

# Keeping Credentials Safe in the GitHub Age

Matt Bacchi

Boulder Python

May 12, 2020

# About Me

Sr. Devops Engineer  
Brave Software

# What are Credentials?

API Tokens

Access Keys / Secret Keys

# Why is it important to keep credentials secure?

- Losing control of your project
- Nefarious use of your resources
- Soaring cloud costs
- Botnets on your servers
- Company reputation damaged

# Examples of credential leaks

# PyPI .pypirc security vulnerability

Source: [https://python-security.readthedocs.io/pypi-vuln/index-2017-11-08-pypirc\\_exposure\\_on\\_github.html](https://python-security.readthedocs.io/pypi-vuln/index-2017-11-08-pypirc_exposure_on_github.html)

# npm package vulnerabilities

ALL POSTS / COMMUNITY

## A High Level Post Mortem of the eslint-scope Security Incident

by:  Tierney Cyren in Community on Jul 12 2018

Security

npm

JavaScript

Packages

SHARE



Early today there was an incident regarding a module (`eslint-scope`) that was hijacked on npm. The hijacked module attempted to steal tokens contained within `.npmrc` to enable additional publishes. Here's a quick overview of everything that's happened, based on the information available so far.

\*This post will be updated as more information becomes available. Feel free to ping me on Twitter with additional information, I'd be happy to make updates/corrections as necessary. \*

This has been totally from my point of view while observing and reaching out to individuals who I know it matters to. I figured I'd take a minute to summarize the entire event and give a quick post-mortem for y'all in case you don't have as much time to investigate and work on this stuff as I do.

### Module Hijacking

The hijacker of `eslint-scope` published a new patch release of the module at 10:40 UTC. Normally, a new patch would automatically hit many of the individuals using the module (either directly or as a dependency of another module).

Today at 11:17 UTC, an issue was created in the GitHub repository of `eslint-scope` regarding unexpected error messages which

Source: <https://nodesource.com/blog/a-high-level-post-mortem-of-the-eslint-scope-security-incident>

How widespread is this problem?



# 2019 public repository research paper findings

*“We find that not only is secret leakage pervasive — affecting over 100,000 repositories — but that thousands of new, unique secrets are leaked every day.”*

- Meli, McNiece, Reaves

Source: [https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019\\_04B.3\\_Meli\\_paper.pdf](https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_04B.3_Meli_paper.pdf)

Over 100,000  
repositories affected...

Source: [https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019\\_04B-3\\_Meli\\_paper.pdf](https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_04B-3_Meli_paper.pdf)

How much can it cost to accidentally  
release credentials?

# 2017 DXC leak: the \$64,000 question

**The Register**  
Biting the hand that feeds IT

DATA CENTRE SOFTWARE SECURITY DEVOPS BUSINESS PERSONAL TECH SCIENCE EMERGENT TECH BOOTNOTES

**Business**  
**DXC spills AWS private keys on public GitHub**  
'Unknown persons' spin up 244 VMs at cost of \$64k. Whoops  
By Paul Kunert 14 Nov 2017 at 13:08 49 SHARE ▼



Miscreants racked up a \$64,000 bill on DXC Technologies' tab after a techie accidentally uploaded the outsourcing firm's private AWS key to a public GitHub repo.  
  
It was red faces all round as the business opened up on the classic crypto key fumble in a PDF memo to staff, the contents of which were seen by *The Register*.

**// MOST READ**  
 If you're Python, and PH trendy l miles b says  
 Maersk the Mai who res NotPety  
 Microsc back on are strip 10  
 We reg there at the toke nerds b bloody  
 Window screen passers upgrade  
**SUBSCRIBE TO OUR TECH NEWSLETTER**

Source: [https://www.theregister.co.uk/2017/11/14/dxc\\_github\\_aws\\_keys\\_leaked/](https://www.theregister.co.uk/2017/11/14/dxc_github_aws_keys_leaked/)

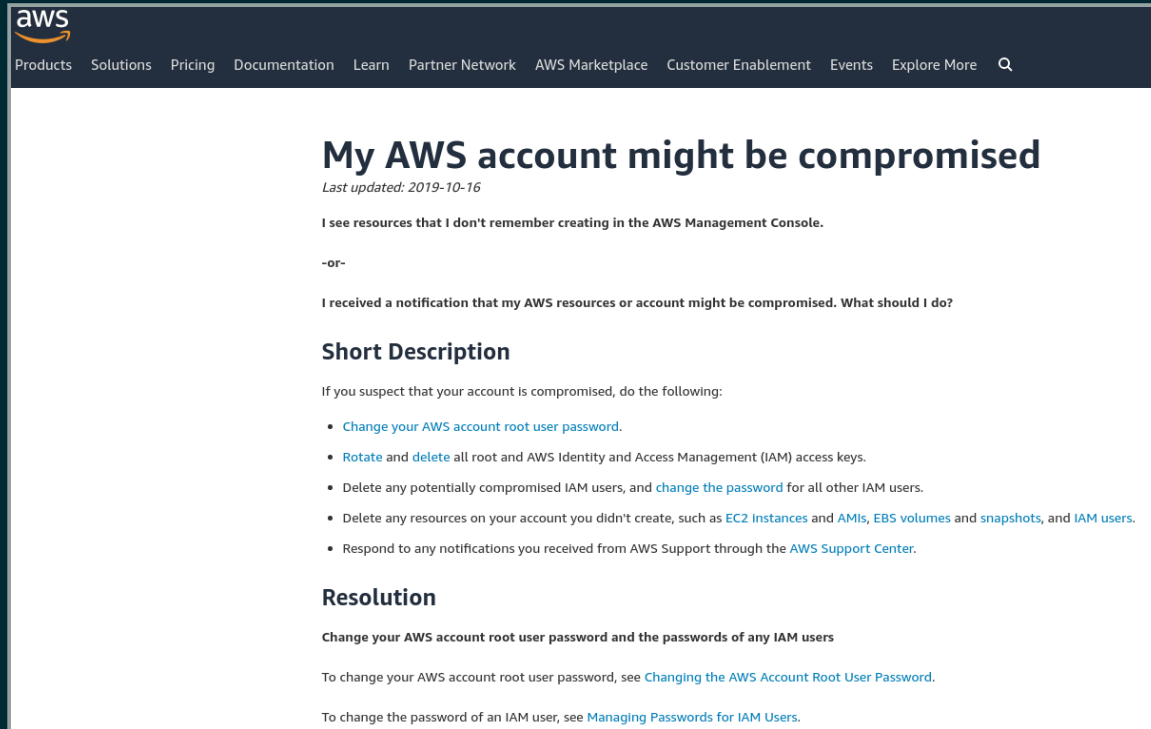
A man in a white shirt and striped tie sits at a desk, looking shocked with his mouth wide open. He is gesturing with his hands while talking to another person whose back is to the camera. The scene is set in an office with a computer monitor visible in the background.

\$64,000 ...

in less than 24 hours

How to respond if this happens

# AWS steps to mitigate a compromise



The screenshot shows the AWS Knowledge Center page for the article "My AWS account might be compromised". The page has a dark blue header with the AWS logo and navigation links: Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More, and a search icon. The main content area is white and contains the following text:

## My AWS account might be compromised

*Last updated: 2019-10-16*

I see resources that I don't remember creating in the AWS Management Console.

-or-

I received a notification that my AWS resources or account might be compromised. What should I do?

### Short Description

If you suspect that your account is compromised, do the following:

- [Change your AWS account root user password.](#)
- [Rotate](#) and [delete](#) all root and AWS Identity and Access Management (IAM) access keys.
- Delete any potentially compromised IAM users, and [change the password](#) for all other IAM users.
- Delete any resources on your account you didn't create, such as [EC2 instances](#) and [AMIs](#), [EBS volumes](#) and [snapshots](#), and [IAM users](#).
- Respond to any notifications you received from AWS Support through the [AWS Support Center](#).

### Resolution

Change your AWS account root user password and the passwords of any IAM users

To change your AWS account root user password, see [Changing the AWS Account Root User Password](#).

To change the password of an IAM user, see [Managing Passwords for IAM Users](#).

Source: <https://aws.amazon.com/premiumsupport/knowledge-center/potential-account-compromise>

# In a nutshell

- Change AWS root account password, all user passwords
- Rotate AWS access keys
- Delete any compromised user accounts
- Delete any unrecognizable user accounts
- Delete any unrecognizable resources (EC2 instances, S3 buckets, Lambda functions)
- Contact AWS support (they may have contacted you)



Securing your mission critical secrets  
can be difficult

(But you can make the job easier with tools)

*“Prevention is better than cure.”*

Desiderius Erasmus

# Secret Leak Prevention

Content based on my [blog post](#) and [Github repository](#)

# Leak prevention methods

- Git pre-commit hooks
- AWS Labs git-secrets
- python-git-secrets
- IAM roles

# Using the Git pre-commit hook

Demo

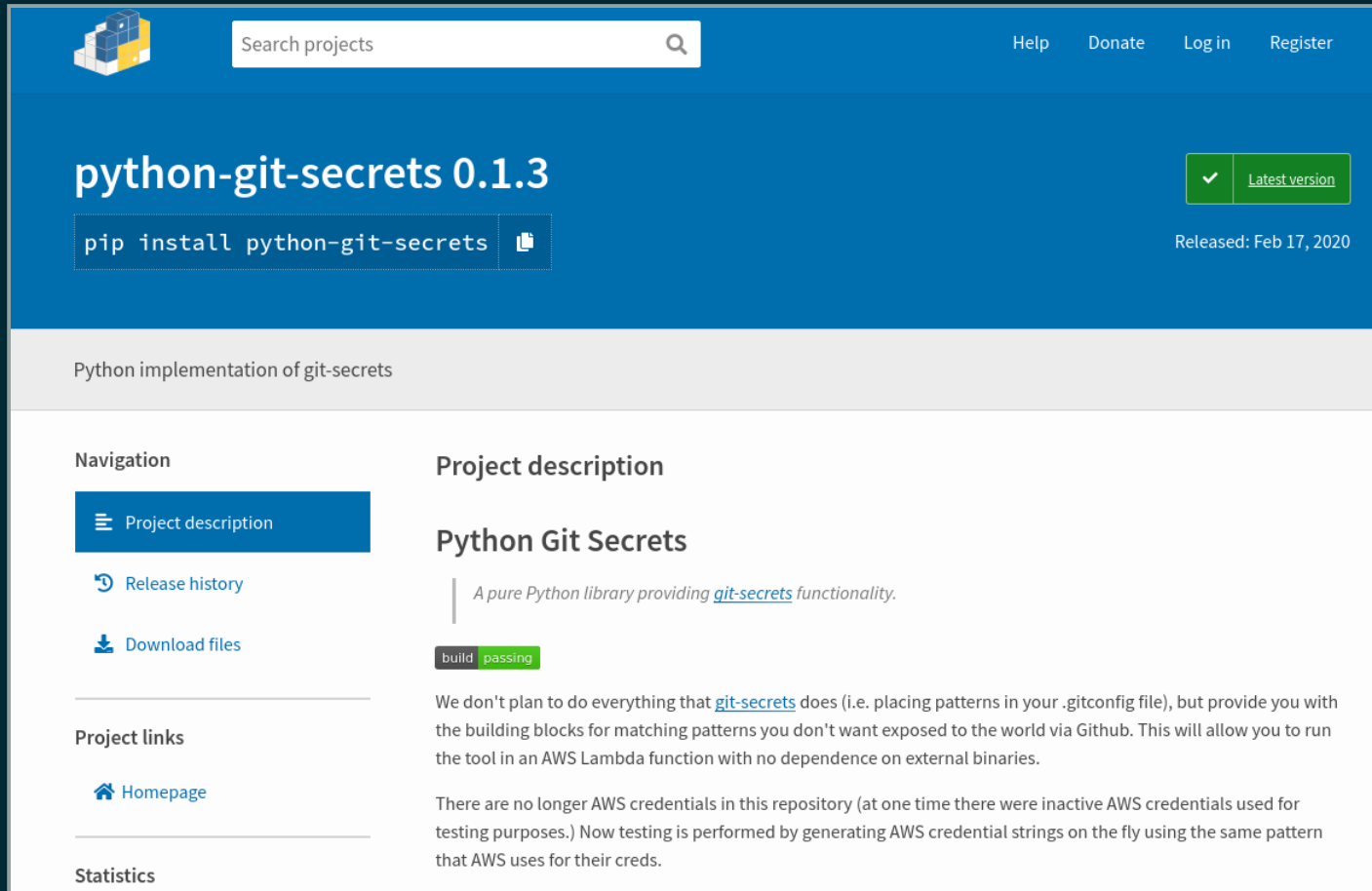
AWS Labs git-secrets

Demo



python-git-secrets Python package

# python-git-secrets available via PyPI



The screenshot shows the PyPI project page for **python-git-secrets 0.1.3**. The header includes a search bar, navigation links (Help, Donate, Log in, Register), and a status bar indicating it is the **Latest version** (checked) and was released on **Feb 17, 2020**. The main content area features the command `pip install python-git-secrets` and a brief description: "Python implementation of git-secrets". The left sidebar contains navigation links: "Project description" (selected), "Release history", and "Download files". The right section, titled "Project description", includes a sub-header "Python Git Secrets" and a tagline: "A pure Python library providing [git-secrets](#) functionality." Below this, a "build passing" badge is shown. The description text states: "We don't plan to do everything that [git-secrets](#) does (i.e. placing patterns in your .gitconfig file), but provide you with the building blocks for matching patterns you don't want exposed to the world via Github. This will allow you to run the tool in an AWS Lambda function with no dependence on external binaries." A final paragraph notes: "There are no longer AWS credentials in this repository (at one time there were inactive AWS credentials used for testing purposes.) Now testing is performed by generating AWS credential strings on the fly using the same pattern that AWS uses for their creds."

Source: <https://pypi.org/project/python-git-secrets> & <https://github.com/mbacchi/python-git-secrets>

Demo

# Using IAM Roles

# AWS IAM and IAM Roles

- Identity and Access Management
- Provides the ability to assume a role
- EC2 instance can assume a role to interact with other AWS services
- Access Key/Secret Access Key not needed on EC2 instance

# EC2 Role with access to S3 bucket

Roles > terraform-20200308171641998100000001

## Summary

Role ARN	arn:aws:iam::592431548397:role/terraform-20200308171641998100000001
Role description	<a href="#">Edit</a>
Instance Profile ARNs	arn:aws:iam::592431548397:instance-profile/demo_instance_profile
Path	/
Creation time	2020-03-08 11:16 MDT
Last activity	Not accessed in the tracking period
Maximum CLI/API session duration	1 hour <a href="#">Edit</a>

[Permissions](#) [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

▼ Permissions policies (1 policy applied)

[Attach policies](#)

Policy name ▼	Policy type
▼ terraform-202003081716487280000000002	Inline policy

[Policy summary](#) [{} JSON](#) [Edit policy](#)

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Action": [
6         "s3:*"
7       ],
8       "Effect": "Allow",
9       "Resource": [
10        "arn:aws:s3::s3-bucket-test-ec2-assumerole-9so-uvnz",
11        "arn:aws:s3::s3-bucket-test-ec2-assumerole-9so-uvnz/*"
```

# EC2 Role trust relationship

[Roles](#) > terraform-20200308171641998100000001

## Summary

Role ARN	arn:aws:iam::592431548397:role/terraform-20200308171641998100000001
Role description	<a href="#">Edit</a>
Instance Profile ARNs	arn:aws:iam::592431548397:instance-profile/demo_instance_profile
Path	/
Creation time	2020-03-08 11:16 MDT
Last activity	Not accessed in the tracking period
Maximum CLI/API session duration	1 hour <a href="#">Edit</a>

[Permissions](#) [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

You can view the trusted entities that can assume the role and the access conditions for the role. [Show policy document](#)

[Edit trust relationship](#)

### Trusted entities

The following trusted entities can assume this role.

**Trusted entities**

The identity provider(s) ec2.amazonaws.com

### Conditions

The following conditions define

There are no conditions associa

Demo



# Thank you!

QR code for <https://github.com/mbacchi>

# Resources and Links



- How Bad Can it Git? research paper
- Secret Leak Prevention Demo (shameless plug...)
- 3 Ways to Prevent Secret Leaks [blog post](#) (shameless plug...)
- python-git-secrets on [GitHub](#), [PyPi](#) (shameless plug...)