BLACKSWAN CYBERSECURITY

THREAT ADVISORY
Stripe Merchant API Keys Exposed

August 19, 2026

Executive Summary

Multiple security researchers and news outlets have reported the public release of a dataset containing allegedly live Stripe API credentials associated with hundreds of merchant accounts. Available reporting indicates the dataset contains API keys from approximately **659 merchant accounts**, associated with roughly **35 GB of customer and payment-related data impacting approximately 688,000 customer records across 42 countries**. The evidence currently suggests this is **not a compromise of Stripe infrastructure itself**, but rather the compromise and collection of merchant-owned API credentials.

Researchers report that approximately **650 exposed keys were live secret keys (sk_live)**, with additional restricted keys included in the dataset. Exposed credentials allegedly enabled access to Stripe API resources including customer records, charges, invoices, subscriptions, payment intents, refunds, disputes, payouts, and balance transactions.

Separate reporting also identified more than **50,000 Stripe API keys exposed in public repositories, GitHub Actions logs, and misconfigured web servers**, suggesting the incident may represent a broader ecosystem-wide secret management problem rather than an isolated breach.

What Happened?

According to reporting from RansomNews, Cyber Security News, GBHackers, CyberPress, Security Affairs, and others:

- A threat actor operating under the alias **"Satanic"** published a freely downloadable archive on a cybercrime forum on **August 18, 2026**.
- The archive reportedly contains:
  - Live Stripe merchant API keys
  - Customer records
  - Payment-related records
  - Charges
  - Checkout sessions
  - Invoices
  - Subscriptions
  - Refunds
  - Disputes
  - Payout information
- Researchers indicate the data appears consistent with legitimate Stripe API output.
- Current evidence indicates attackers likely obtained merchant credentials through:

- Public code repositories
- GitHub Actions logs
- Exposed .env files
- Misconfigured web servers
- Backups
- Infostealer malware logs

Indicators of Compromise (IOCs)

Credential Indicators

Stripe secret keys:

1 sk_live_

Stripe restricted keys:

1 rk_live_

Examples of Stripe object identifiers observed in reporting:

1 acct_
2 cus_
3 ch_
4 pi_
5 cs_live_
6 dp_
7 req_

Researchers reported the leaked dataset contained authentic-looking Stripe object formats using these prefixes.

MITRE ATT&CK Mapping

ATT&CK Technique	Description
T1552.001	Credentials in Files (.env files, source code)
T1552.004	Private Keys
T1589	Gather Victim Identity Information

THREAT ADVISORY
Stripe Merchant API Keys Exposed
August 19, 2026

ATT&CK Technique	Description
T1078	Valid Accounts
T1555	Credentials from Password Stores
T1213	Data from Information Repositories
T1530	Data from Cloud Storage
T1656	Financial Theft / Fraud-related abuse

ATT&CK mapping is analytical and based on reported activity patterns, not directly attributed by Stripe or the reporting organizations.

Threat Actor Tactics, Techniques, and Procedures (TTPs)

Initial Access

Source Code Exposure

Researchers identified Stripe keys embedded in:

- Public GitHub repositories
- Accidentally-public repositories
- Configuration files
- Source code comments
- Environment files (.env)

CI/CD Pipeline Leakage

Credentials exposed through:

- GitHub Actions logs
- Debug output
- Build artifacts
- Improper secret masking

Misconfigured Infrastructure

Researchers reported discovering thousands of servers exposing Stripe-related strings, some of which contained functioning API credentials.

THREAT ADVISORY
Stripe Merchant API Keys Exposed
August 19, 2026

Infostealer Collection

Several reports indicate the exposed merchant credentials may also have originated from infostealer logs harvested from compromised developer workstations.

Actions on Objectives

Once a valid sk_live key is obtained, reporting indicates attackers may be able to:

- Enumerate customer data
- Retrieve transaction histories
- Create charges
- Generate payment links
- Issue refunds
- Access invoices
- Access payouts
- Modify payment workflows
- Alter webhook configurations
- Potentially access connected accounts in Stripe Connect deployments

Researchers reported successfully creating a fraudulent payment link and conducting a test transaction using an exposed key during their validation activities.

Detection Opportunities

Search Source Repositories

Scan (shell):

```
1 sk_live_
2 rk_live_
```

Across:

- GitHub
- GitLab
- Bitbucket
- Azure DevOps
- Internal repositories

Search Environment Files

Look for:

```
1 STRIPE_SECRET_KEY=
2 STRIPE_API_KEY=
3 STRIPE_LIVE_KEY=
```

Locations:

```
1 .env
2 config.json
3 application.properties
4 settings.py
5 web.config
```

Monitor Stripe Activity

Investigate for:

Unexpected API Operations

- New payment intents
- Unexpected refunds
- Unusual charge activity
- New products created
- New webhook endpoints

Payout Changes

Investigate:

- Bank account modifications
- Payout destination changes
- Increased payout volume
- New connected accounts

SIEM Detection Recommendations

Alert on:

- Stripe API activity from unusual geographies
- API keys used from new IP addresses
- Sudden increases in API calls
- Abnormal refund activity
- Large exports of customer data
- Creation of previously unseen webhooks

Remediation Recommendations

Immediate (0-24 Hours)

1. Rotate All Stripe Secret Keys

Immediately rotate:

- sk_live_*
- rk_live_*

Do not simply disable compromised systems without rotating credentials.

2. Audit Webhook Configurations

Review:

- Recently added webhooks
- Modified webhook URLs
- Unknown webhook destinations

3. Review Transaction History

Look for:

- Unauthorized charges
- Refund abuse
- Suspicious payment links
- Unrecognized subscriptions

4. Review Payout Destinations

Validate:

- Bank account details
- Connected accounts
- Recent payout routing changes

Because reporting indicates hundreds of affected accounts retained payout permissions.

Short-Term (1-7 Days)

Secret Discovery

Run organization-wide scans for:

1 sk_live_
2 rk_live_

Across:

- Source control
- CI/CD logs
- Cloud storage
- Backups
- Documentation systems
- Ticketing systems

Implement Least Privilege

Where possible:

- Replace full-access keys with restricted keys
- Limit permissions to required operations
- Separate production and development credentials

Endpoint Investigation

Review developer systems for evidence of:

- Lumma
- RedLine
- Vidar
- Raccoon

- Other credential-stealing malware

Since infostealer activity is one of the suspected collection vectors.

Long-Term Controls

Secrets Management

Implement:

- Azure Key Vault
- AWS Secrets Manager
- HashiCorp Vault
- GCP Secret Manager

Avoid hardcoded secrets.

Developer Security Controls

Implement:

- Pre-commit secret scanning
- Repository secret scanning
- CI/CD secret scanning
- DLP controls for credentials

Continuous Monitoring

Deploy:

- Credential exposure monitoring
- GitHub secret scanning
- Dark web monitoring
- SaaS configuration monitoring

Executive Assessment

This incident highlights a growing trend where attackers bypass hardened cloud providers and instead target **customer-managed credentials**. Available evidence indicates the threat stems primarily from exposed merchant API secrets rather than a compromise of Stripe itself.

THREAT ADVISORY
Stripe Merchant API Keys Exposed
August 19, 2026

Organizations using Stripe should assume that any secret key exposed in source code, CI/CD logs, backups, or infected developer endpoints can be weaponized rapidly for financial fraud, customer data access, webhook manipulation, and payout abuse. Immediate credential rotation, transaction review, secret-scanning, and webhook validation are the highest-priority response actions.

Primary Sources: Cyber Security News, RansomNews, Security Affairs, GBHackers, CyberPress, and CyberNews.