black hat EUROPE 2020

DECEMBER 9–10

ARSENAL

Git Wild Hunt

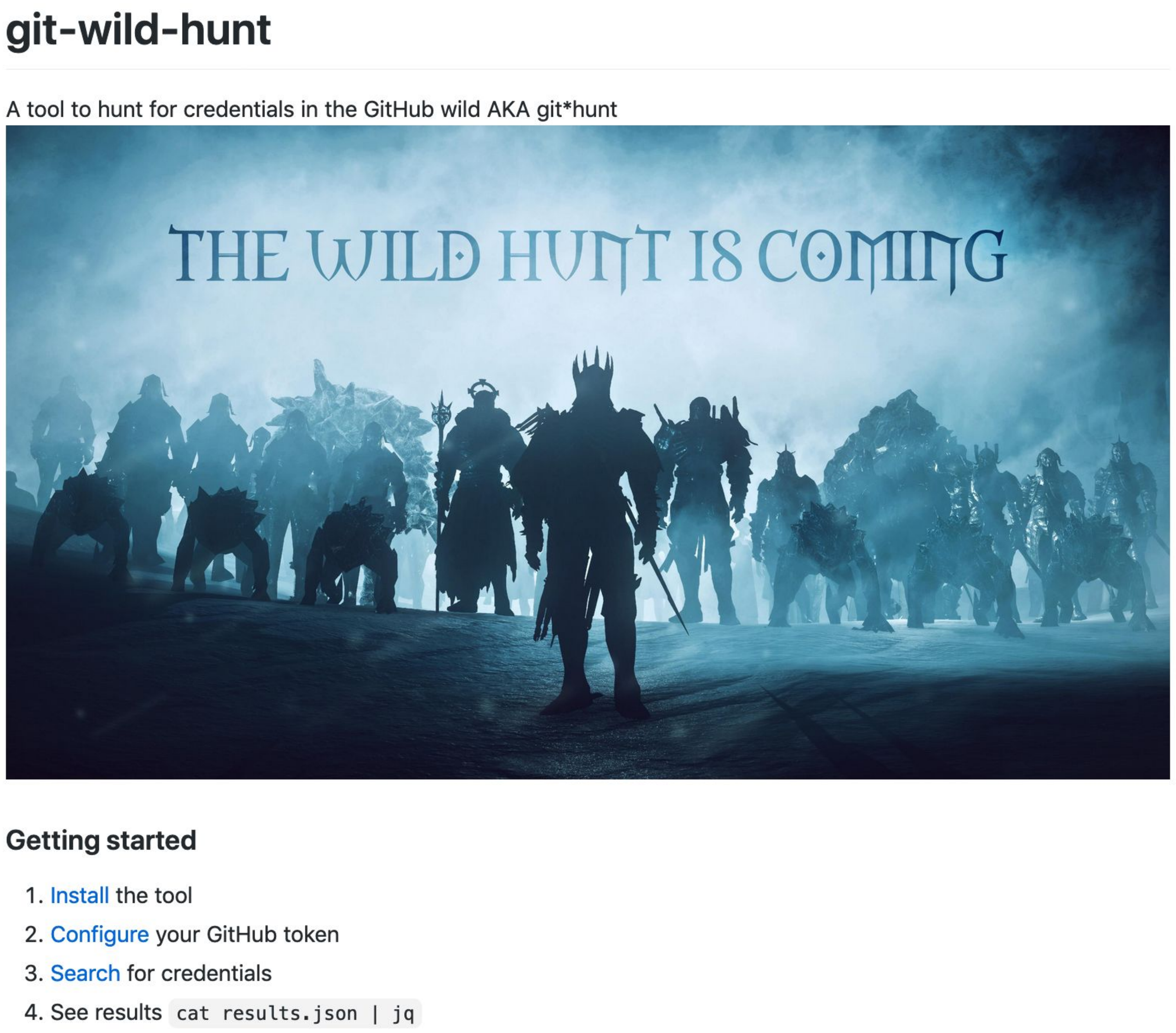
Jose Hernandez
Rod Soto

Whoami

José Hernandez @d1vious
Principal Security Researcher at Splunk. He started his professional career at Prolexic Technologies (now Akamai), fighting DDOS attacks against Fortune 100 companies perpetrated by “anonymous” and “lulzsec.” As an engineering co-founder of Zenedge Inc. (acquired by Oracle Inc.), José helped build technologies to fight bots and web-application attacks. He has also built security operation centers and run a public threat-intelligence service.

Rod Soto @rodsoto
Principal Security Research Engineer at Splunk. Worked at Prolexic Technologies (now Akamai), and Caspida. Cofounder of Hackmiami and Pacific Hackers meetups and conferences. Creator of Kommand && KonTroll / NoQrtr-CTF.

What Is Git Wild Hunt



git-wild-hunt

A tool to hunt for credentials in the GitHub wild AKA git*hunt

Getting started

1. Install the tool
2. Configure your GitHub token
3. Search for credentials
4. See results `cat results.json | jq`

Code bits 💻

`git clone https://github.com/d1vious/git-wild-hunt`

Installation

requirements `virtualenv, python3`

download project `git clone https://github.com/d1vious/git-wild-hunt && cd git-wild-hunt`

create virtualenv and install requirements `pip install virtualenv && virtualenv -p python3 venv && source venv/bin/activate && pip install -r requirements.txt`

Continue to configuring a GitHub API key

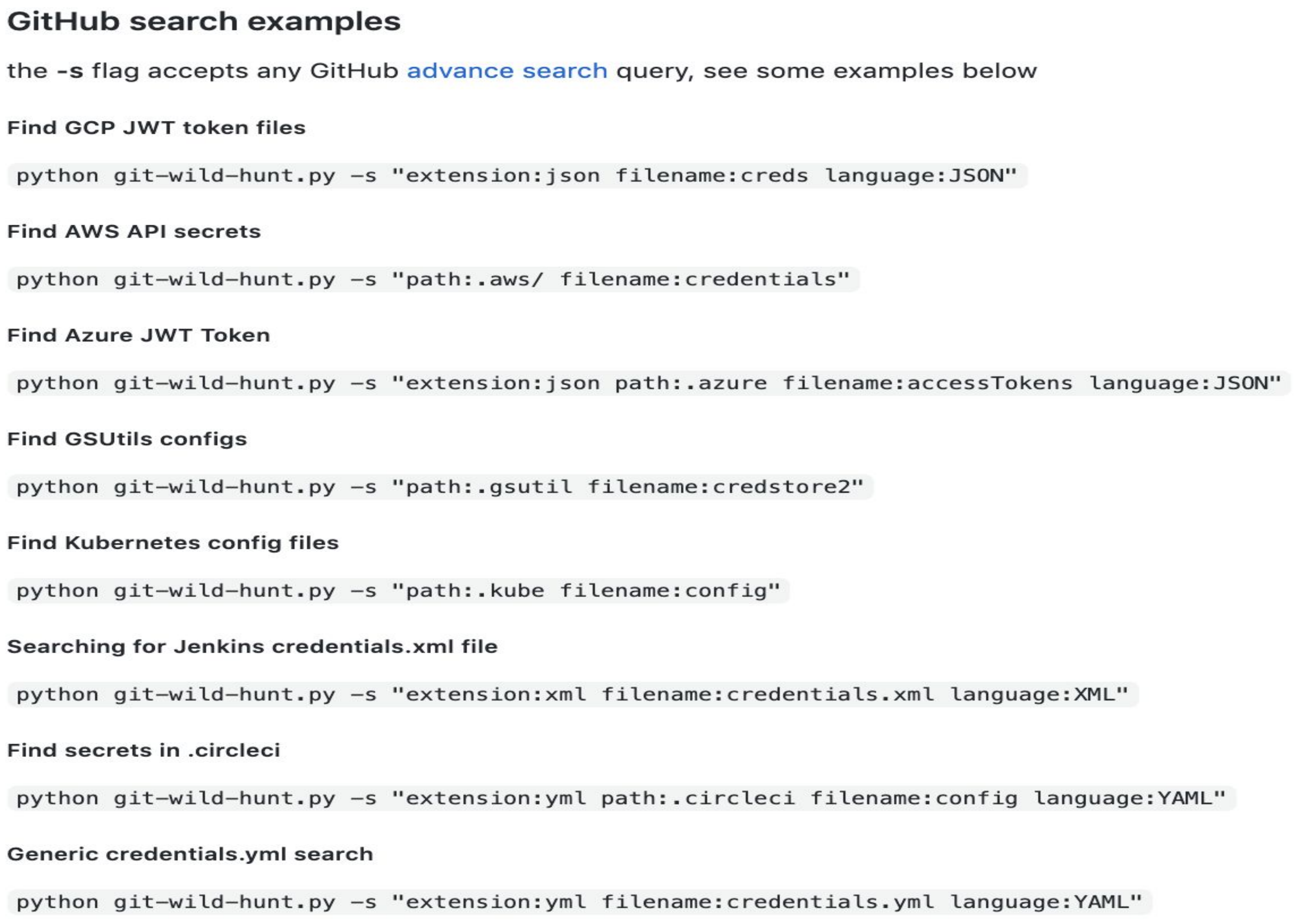
Configuration

Under <project_name>/git-wild-hunt.conf

Make sure you set a GitHub token if you need to create one for your account follow these instructions.

```
[global]
github_token = TOKENHERE
# github token for searching

output = results.json
# stores matches in JSON here

log_path = git-wild-hunt.log
# Sets the log_path for the logging file

log_level = INFO
# Sets the log level for the logging
# Possible values: INFO, ERROR

regexes = regexes.json
# regexes to check the git wild hunt search against
```

Example Searches



GitHub search examples

the **-s** flag accepts any GitHub advance search query, see some examples below

Find GCP JWT token files

```
python git-wild-hunt.py -s "extension:json filename:creds language:JSON"
```

Find AWS API secrets

```
python git-wild-hunt.py -s "path:.aws/ filename:credentials"
```

Find Azure JWT Token

```
python git-wild-hunt.py -s "extension:json path:.azure filename:accessTokens language:JSON"
```

Find GSUtils configs

```
python git-wild-hunt.py -s "path:.gsutil filename:credstore2"
```

Find Kubernetes config files

```
python git-wild-hunt.py -s "path:.kube filename:config"
```

Searching for Jenkins credentials.xml file

```
python git-wild-hunt.py -s "extension:xml filename:credentials.xml language:XML"
```

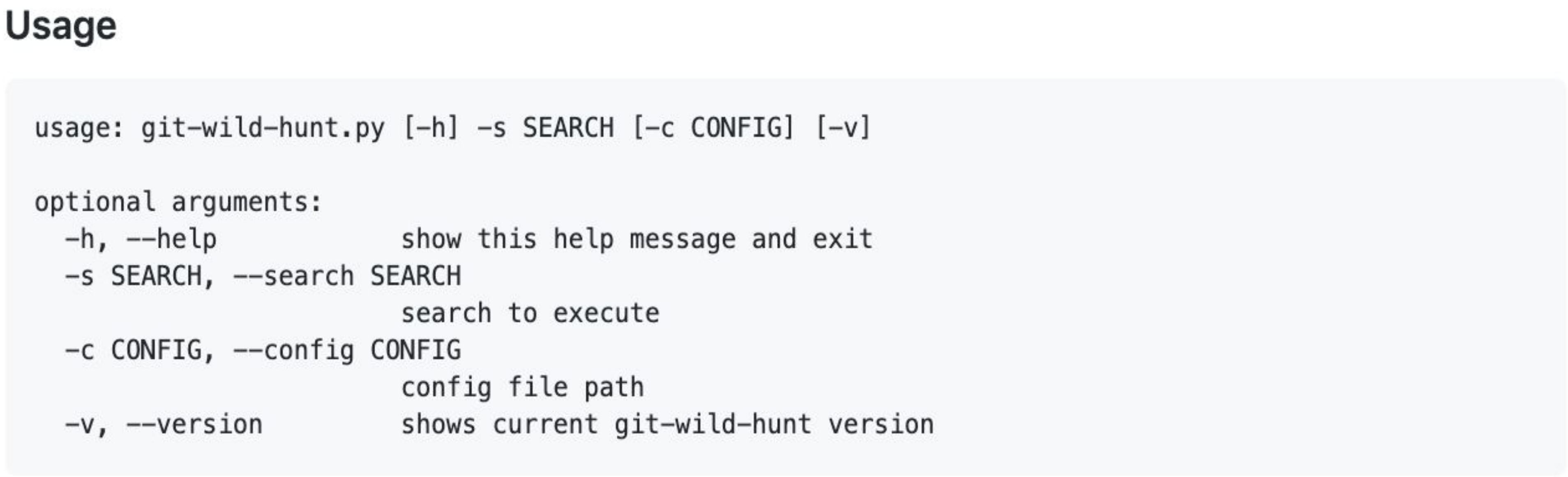
Find secrets in .circleci

```
python git-wild-hunt.py -s "extension:yml path:.circleci filename:config language:YAML"
```

Generic credentials.yml search

```
python git-wild-hunt.py -s "extension:yml filename:credentials.yml language:YAML"
```

How To Use It

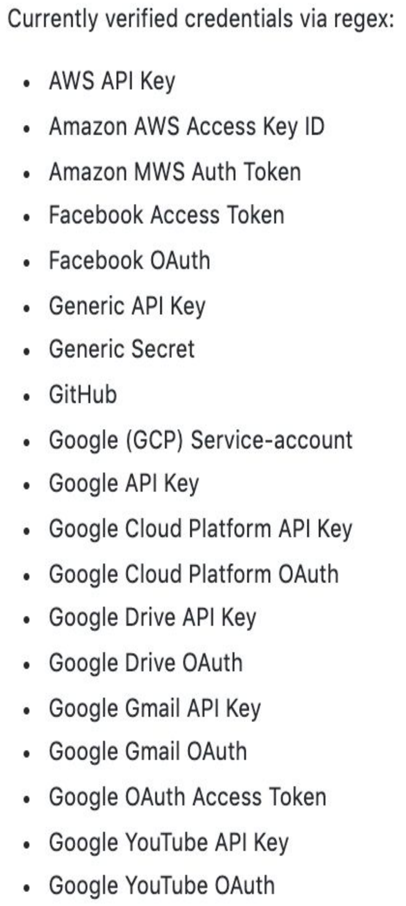


Usage

```
usage: git-wild-hunt.py [-h] -s SEARCH [-c CONFIG] [-v]

optional arguments:
  -h, --help            show this help message and exit
  -s SEARCH, --search SEARCH
                        search to execute
  -c CONFIG, --config CONFIG
                        config file path
  -v, --version         shows current git-wild-hunt version
```

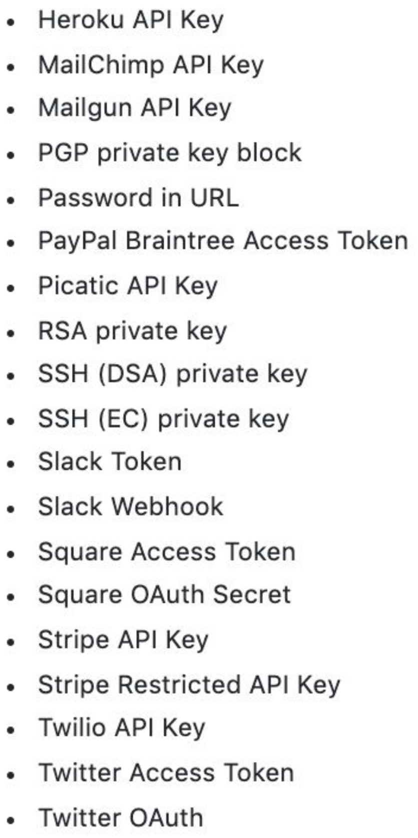
black hat EUROPE 2020

Regexes

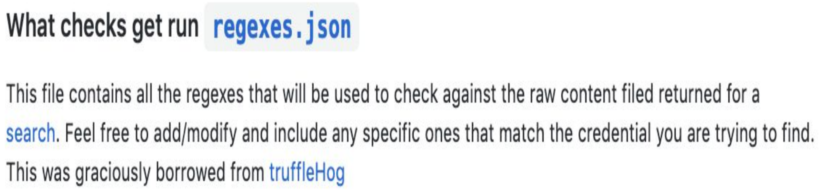


Currently verified credentials via regex:

- AWS API Key
- Amazon AWS Access Key ID
- Amazon MWS Auth Token
- Facebook Access Token
- Facebook OAuth
- Generic API Key
- Generic Secret
- GitHub
- Google (GCP) Service-account
- Google API Key
- Google Cloud Platform API Key
- Google Cloud Platform OAuth
- Google Drive API Key
- Google Drive OAuth
- Google Gmail API Key
- Google Gmail OAuth
- Google OAuth Access Token
- Google YouTube API Key
- Google YouTube OAuth



- Heroku API Key
- MailChimp API Key
- Mailgun API Key
- PGP private key block
- Password in URL
- PayPal Braintree Access Token
- Picatic API Key
- RSA private key
- SSH (DSA) private key
- SSH (EC) private key
- Slack Token
- Slack Webhook
- Square Access Token
- Square OAuth Secret
- Stripe API Key
- Stripe Restricted API Key
- Twilio API Key
- Twitter Access Token
- Twitter OAuth

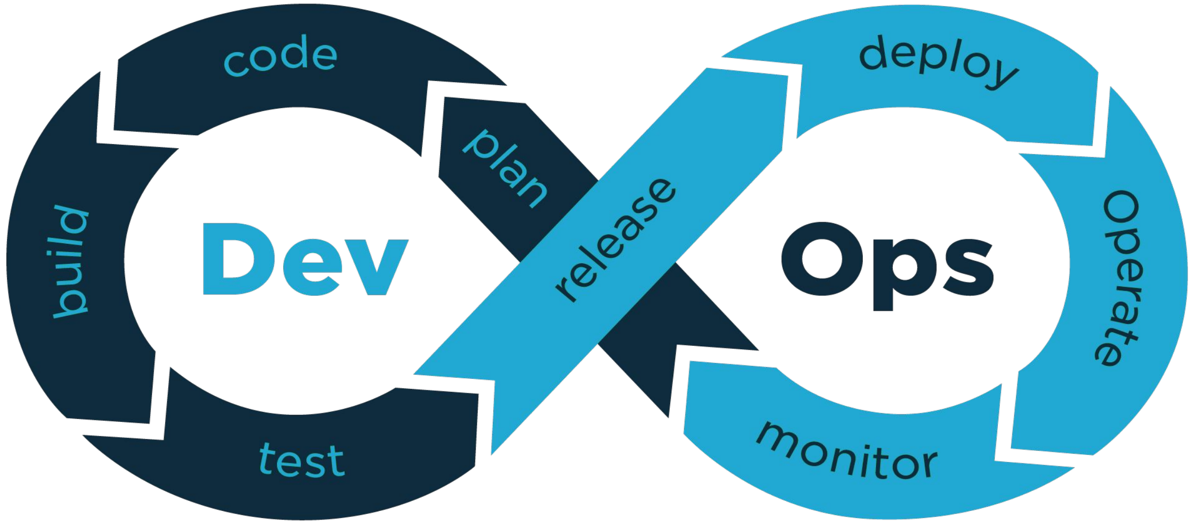


What checks get run `regexes.json`

This file contains all the regexes that will be used to check against the raw content filed returned for a search. Feel free to add/modify and include any specific ones that match the credential you are trying to find. This was graciously borrowed from truffleHog

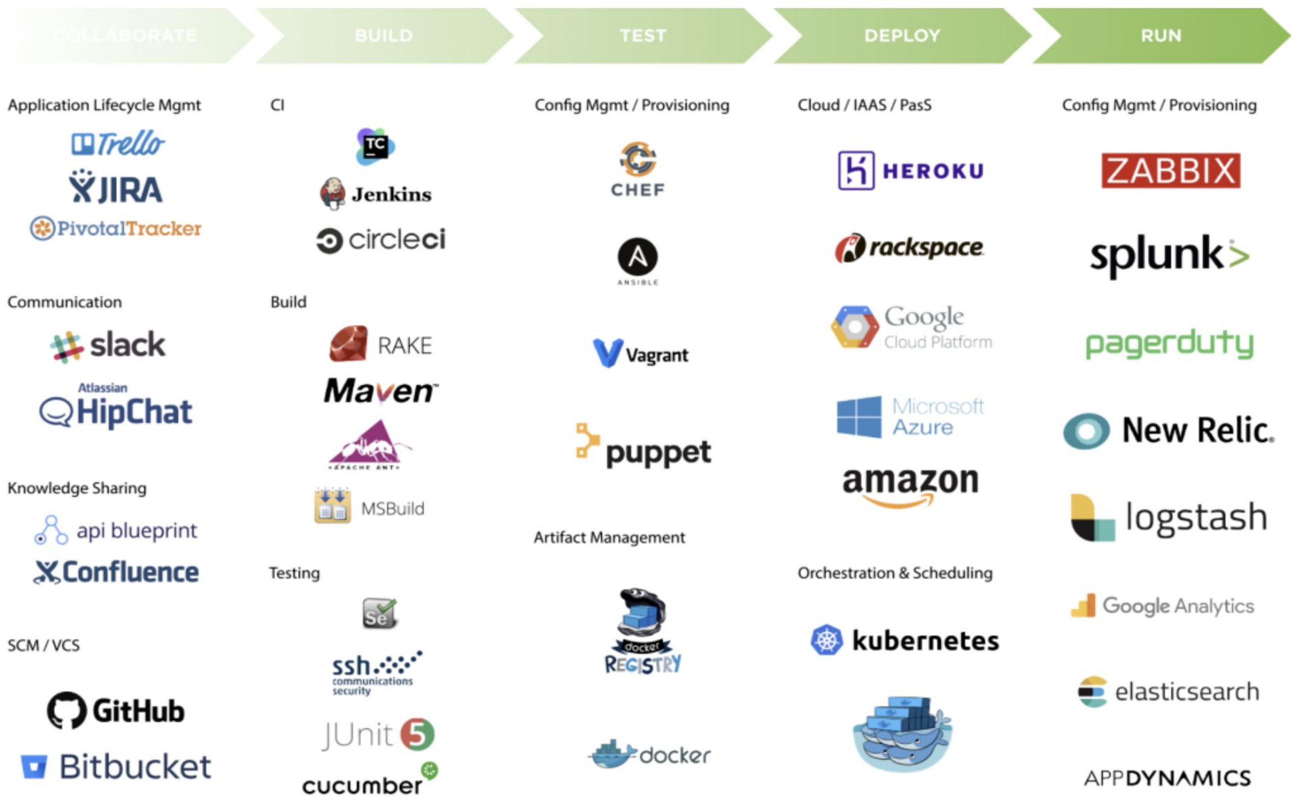
Target Rich Environments

DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with Agile software development; several DevOps aspects came from Agile methodology. Wikipedia



Target Rich Environments

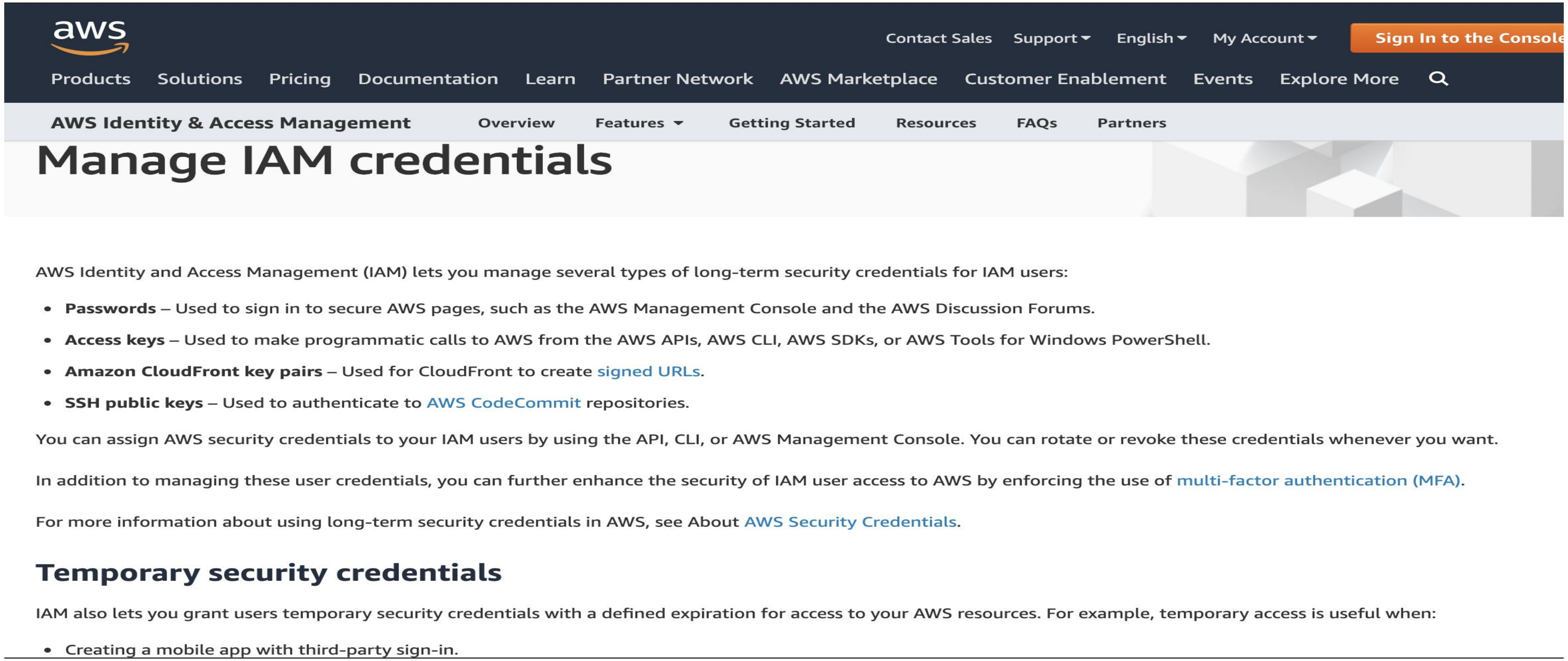
A DevOps toolchain is a set or combination of tools that aid in the delivery, development, and management of software applications throughout the systems development life cycle, as coordinated by an organisation that uses DevOps practices. Wikipedia



What Are We Looking For?

- Developer usually have high privilege credentials
- Ephemeral environments dismissed and poorly monitored
- Disconnection between dev and sec ops
- Widely spread use of open source tools and code at times trusted by default
- Embedded credentials usually end up in public repositories
- Higher risk of rogue insider or abuse of high privilege credentials
- Due to the CI/CD nature link with production environments is immediate
- Cloud environments have made these risks even higher

How Do Cloud Environments Manage Credentials?



Manage IAM credentials

AWS Identity and Access Management (IAM) lets you manage several types of long-term security credentials for IAM users:

- **Passwords** – Used to sign in to secure AWS pages, such as the AWS Management Console and the AWS Discussion Forums.
- **Access keys** – Used to make programmatic calls to AWS from the AWS APIs, AWS CLI, AWS SDKs, or AWS Tools for Windows PowerShell.
- **Amazon CloudFront key pairs** – Used for CloudFront to create signed URLs.
- **SSH public keys** – Used to authenticate to AWS CodeCommit repositories.

You can assign AWS security credentials to your IAM users by using the API, CLI, or AWS Management Console. You can rotate or revoke these credentials whenever you want.

In addition to managing these user credentials, you can further enhance the security of IAM user access to AWS by enforcing the use of multi-factor authentication (MFA).

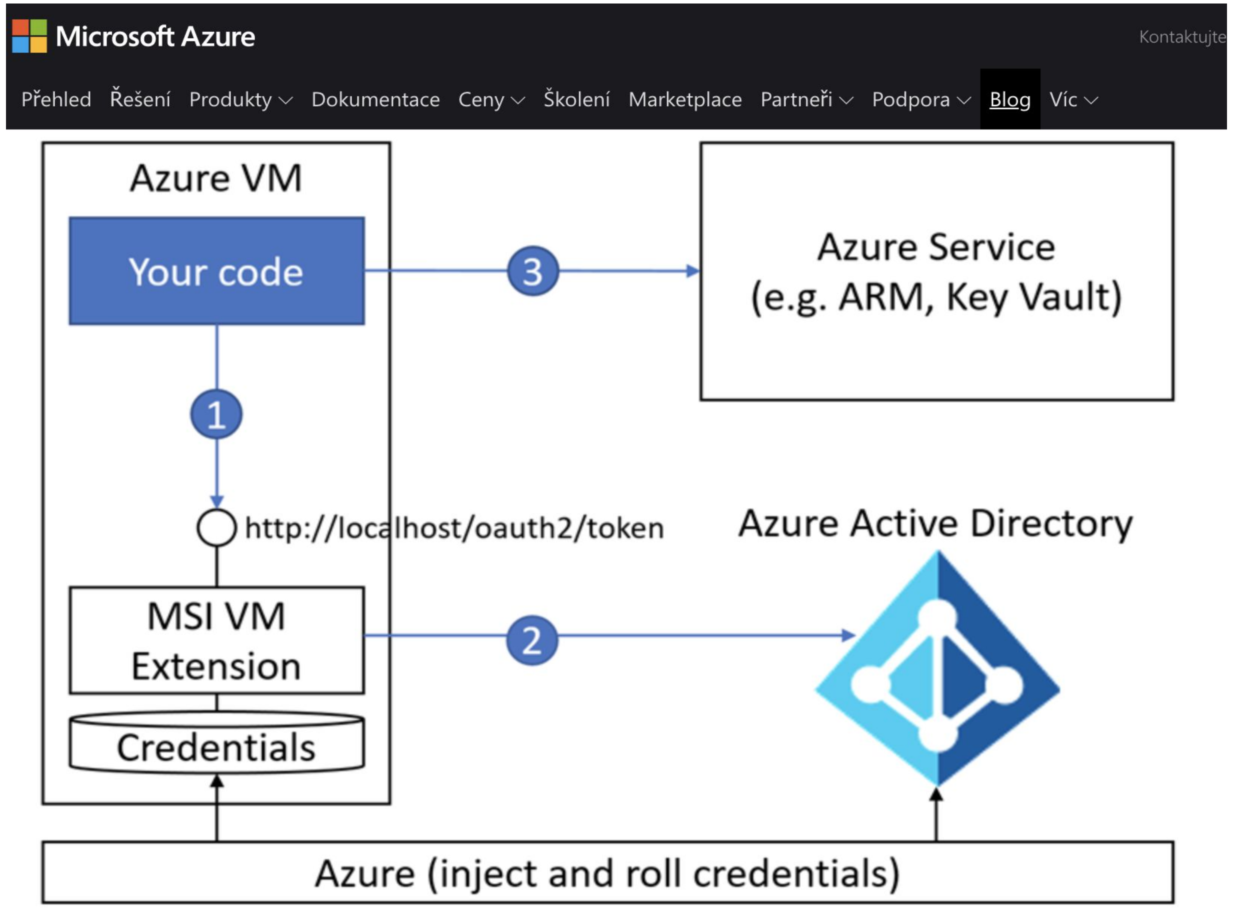
For more information about using long-term security credentials in AWS, see About AWS Security Credentials.

Temporary security credentials

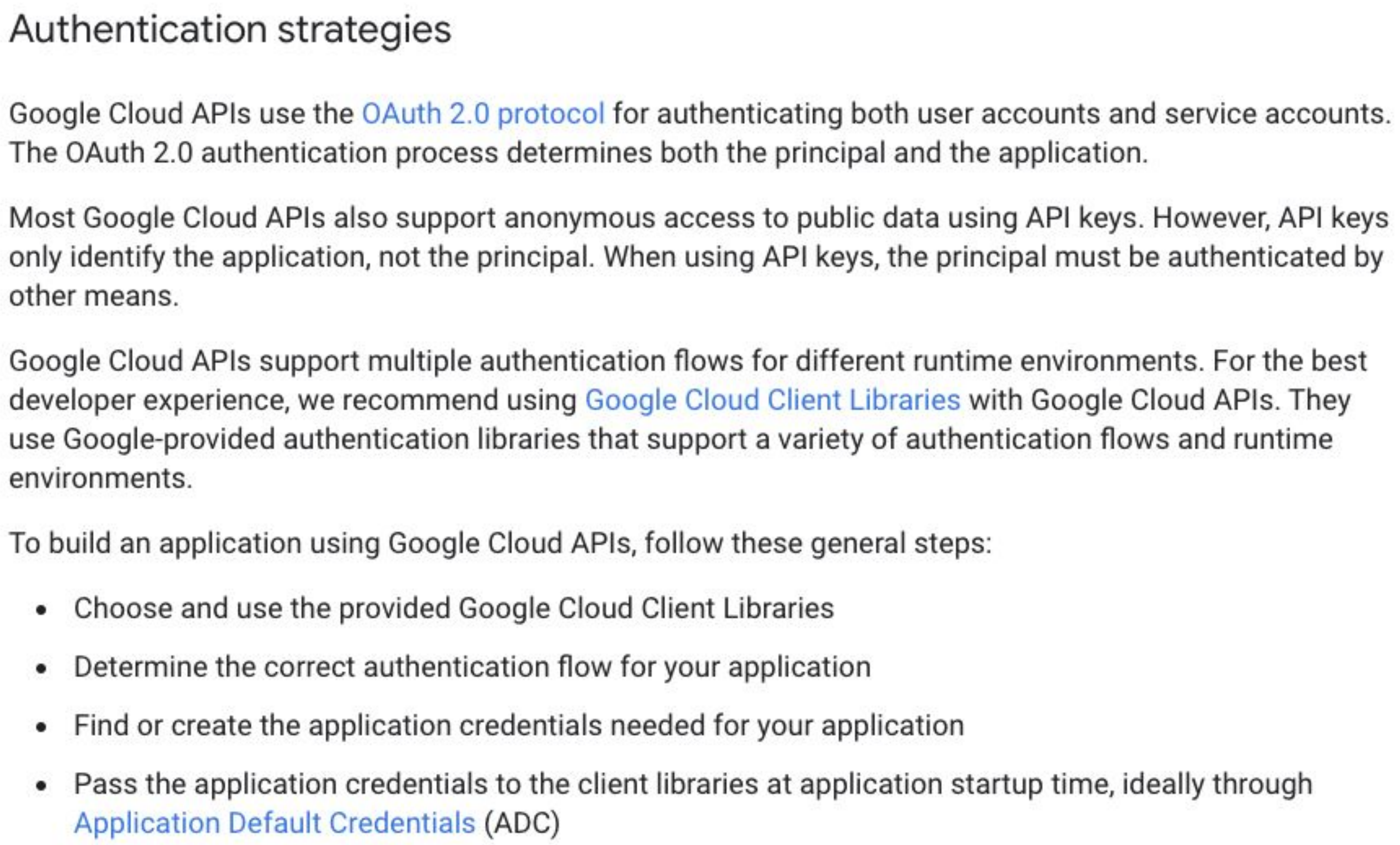
IAM also lets you grant users temporary security credentials with a defined expiration for access to your AWS resources. For example, temporary access is useful when:

- Creating a mobile app with third-party sign-in.

How Do Cloud Environments Manage Credentials?



GCP



Authentication strategies

Google Cloud APIs use the OAuth 2.0 protocol for authenticating both user accounts and service accounts. The OAuth 2.0 authentication process determines both the principal and the application.

Most Google Cloud APIs also support anonymous access to public data using API keys. However, API keys only identify the application, not the principal. When using API keys, the principal must be authenticated by other means.

Google Cloud APIs support multiple authentication flows for different runtime environments. For the best developer experience, we recommend using Google Cloud Client Libraries with Google Cloud APIs. They use Google-provided authentication libraries that support a variety of authentication flows and runtime environments.

To build an application using Google Cloud APIs, follow these general steps:

- Choose and use the provided Google Cloud Client Libraries
- Determine the correct authentication flow for your application
- Find or create the application credentials needed for your application
- Pass the application credentials to the client libraries at application startup time, ideally through Application Default Credentials (ADC)

Other Types Of Credentials Used

- Email & password
- IAM username & Password
- MFA
- Access Keys
- Key pairs
- Account identifiers
- X.509 Certificates

Primary Source Of Leaked Credentials

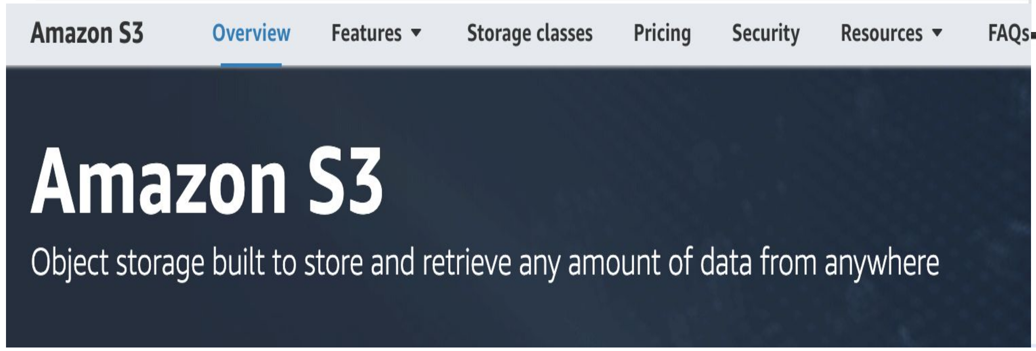
GitHub

IT service management company

GitLab

GitLab

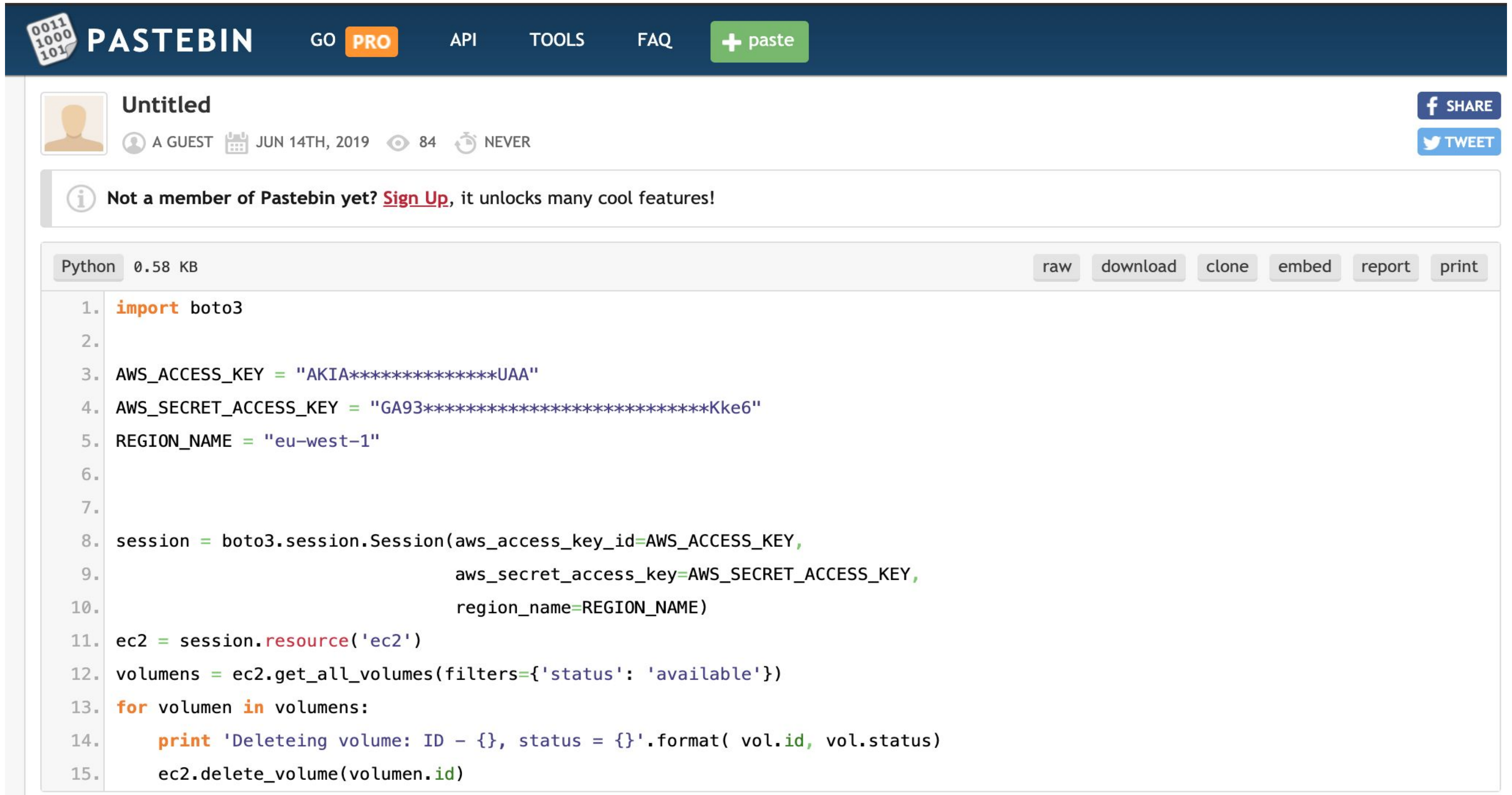
Software



Amazon S3

Object storage built to store and retrieve any amount of data from anywhere

Anywhere Code Is Stored...



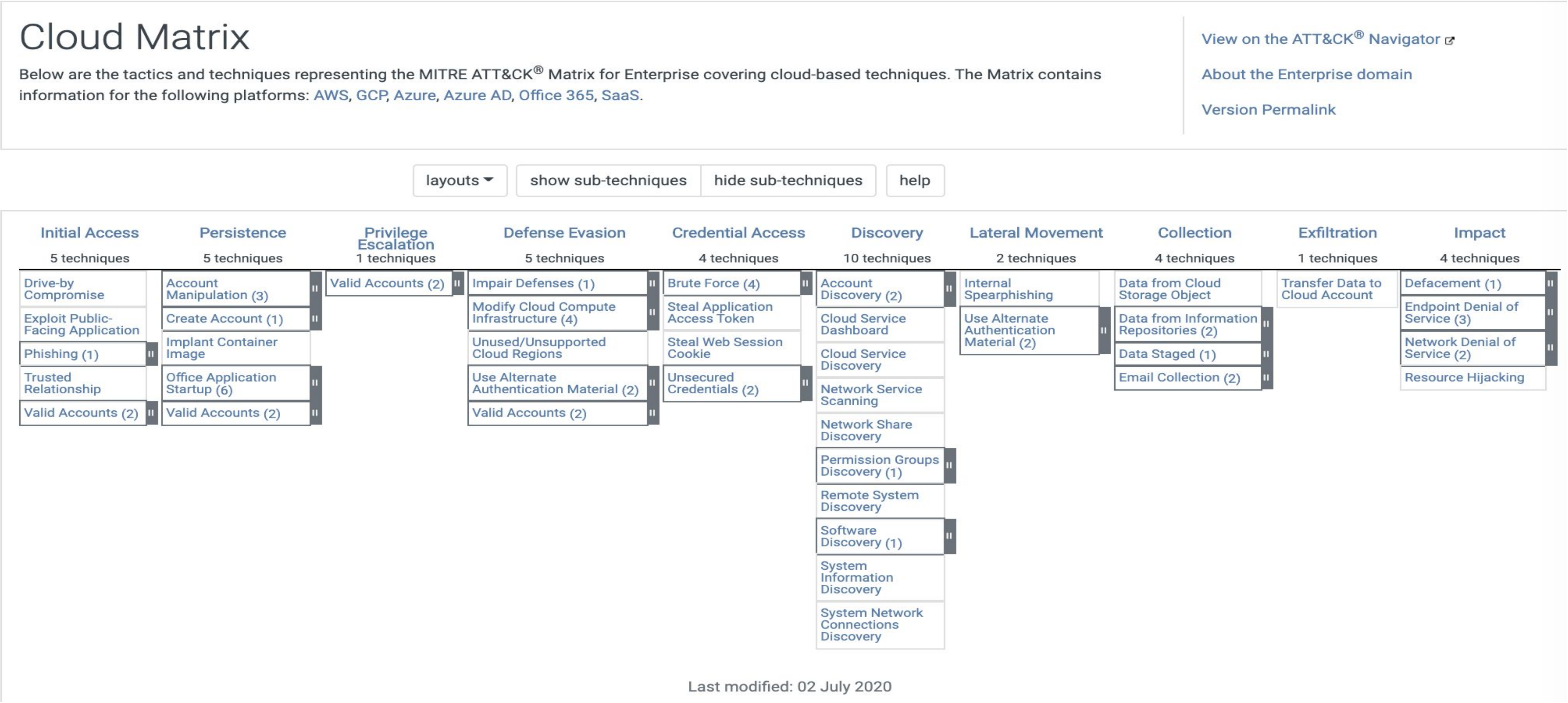
Untitled

A GUEST JUN 14TH, 2019 84 NEVER

Not a member of Pastebin yet? Sign Up, it unlocks many cool features!

```python
import boto3

AWS_ACCESS_KEY = "AKIA**************UAA"
AWS_SECRET_ACCESS_KEY = "GA93***************************Kke6"
REGION_NAME = "eu-west-1"


session = boto3.session.Session(aws_access_key_id=AWS_ACCESS_KEY,
                                aws_secret_access_key=AWS_SECRET_ACCESS_KEY,
                                region_name=REGION_NAME)
ec2 = session.resource('ec2')
volumens = ec2.get_all_volumes(filters={'status': 'available'})
for volumen in volumens:
    print 'Deleteing volume: ID - {}, status = {}'.format( vol.id, vol.status)
    ec2.delete_volume(volumen.id)
```

Unsecured Credentials Mitre Cloud - ATT&CK Matrix



Cloud Matrix

Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for Enterprise covering cloud-based techniques. The Matrix contains information for the following platforms: AWS, GCP, Azure, Azure AD, Office 365, SaaS.

View on the ATT&CK® Navigator

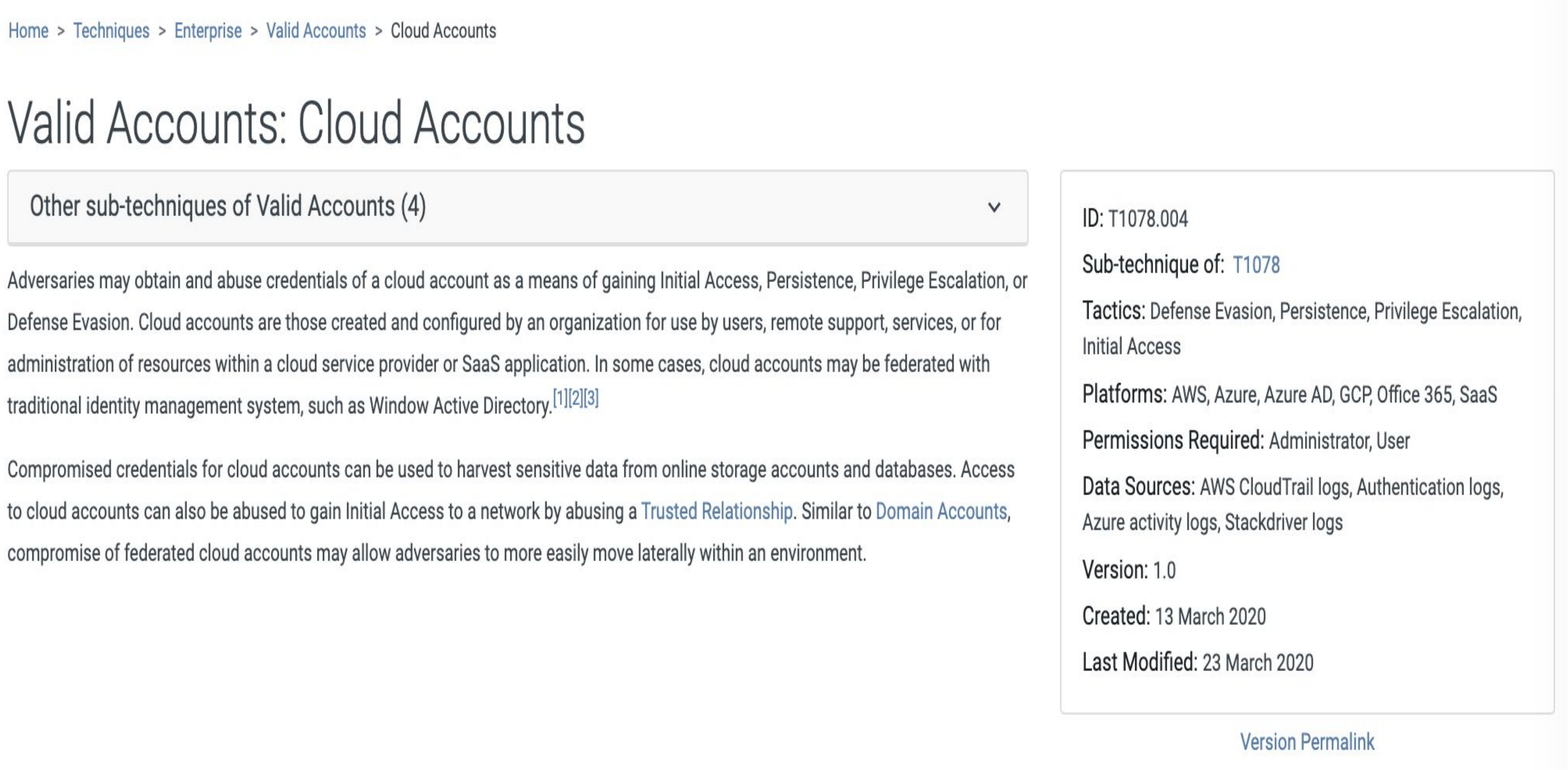
About the Enterprise domain

Version Permalink

Initial Access 5 techniques	Persistence 5 techniques	Privilege Escalation 1 techniques	Defense Evasion 5 techniques	Credential Access 4 techniques	Discovery 10 techniques	Lateral Movement 2 techniques	Collection 4 techniques	Exfiltration 1 techniques	Impact 4 techniques
Drive-by Compromise	Account Manipulation (3)	Valid Accounts (2)	Impair Defenses (1)	Brute Force (4)	Account Discovery (2)	Internal Spearphishing	Data from Cloud Storage Object	Transfer Data to Cloud Account	Defacement (1)
Exploit Public-Facing Application	Create Account (1)		Modify Cloud Compute Infrastructure (4)	Steal Application Access Token	Cloud Service Dashboard	Use Alternate Authentication Material (2)	Data from Information Repositories (2)		Endpoint Denial of Service (3)
Phishing (1)	Implant Container Image		Unused/Unsupported Cloud Regions	Steal Web Session Cookie	Cloud Service Discovery		Data Staged (1)		Network Denial of Service (2)
Trusted Relationship	Office Application Startup (6)		Use Alternate Authentication Material (2)	Unsecured Credentials (2)	Network Service Scanning		Email Collection (2)		Resource Hijacking
Valid Accounts (2)	Valid Accounts (2)		Valid Accounts (2)		Network Share Discovery				
					Permission Groups Discovery (1)				
					Remote System Discovery				
					Software Discovery (1)				
					System Information Discovery				
					System Network Connections Discovery				

Last modified: 02 July 2020

Unsecured Credentials
Mitre Cloud - ATT&CK Matrix



Home > Techniques > Enterprise > Valid Accounts > Cloud Accounts

Valid Accounts: Cloud Accounts

Other sub-techniques of Valid Accounts (4)

Adversaries may obtain and abuse credentials of a cloud account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application. In some cases, cloud accounts may be federated with traditional identity management system, such as Window Active Directory.[1][2][3]

Compromised credentials for cloud accounts can be used to harvest sensitive data from online storage accounts and databases. Access to cloud accounts can also be abused to gain Initial Access to a network by abusing a Trusted Relationship. Similar to Domain Accounts, compromise of federated cloud accounts may allow adversaries to more easily move laterally within an environment.

ID: T1078.004

Sub-technique of: T1078

Tactics: Defense Evasion, Persistence, Privilege Escalation, Initial Access

Platforms: AWS, Azure, Azure AD, GCP, Office 365, SaaS

Permissions Required: Administrator, User

Data Sources: AWS CloudTrail logs, Authentication logs, Azure activity logs, Stackdriver logs

Version: 1.0

Created: 13 March 2020

Last Modified: 23 March 2020

Version Permalink

Lateral Movement - Escalation Of Privileges

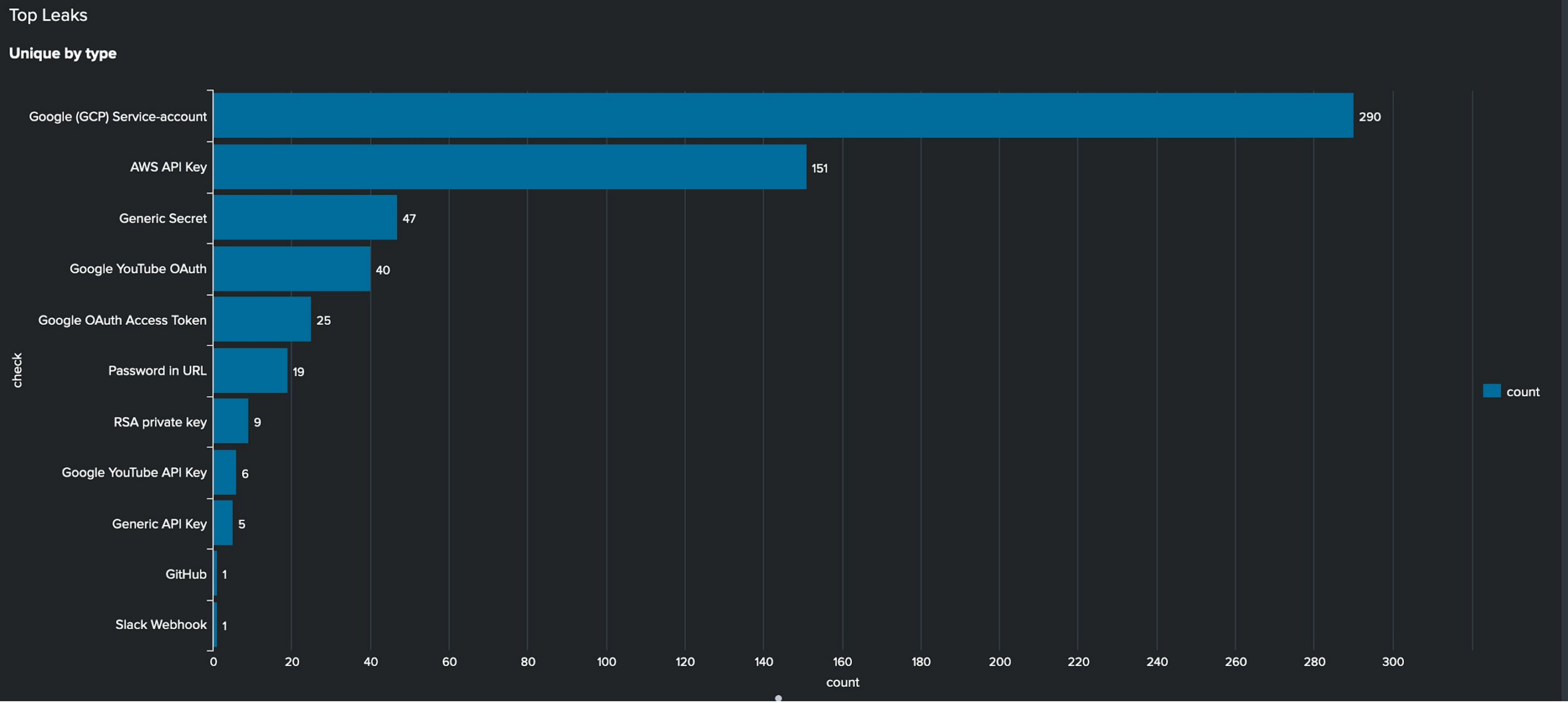
Example:

AWS Security Token Service:

- Create temporary keys = ASIA*
- Create permanent keys = AKIA*
- Create new role trust policies or at yourself to current
- Abuse temporary tokens: sts:AssumeRole / GetSessionToken

Demo 📺

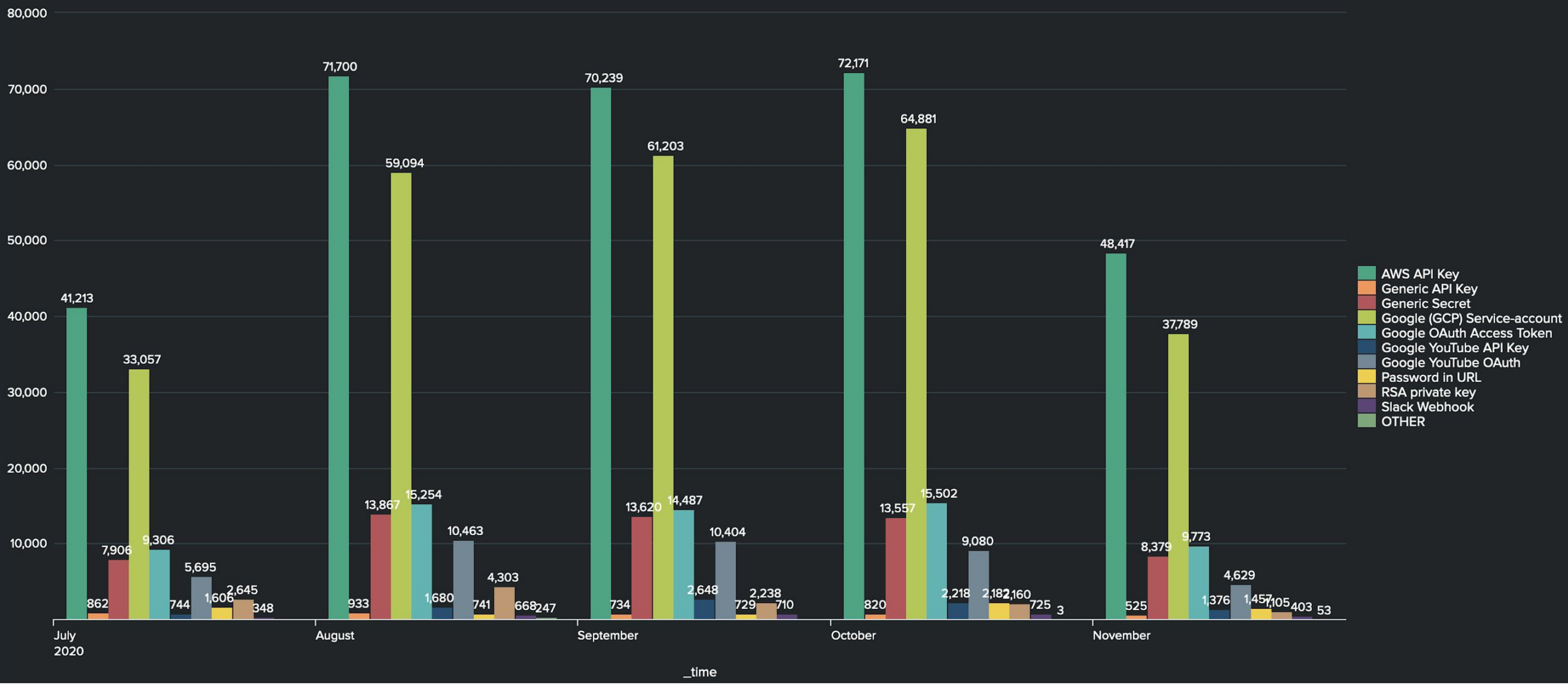
Mining the Data ⛏️👷‍♀️👷‍♂️



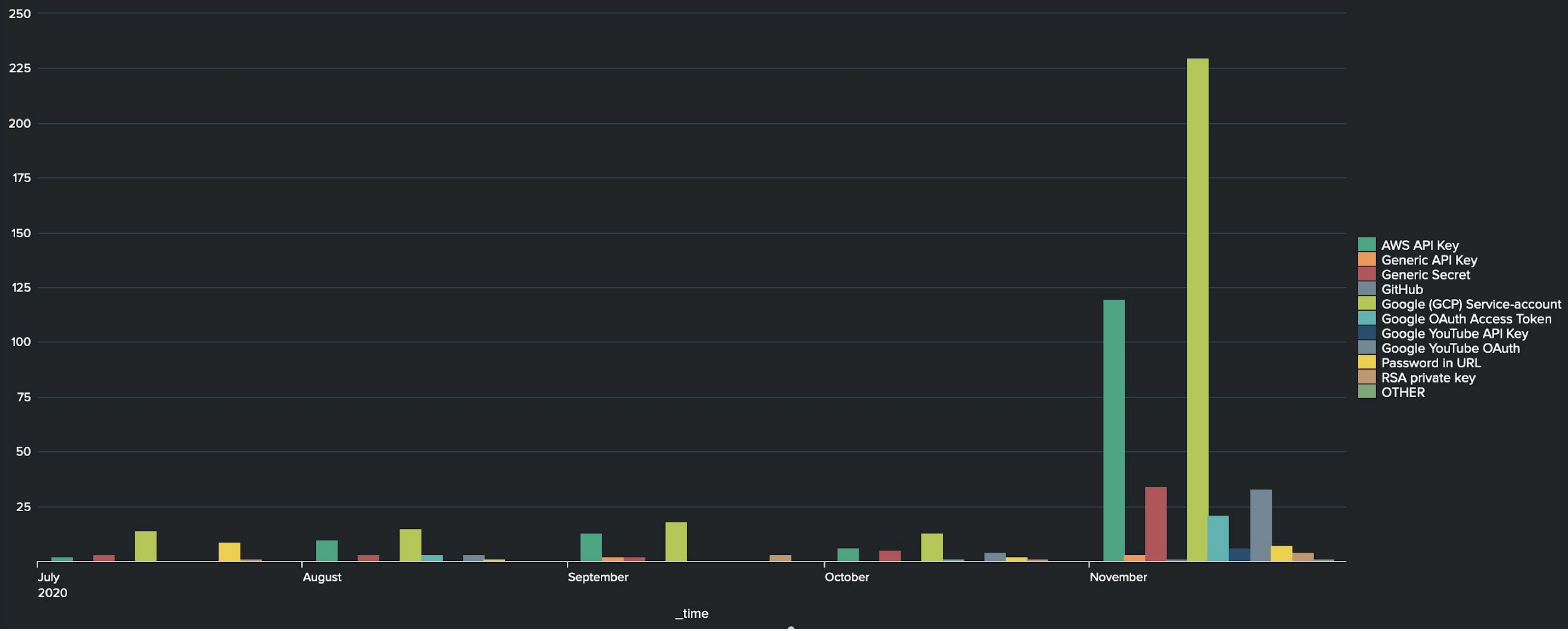
Top Leaks

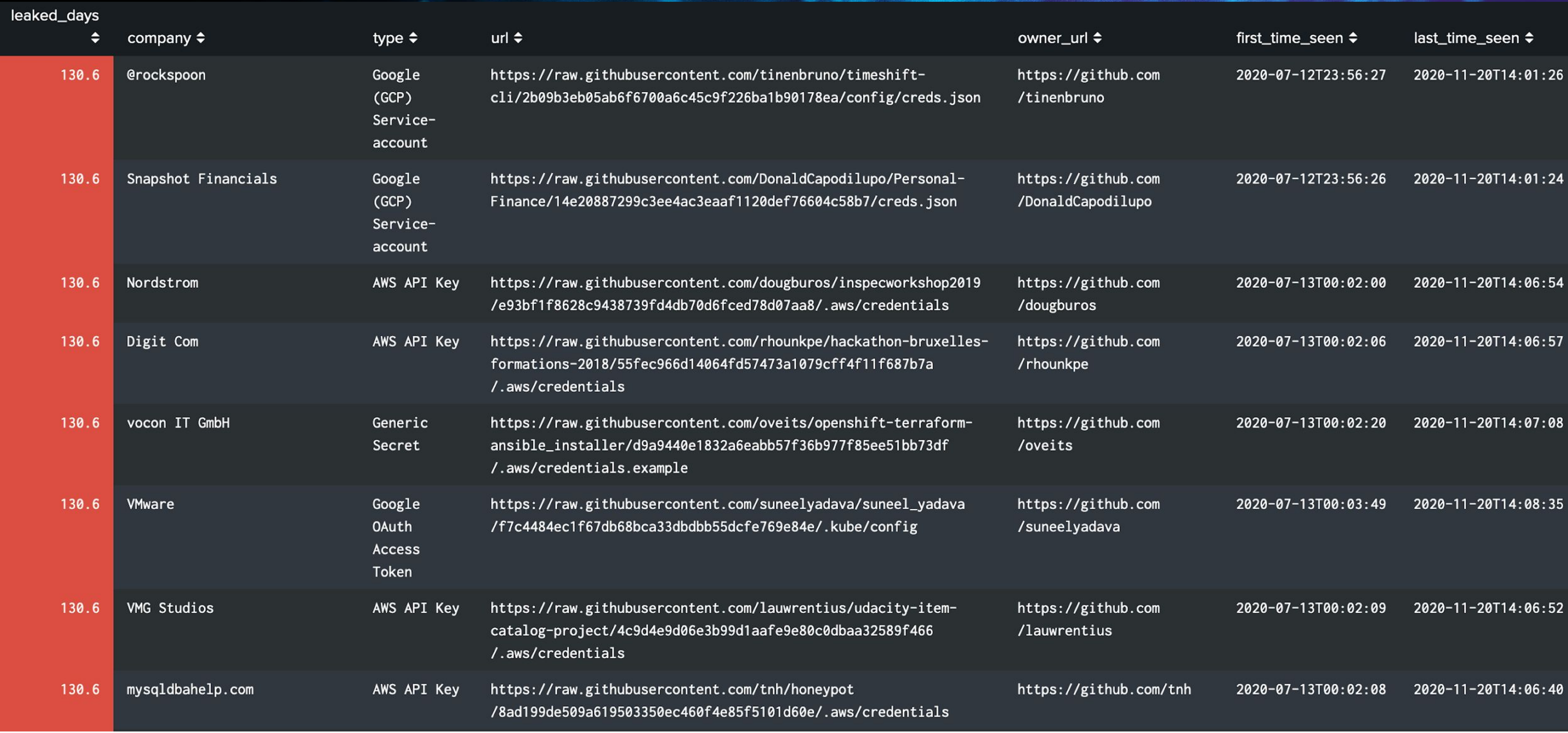
Unique by type

Over time by Type



Over time Unique By Type





leaked_days ⇕	company ⇕	type ⇕	url ⇕	owner_url ⇕	first_time_seen ⇕	last_time_seen ⇕
130.6	@rockspoon	Google (GCP) Service-account	https://raw.githubusercontent.com/tinenbruno/timeshift-cli/2b09b3eb05ab6f6700a6c45c9f226ba1b90178ea/config/creds.json	https://github.com/tinenbruno	2020-07-12T23:56:27	2020-11-20T14:01:26
130.6	Snapshot Financials	Google (GCP) Service-account	https://raw.githubusercontent.com/DonaldCapodilupo/Personal-Finance/14e20887299c3ee4ac3eaaf1120def76604c58b7/creds.json	https://github.com/DonaldCapodilupo	2020-07-12T23:56:26	2020-11-20T14:01:24
130.6	Nordstrom	AWS API Key	https://raw.githubusercontent.com/dougburos/inspecworkshop2019/e93bf1f8628c9438739fd4db70d6fced78d07aa8/.aws/credentials	https://github.com/dougburos	2020-07-13T00:02:00	2020-11-20T14:06:54
130.6	Digit Com	AWS API Key	https://raw.githubusercontent.com/rhounkpe/hackathon-bruxelles-formations-2018/55fec966d14064fd57473a1079cff4f11f687b7a/.aws/credentials	https://github.com/rhounkpe	2020-07-13T00:02:06	2020-11-20T14:06:57
130.6	vocon IT GmbH	Generic Secret	https://raw.githubusercontent.com/oveits/openshift-terraform-ansible_installer/d9a9440e1832a6eabb57f36b977f85ee51bb73df/.aws/credentials.example	https://github.com/oveits	2020-07-13T00:02:20	2020-11-20T14:07:08
130.6	VMware	Google OAuth Access Token	https://raw.githubusercontent.com/suneelyadava/suneel_yadava/f7c4484ec1f67db68bca33dbdbb55dcfe769e84e/.kube/config	https://github.com/suneelyadava	2020-07-13T00:03:49	2020-11-20T14:08:35
130.6	VMG Studios	AWS API Key	https://raw.githubusercontent.com/lauwrentius/udacity-item-catalog-project/4c9d4e9d06e3b99d1aafe9e80c0dbaa32589f466/.aws/credentials	https://github.com/lauwrentius	2020-07-13T00:02:09	2020-11-20T14:06:52
130.6	mysqldbahelp.com	AWS API Key	https://raw.githubusercontent.com/tnh/honeypot/8ad199de509a619503350ec460f4e85f5101d60e/.aws/credentials	https://github.com/tnh	2020-07-13T00:02:08	2020-11-20T14:06:40

Total Companies 97

Daily Leaks across all Companies **7990 per day**

Total unique GCP credentials **290 secrets**

Average time a leak was exposed **44 days**

Takeaways 🥡

- Implementing git-secrets to catch leaks before they are committed is ideal
- Always monitor for leaks using git-wild-hunt or similar tools
- If a secret does get leak clean up the commit history with bfg

Resources 📚

- dataset of all collected leaks https://bit.ly/2IXiamD
- git-wild-hunt https://bit.ly/3kNT0Up
- git-secrets https://bit.ly/36RGm1H
- bfg https://bit.ly/334bud5

Thank you for watching

Feel free to contact us for questions via twitter 🐦 @rodsoto and @d1vious