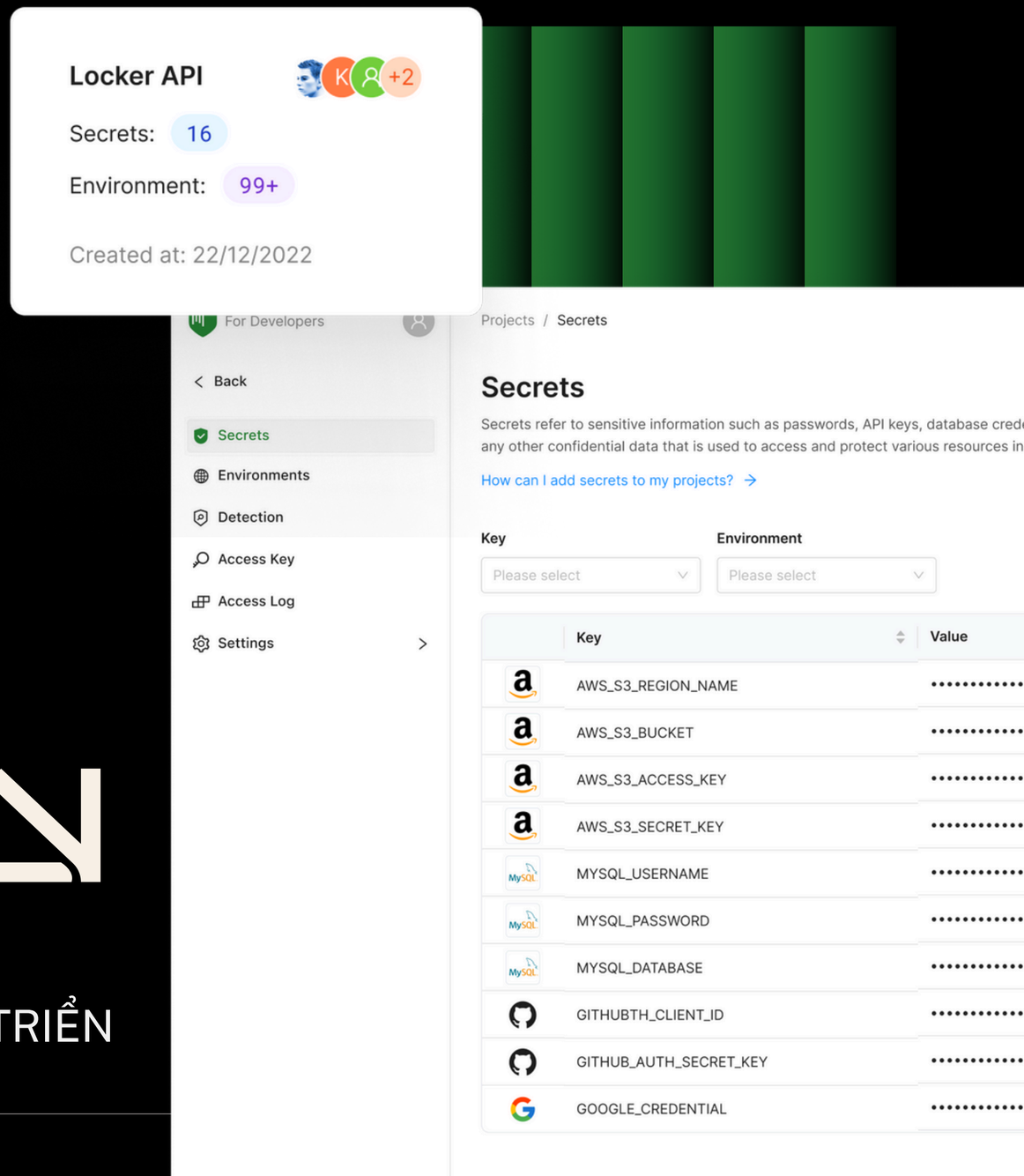
Locker

Một sản phẩm của CyStack

LOCKER SECRETS MANAGER

TRÌNH QUẢN LÝ DỮ LIỆU BÍ MẬT LOCKER
THIẾT KẾ DÀNH CHO NHU CẦU CỦA CÁC NHÀ PHÁT TRIỂN



Locker

Nội dung chính

LOCKER SECRETS MANAGER

01 Secrets và ba sai lầm thường gặp khi lưu trữ và sử dụng

02 Giới thiệu về trình quản lý dữ liệu bí mật Locker (Locker SM)

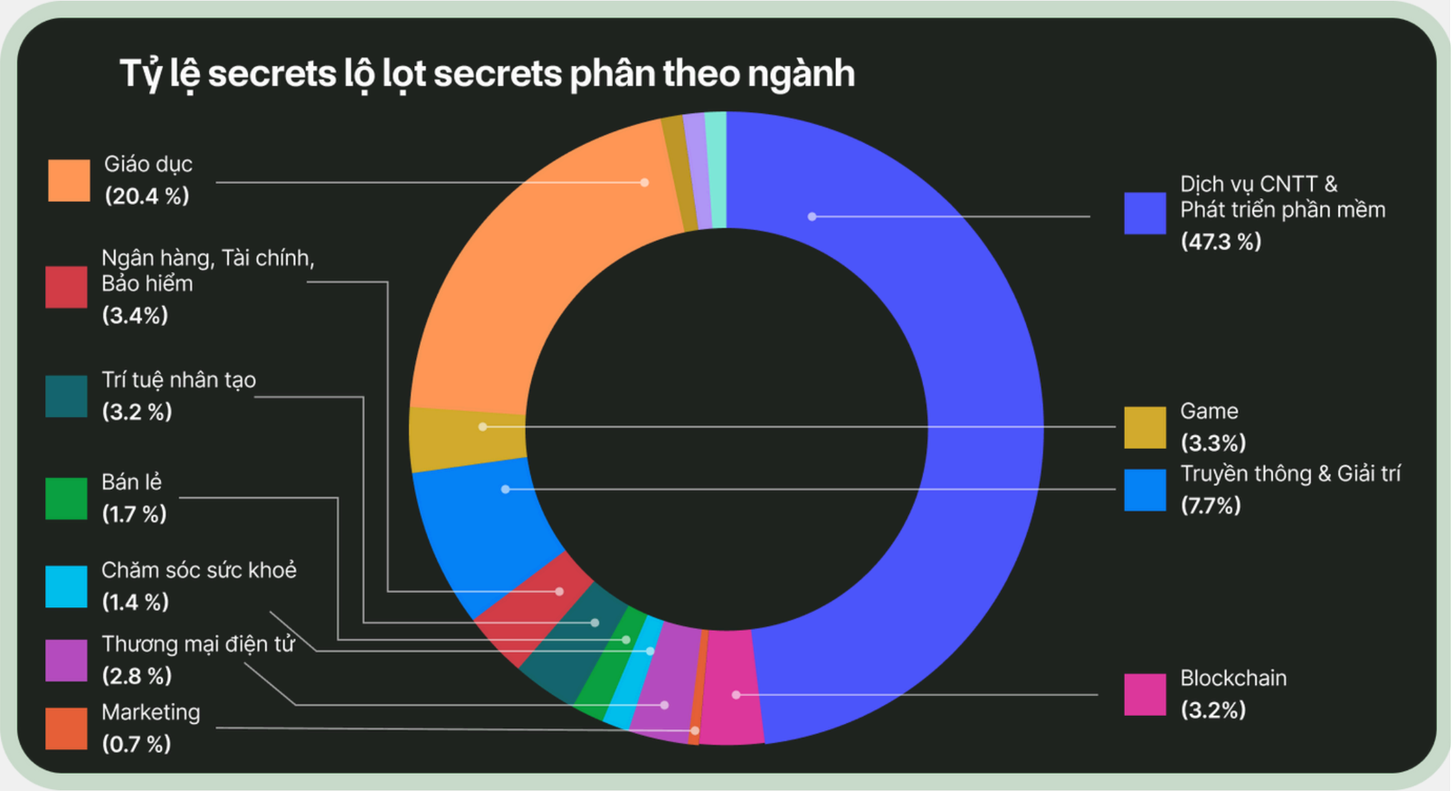
03 Ngữ cảnh sử dụng secrets an toàn với Locker

- Truy xuất secrets từ Locker
- Truy xuất secrets theo môi trường sử dụng
- Lưu secrets trong quá trình chạy ứng dụng
- Cập nhật secrets mới nhất
- Tích hợp vào quá trình CI/CD
- Kiểm soát truy cập secrets

04 Giới thiệu về CyStack

Secrets là gì?

Trong lập trình, "secrets" là các thông tin nhạy cảm như mật khẩu, khóa API, token truy cập, hoặc bất kỳ dữ liệu nào mà ứng dụng hoặc hệ thống cần để truy cập vào các dịch vụ bên ngoài (authorization) hoặc để xác thực người dùng (authentication). Việc bảo vệ các secrets rất quan trọng để tránh truy cập trái phép hoặc lộ thông tin nhạy cảm.



(*) theo Báo cáo Tình trạng lộ lọt secrets trên GitHub Việt Nam 2024, trên tổng 2.37 triệu secrets lộ lọt.

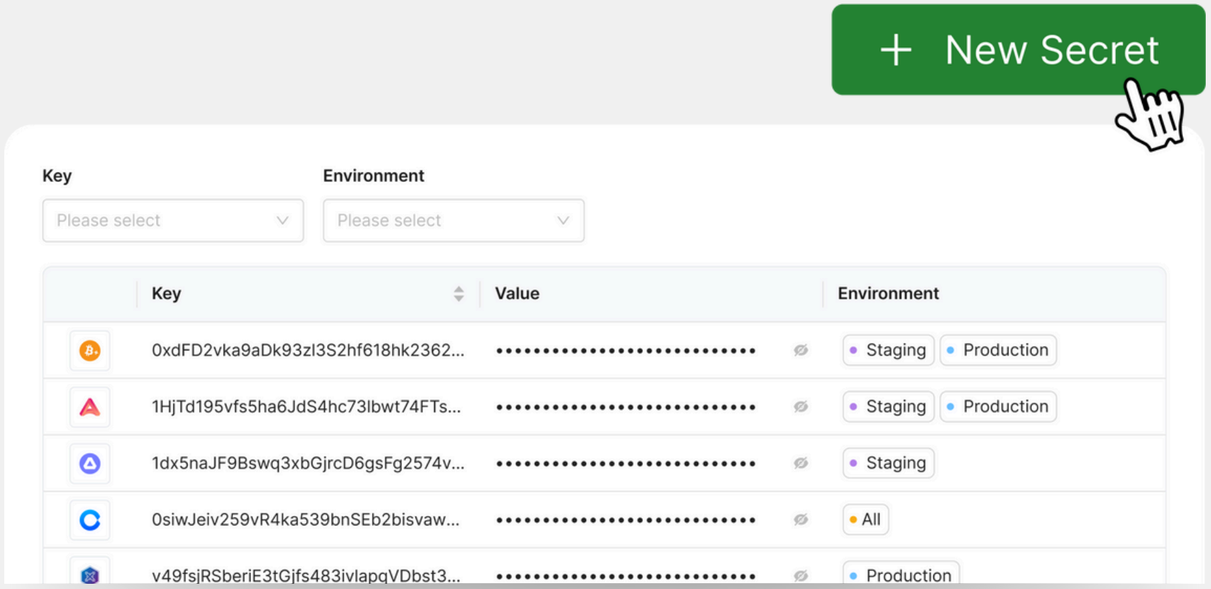
Các sai lầm nghiêm trọng thường gặp của secrets

Nơi lưu trữ	Độ phổ biến (*)	Độ nghiêm trọng	Khuyến nghị
Files lập trình	5.56%	Lộ secrets trong file lập tình có thể dẫn đến xâm nhập hệ thống, mất dữ liệu, hoặc thiệt hại tài chính (như lạm dụng API).	• Phòng ngừa & xử lý: sử dụng công cụ rà soát mã nguồn, thu hồi & thay thế secrets bị lộ, xóa lịch sử Git khi cần. • Giải pháp quản lý: Sử dụng công cụ Secrets Manager.
Files config	93.72%	• Rủi ro bảo mật cao: File config thường được chia sẻ giữa các môi trường hoặc đội nhóm, dễ dẫn đến rò rỉ nếu không được bảo vệ kỹ (ví dụ: để file config trong repo Git). • Tăng nguy cơ tấn công: Secrets trong file config bị lộ có thể bị khai thác để truy cập trái phép vào hệ thống, gây mất dữ liệu hoặc chiếm dụng tài nguyên. • Khó khăn trong kiểm soát: Khi file config bị sao chép hoặc chia sẻ, rất khó đảm bảo rằng secrets không bị lộ ra bên ngoài.	• Xoá file. • Giải pháp quản lý: Dùng công cụ quản lý secrets chuyên dụng.
File biến môi trường	0.62%	File .env dễ bị lộ nếu không bảo vệ kỹ, đặc biệt khi vô tình commit vào repo hoặc máy chủ bị xâm nhập.	• Xoá file. • Giải pháp quản lý: Dùng công cụ quản lý secrets chuyên dụng.

(*) theo Báo cáo Tình trạng lộ lọt secrets trên GitHub Việt Nam 2024, trên tổng 2.37 triệu secrets lộ lọt.

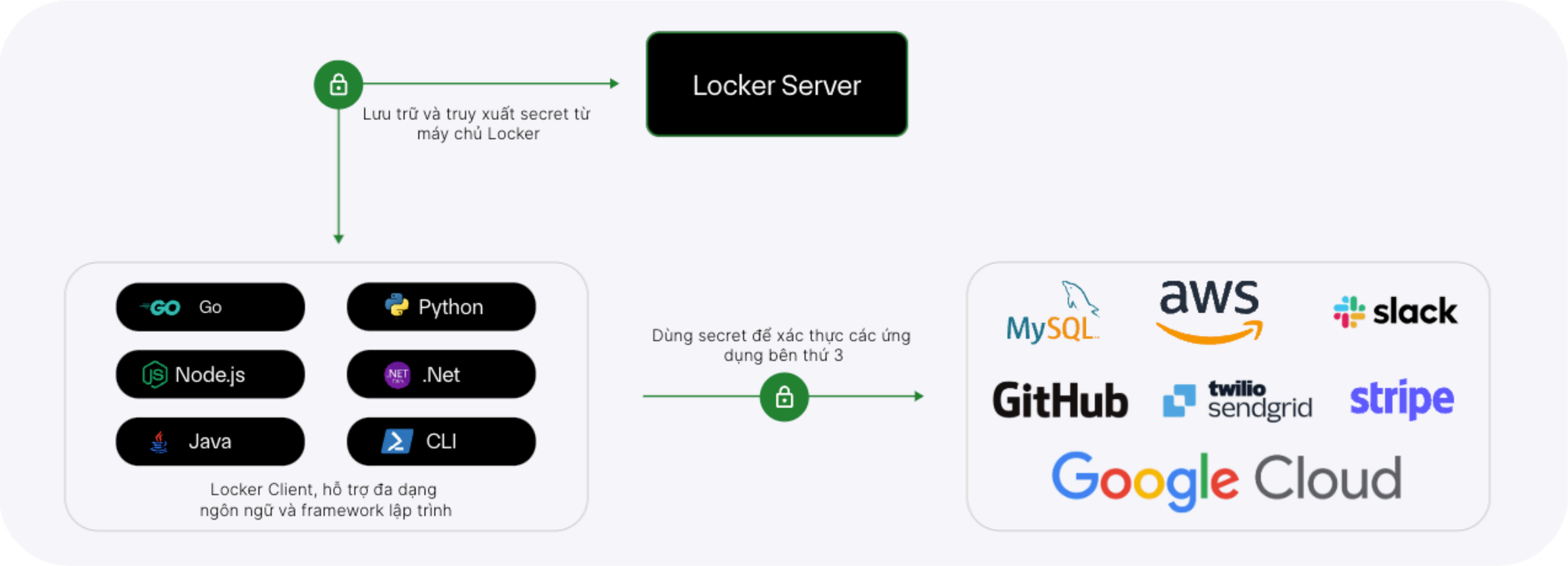
Giới thiệu về trình quản lý dữ liệu bí mật Locker

Trình quản lý dữ liệu bí mật Locker đơn giản hoá việc lưu trữ và quản lý các dữ liệu phục vụ vòng đời phát triển phần mềm (gọi là secrets) như SSH key, API key, mật khẩu cơ sở dữ liệu... Nhà phát triển có thể truy cập secrets thông qua CLI và SDK lập trình được tích hợp vào mã nguồn của dự án.



Cách Locker SM hoạt động

Locker mã hóa và lưu trữ secret vào **một kho dữ liệu tập trung**. Các hoạt động lưu trữ, truy xuất secret được thực hiện thông qua CLI và các SDK lập trình, và **được bảo vệ, giám sát nghiêm ngặt** thông qua nhiều cơ chế bảo mật.



Locker sử dụng mã hóa đầu cuối (end-to-end encryption), kiến trúc mã hóa không thông tin (zero-knowledge encryption) và mã hóa đường truyền để đảm bảo dữ liệu ***không thể bị xâm phạm, kể cả bởi nhóm phát triển Locker.***

BẢO MẬT & MINH BẠCH

NGỮ CẢNH SỬ DỤNG SECRETS AN TOÀN VỚI LOCKER

NGỮ CẢNH SỬ DỤNG

Truy xuất secret từ Locker

01 - NGỮ CẢNH

Lập trình viên cần một secret key để xác thực với ứng dụng bên thứ 3 nhưng không muốn lưu giá trị này trong code để tránh gây mất an toàn.

02 - GIẢI PHÁP

- Lưu trữ secret key vào Locker thông qua giao diện web hoặc SDK.
- Truy xuất secret key trong quá trình chạy chương trình thông qua SDK.

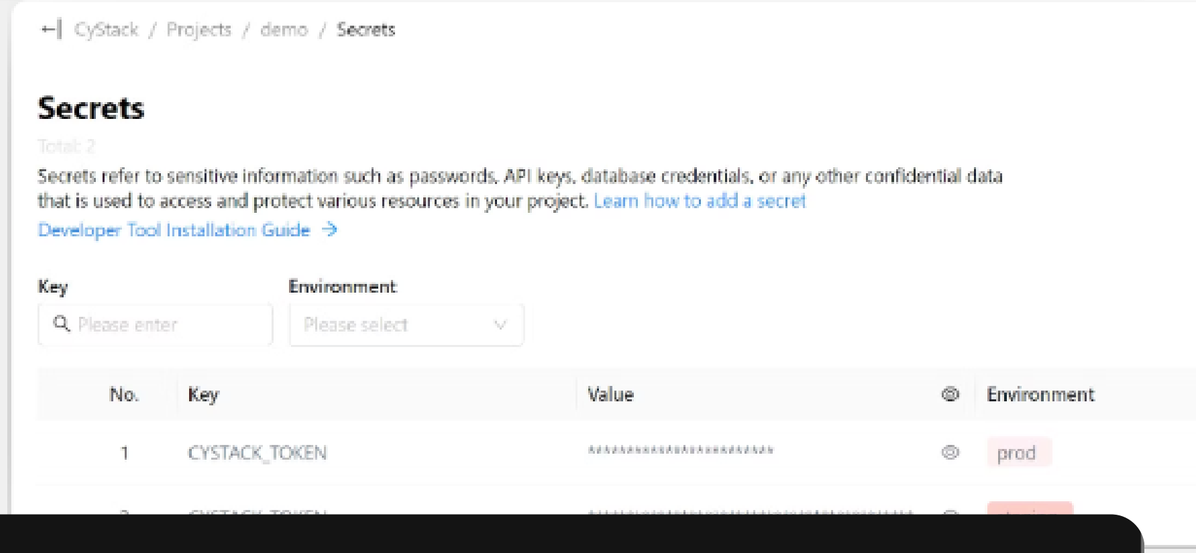
```python
import os
from locker import Locker

locker_inst = Locker(
    access_key_id=os.getenv('LOCKER_ACCESS_KEY_ID'),
    secret_access_key=os.getenv('LOCKER_SECRET_ACCESS_KEY')
)

sentry_dsn = locker_inst.get('SENTRY_DSN')

print(sentry_dsn)

# https://19bbbef9e5a535038.ingest.us.sentry.io/345345190347234 34
```

NGỮ CẢNH SỬ DỤNG

Truy xuất secret theo môi trường sử dụng

01 - NGỮ CẢNH

Trong quá trình phát triển phần mềm, lập trình viên thường thiết lập nhiều môi trường khác nhau (dev/staging/production). Mỗi môi trường, secret cần một giá trị khác nhau cho cùng 1 tên biến.



02 - GIẢI PHÁP

Locker cho phép lưu trữ cùng một secret key với nhiều môi trường khác nhau. Lập trình viên chỉ cần truyền vào đúng tên môi trường mình cần.

```python
import os
from locker import Locker

locker_inst = Locker(
    access_key_id=os.getenv('LOCKER_ACCESS_KEY_ID'),
    secret_access_key=os.getenv('LOCKER_SECRET_ACCESS_KEY')
)

token = locker_inst.get('CYSTACK_TOKEN', environment_name='prod')

print(token)
```

NGỮ CẢNH SỬ DỤNG

Lưu secret trong quá trình chạy ứng dụng (dynamic secret*)

01 - NGỮ CẢNH

Có những secret cần được lưu và sử dụng lại trong quá trình chạy ứng dụng, đây là những secret chưa được khai báo từ trước.

02 - GIẢI PHÁP

Locker cho phép lưu trữ và truy cập các secret bất kỳ trong quá trình hoạt động của ứng dụng.

***Dynamic secrets** là các secret được tạo ra tự động và chỉ khi cần thiết trong quá trình chạy ứng dụng. Chúng không tồn tại trước đó, giúp tăng cường bảo mật bằng cách giảm thiểu rủi ro liên quan đến việc sử dụng các thông tin xác thực cố định.*

NGỮ CẢNH SỬ DỤNG

```python
import os

from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.backends import default_backend
from locker import Locker


def generate_ssh_key_pair(bits=2048, passphrase=None):
    """
    Generate SSH key pair and return them as strings

    Args:
        bits (int): Key length in bits
        passphrase (str): Optional passphrase to encrypt private key

    Returns:
        tuple: (private_key_str, public_key_str)
    """
    # Generate private key
    private_key = rsa.generate_private_key(
        public_exponent=65537,
        key_size=bits,
        backend=default_backend()
    )
```

```python
 # Get public key
    public_key = private_key.public_key()

    # Serialize private key with optional passphrase
    encryption_algorithm = (
        serialization.BestAvailableEncryption(passphrase.encode())
        if passphrase else serialization.NoEncryption()
    )

    # Convert private key to string
    private_key_str = private_key.private_bytes(
        encoding=serialization.Encoding.PEM,
        format=serialization.PrivateFormat.PKCS8,
        encryption_algorithm=encryption_algorithm
    ).decode('utf-8')

    # Convert public key to string
    public_key_str = public_key.public_bytes(
        encoding=serialization.Encoding.PEM,
        format=serialization.PublicFormat.SubjectPublicKeyInfo
    ).decode('utf-8')

    return private_key_str, public_key_str


try:
    locker_inst = Locker(
        access_key_id=os.getenv("LOCKER_ACCESS_KEY_ID"),
        secret_access_key=os.getenv("LOCKER_SECRET_ACCESS_KEY")
    )
    private_key, public_key = generate_ssh_key_pair(bits=2048)
    locker_inst.create(key="SSH_PRIVATE_KEY", value=private_key)
except Exception as e:
    print(f"error: {e}")
```

Lưu secret trong quá trình chạy ứng dụng (dynamic secret*)

NGỮ CẢNH SỬ DỤNG

Cập nhật secret mới nhất

LOCKER SECRETS MANAGER

01 - NGỮ CẢNH

Khi một secret bị thay đổi do hết hạn hoặc bị lộ, lập trình viên cần cập nhật giá trị mới mà không làm gián đoạn các ứng dụng đang sử dụng secret này.

02 - GIẢI PHÁP

- Locker tự động đồng bộ giá trị secret mới đến các ứng dụng hoặc dịch vụ đang sử dụng thông qua SDK.
- Sử dụng tính năng thông báo của Locker để cảnh báo lập trình viên và các bên liên quan về việc secret đã được thay đổi (upcoming).

```python
import os
from locker import Locker

locker_inst = Locker(
  access_key_id=os.getenv("LOCKER_ACCESS_KEY_ID"),
  secret_access_key=os.getenv("LOCKR_SECRET_ACCESS_KEY")
)
locker_inst.resttime = 120
secret = locker_inst.modify(key="YOUR_NEW_SECRET_KEY",
                            value="UPDATED_SECRET_VALUE",
                            environment_name="prod")
print(secret.key, secret.value, secret.environment_name)
```

NGỮ CẢNH SỬ DỤNG

Tích hợp vào quá trình CI/CD

01 - NGỮ CẢNH

Trong các pipeline CI/CD, các secrets cần được sử dụng để triển khai ứng dụng, kết nối với các dịch vụ bên thứ ba, hoặc chạy các bài kiểm tra tự động.

02 - GIẢI PHÁP

Theo dõi lịch sử truy cập và cập nhật secrets dễ dàng mà không làm gián đoạn pipeline.

```
export DB_PASSWORD=$(locker secret get --key DB_PASSWORD --env prod)

export API_KEY=$(locker secret get --key API_KEY --env prod)

echo "Connecting to database with password..."
docker run -e DB_PASSWORD=$DB_PASSWORD my-database-image

echo "Deploying application with API key..."
curl -H "Authorization: Bearer $API_KEY" -X POST https://api.example.com/deploy
```

NGỮ CẢNH SỬ DỤNG

Kiểm soát truy cập secrets

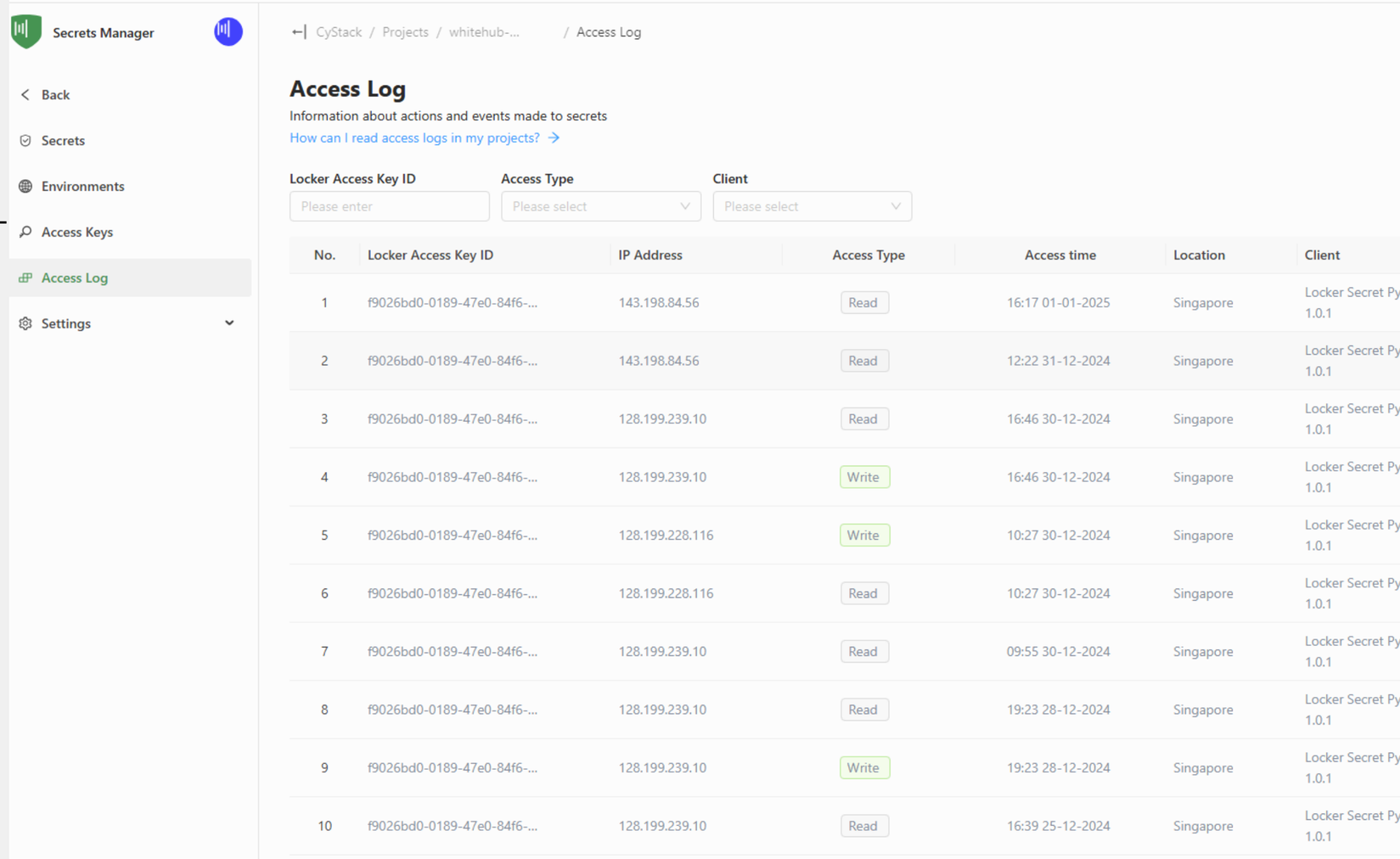
LOCKER SECRETS MANAGER

01 - NGỮ CẢNH

Tổ chức cần đảm bảo rằng chỉ những lập trình viên được uỷ quyền mới có quyền truy cập vào các secrets quan trọng như API keys hoặc thông tin đăng nhập vào database.

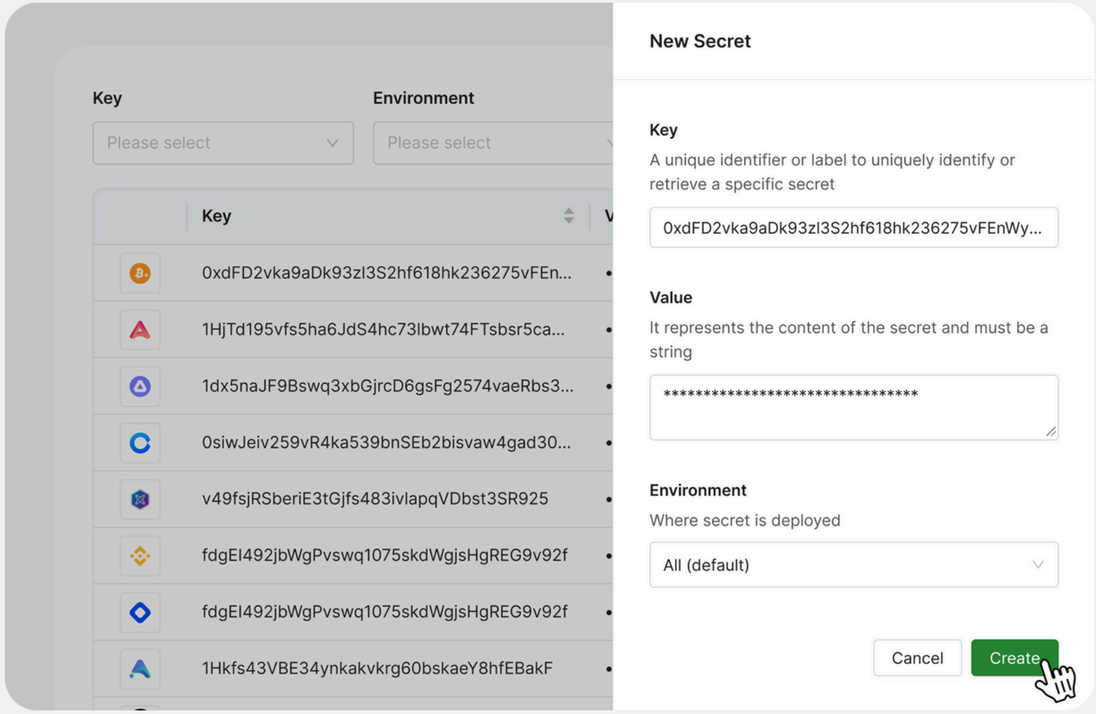
02 - GIẢI PHÁP

- Lưu trữ secret key vào Locker thông qua giao diện web hoặc SDK.
- Sử dụng tính năng phân quyền của Locker để gán quyền truy cập cho từng thành viên hoặc nhóm cụ thể.
- Nhật ký truy cập hiển thị thông tin về các hành động và sự kiện được thực hiện trên Locker CLI và Locker SDK đối với secrets trong một dự án.



No.	Locker Access Key ID	IP Address	Access Type	Access time	Location	Client
1	f9026bd0-0189-47e0-84f6-...	143.198.84.56	Read	16:17 01-01-2025	Singapore	Locker Secret Py 1.0.1
2	f9026bd0-0189-47e0-84f6-...	143.198.84.56	Read	12:22 31-12-2024	Singapore	Locker Secret Py 1.0.1
3	f9026bd0-0189-47e0-84f6-...	128.199.239.10	Read	16:46 30-12-2024	Singapore	Locker Secret Py 1.0.1
4	f9026bd0-0189-47e0-84f6-...	128.199.239.10	Write	16:46 30-12-2024	Singapore	Locker Secret Py 1.0.1
5	f9026bd0-0189-47e0-84f6-...	128.199.228.116	Write	10:27 30-12-2024	Singapore	Locker Secret Py 1.0.1
6	f9026bd0-0189-47e0-84f6-...	128.199.228.116	Read	10:27 30-12-2024	Singapore	Locker Secret Py 1.0.1
7	f9026bd0-0189-47e0-84f6-...	128.199.239.10	Read	09:55 30-12-2024	Singapore	Locker Secret Py 1.0.1
8	f9026bd0-0189-47e0-84f6-...	128.199.239.10	Read	19:23 28-12-2024	Singapore	Locker Secret Py 1.0.1
9	f9026bd0-0189-47e0-84f6-...	128.199.239.10	Write	19:23 28-12-2024	Singapore	Locker Secret Py 1.0.1
10	f9026bd0-0189-47e0-84f6-...	128.199.239.10	Read	16:39 25-12-2024	Singapore	Locker Secret Py 1.0.1

Tính năng của Locker SM



01

Lưu trữ và quản lý secrets được mã hoá.

02

Phân loại secrets theo môi trường phát triển: dev, staging, production...

03

Hỗ trợ SDK lập trình: với các ngôn ngữ lập trình phổ biến như Python, Go, JS, C#

04

Hỗ trợ CLI: thuận tiện cho quá trình CI/CD và DevSecOps.

05

Kiểm soát chia sẻ, truy cập secrets cho thành viên theo từng cấp độ.

06

Kiểm tra nhật ký truy cập qua thời gian, vị trí, địa chỉ IP của người sử dụng

07

Tự động phát hiện secrets được sử dụng không an toàn trong mã nguồn, tự động thay thế bằng Locker SDK, và xoá lịch sử Git.

08

Tự động thay đổi khoá bí mật định kỳ.

Tính khả dụng của Locker SM

Linh hoạt và thích ứng thông qua hỗ trợ các ngôn ngữ lập trình phổ biến

BẢO MẬT VÀ MINH BẠCH

BẢO MẬT VÀ MINH BẠCH

BẢO MẬT VÀ MINH BẠCH

BẢO MẬT VÀ MINH BẠCH

BẢO MẬT VÀ MINH BẠCH

https://locker.io/

Giới thiệu về CyStack

CyStack

CyStack là công ty an ninh mạng hàng đầu tại Việt Nam, được biết đến với khả năng nghiên cứu chuyên sâu và xây dựng các giải pháp tiên phong về bảo mật.

Sản phẩm và dịch vụ của chúng tôi tập trung vào quản lý lỗ hổng, phát hiện mối đe doạ, bảo mật cộng đồng, và bảo mật dữ liệu. Locker là một sản phẩm được phát triển bởi CyStack.

ĐƯỢC TIN TƯỞNG BỞI NHIỀU TỔ CHỨC LỚN

CAKE by VPBank · Sendo · ACB · momo · mitsubishi.com

vntrip.vn · AGRIBANK · OpenCommerce · One Mount

Liên hệ dùng thử

LOCKER
SECRETS
MANAGER

Locker Secrets Manager

Bản cloud

Locker Secrets Manager

Bản self-hosted tự triển khai và quản lý

ĐIỆN THOẠI

02471099656

ĐỊA CHỈ

Toà nhà Tân Hồng Hà Complex,
317 Trường Chinh, Ngã Tư Sở,
Thanh Xuân, Hà Nội

WEBSITE

https://locker.io/

EMAIL

contact@locker.io

Thank you

LOCKER
SECRETS
MANAGER

2025

LOCKER.IO

WHERE SECRETS
STAY SAFE!