique secrets stashed inside a total of 450,000 projects submitted to PyPI, the official code repository for the Python programming language. Nearly 3,000 projects contained at least one unique secret. Many secrets were leaked more than once, bringing the total number of exposed secrets to almost 57,000.

“Exposing secrets in open-source packages carries significant risks for developers and users alike,” GitGuardian researchers wrote. “Attackers can exploit this information to gain unauthorized access, impersonate package maintainers, or manipulate users through social engineering tactics.”

Advertisement

The credentials exposed provided access to a range of resources, including Microsoft Active Directory servers that provision and manage accounts in enterprise networks, OAuth servers allowing single sign-on, SSH servers, and third-party services for customer communications and cryptocurrencies. Examples included:

- Azure Active Directory API Keys
- GitHub OAuth App Keys
- Database credentials for providers such as MongoDB,

MySQL, and PostgreSQL
- Dropbox Key
- Auth0 Keys
- SSH Credentials
- Coinbase Credentials
- Twilio Master Credentials.

Also included in the haul were API keys for interacting with various Google Cloud services, database credentials, and tokens controlling Telegram bots, which automate processes on the messenger service. This week's report said that exposures in all three categories have steadily increased in the past year or two.

The secrets were exposed in various types of files published to PyPI. They included primary .py files, README files, and test folders.

Enlarge / Most common types of files other than .py containing a hardcoded secret in PyPI packages.
GitGuardian
GitGuardian tested the exposed credentials and found that 768 remained active. The risk, however, can extend well beyond that smaller number. GitGuardian explained:

> *It is important to note that just because a credential can not be validated does not mean it should be considered invalid. Only once a secret has been properly rotated can you know if it is invalid. Some types of secrets GitGuardian is*

> *still working toward automatically validating include Hashicorp Vault Tokens, Splunk Authentication Tokens, Kubernetes Cluster Credentials, and Okta Tokens.*

There are no good reasons to expose credentials in code. The report said the most common cause is by accident.

“In the course of outreach for this project, we discovered at least 15 incidents where the publisher was unaware they had made their project public,” the authors wrote. “Without naming any names, we did want to mention some of these were from very large companies that have robust security teams. Accidents can happen to anyone.”

Over the past decade, various mechanisms have become available for allowing code to securely access databases and cloud resources. One is .env files that are stored in private environments outside of the publicly available code repository. Others are tools such as the AWS Secrets Manager, Google Cloud’s Secret Manager, or the Azure Key Vault. Developers can also employ scanners that check code for credentials inadvertently included.

The study examined PyPI, which is just one of many open source repositories. In years past, code hosted in other repositories such as NPM and RubyGems has also been rife with credential exposure, and there’s no reason to suspect the practice doesn’t continue in them now.

https://arstechnica.com/?p=1984368