ICT DIVISION
FUTURE IS HERE

BANGLADESH COMPUTER COUNCIL
COMPUTER FOR EVERYTHING

BGD e-GOV CIRT

TLP:CLEAR


# Python NodeStealer Evolves into Full-Featured Spyware: Browser, Credential, Clipboard, Keylogging and Facebook Account Intelligence Theft

**TLP:** CLEAR
**Distribution:** Public
**Advisory on:** Python NodeStealer Evolves into Full-Featured Spyware: Browser, Credential, Clipboard, Keylogging and Facebook Account Intelligence Theft
**Severity:** High
**Threat Type:** Information Stealer / Spyware / Credential Theft / Account Takeover
**Potentially Affected Sectors:** Government, Banking & Finance, MFS, E-commerce, Telecommunications, Software/IT, Media, Education, Healthcare and SMEs
**Primary Platform:** Windows
**Date:** 06 September 2026

## Executive Summary

The Cyber Threat Intelligence (CTI) Unit of BGD e-GOV CIRT is issuing this advisory regarding a significantly upgraded variant of the **Python-based NodeStealer** malware. The latest observed version represents a substantial evolution from a conventional browser credential stealer into a **full-featured spyware platform** capable of persistent surveillance, credential collection, browser-data theft, clipboard monitoring, keystroke logging, screenshot capture and extensive Facebook account intelligence gathering.

NodeStealer has been tracked since 2023 as an information-stealing malware family targeting browser information and Facebook accounts. Previous versions primarily focused on credentials and sensitive browser data. The newly observed variant expands this functionality considerably and introduces capabilities that allow an operator to monitor a victim's activity continuously.

The latest variant uses the Python ecosystem and incorporates libraries and functionality for:

- Keyboard monitoring
- Clipboard collection
- Screenshot capture
- Browser credential theft
- Browser session/cookie theft
- Wi-Fi password collection
- File collection from the victim's Pictures directory
- Facebook account intelligence collection
- Facebook Ads Manager information theft
- Telegram-based command and control
- Automated data exfiltration

Netskope researchers assessed that portions of the newly added code exhibit characteristics consistent with **AI-assisted development**, including systematic use of decorative emojis in program output/logging that were not present in earlier NodeStealer variants. This assessment should be understood as an observation regarding coding characteristics rather than definitive proof of AI authorship.

The most important development is the transition from **credential theft to surveillance and intelligence collection**. The malware now queries more than **20 Facebook Graph API endpoints**, compared with only two endpoints in earlier versions. This allows attackers to build a significantly broader profile of the individual operating a Facebook account, including identity, social connections, security information and commerce-related data.

The campaign identified by the research primarily affected victims in **Asia and North America**, across multiple sectors, with financial services representing the leading segment. Although the source does not establish specific infections in Bangladesh, the observed targeting geography, functionality and sectoral exposure make the malware relevant to organizations and users in Bangladesh.

For Bangladesh, the principal concern is the potential compromise of corporate browsers, social-media business accounts, advertising accounts, financial-service credentials, email sessions, saved passwords and employee workstations. A single infected endpoint may provide attackers with both credentials and sufficient contextual information to conduct subsequent fraud, impersonation, account takeover or targeted social engineering.

## Threat Overview

The CTI Unit assesses the threat as HIGH because NodeStealer's expanded functionality changes the potential impact from a relatively narrow credential-stealing incident to a broader **endpoint surveillance and identity-compromise scenario**.

The potential attack progression is:

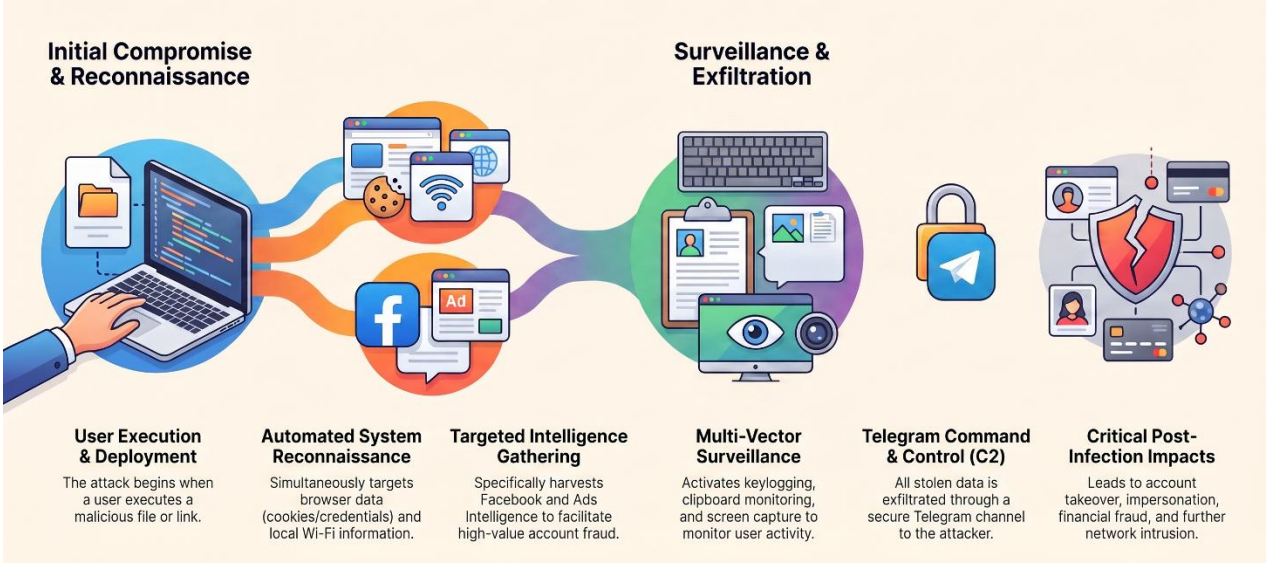



***Figure: Python NodeStealer: Anatomy of a Cyber Attack.***

This architecture means that an infected employee workstation should not be considered only a **malware infection**. It should potentially be treated as an **identity and credential compromise event**.

## Malware Evolution

The evolution of NodeStealer can be summarized as follows.

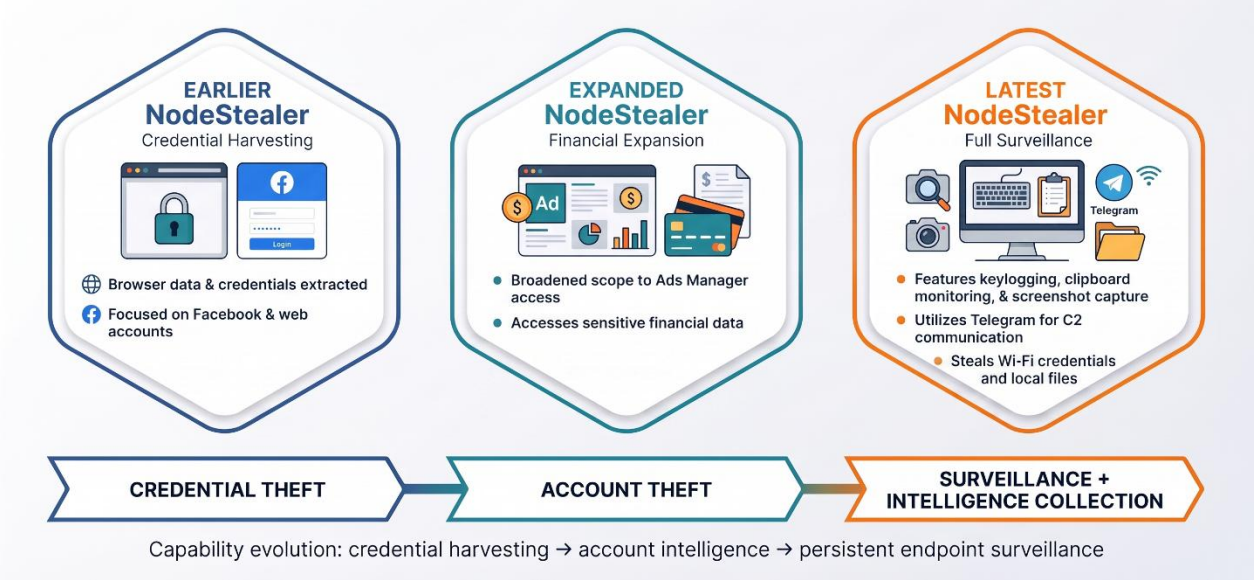



***Figure: Evolution of NodeStealer - From Infostealer to Full Spyware***

## Initial Access and Execution

The referenced research focuses primarily on the malware's capabilities and does not establish a single universal delivery mechanism for every NodeStealer sample.

Organizations should therefore monitor common malware-delivery paths, including:

- Phishing attachments
- Malicious download links
- Fake software installers
- Cracked/pirated applications
- Fake browser updates
- Malicious advertisements
- Social-media-delivered files
- Fake business documents
- Compromised websites

The principal security issue is user execution of a malicious Python-based payload.

An illustrative execution chain is:

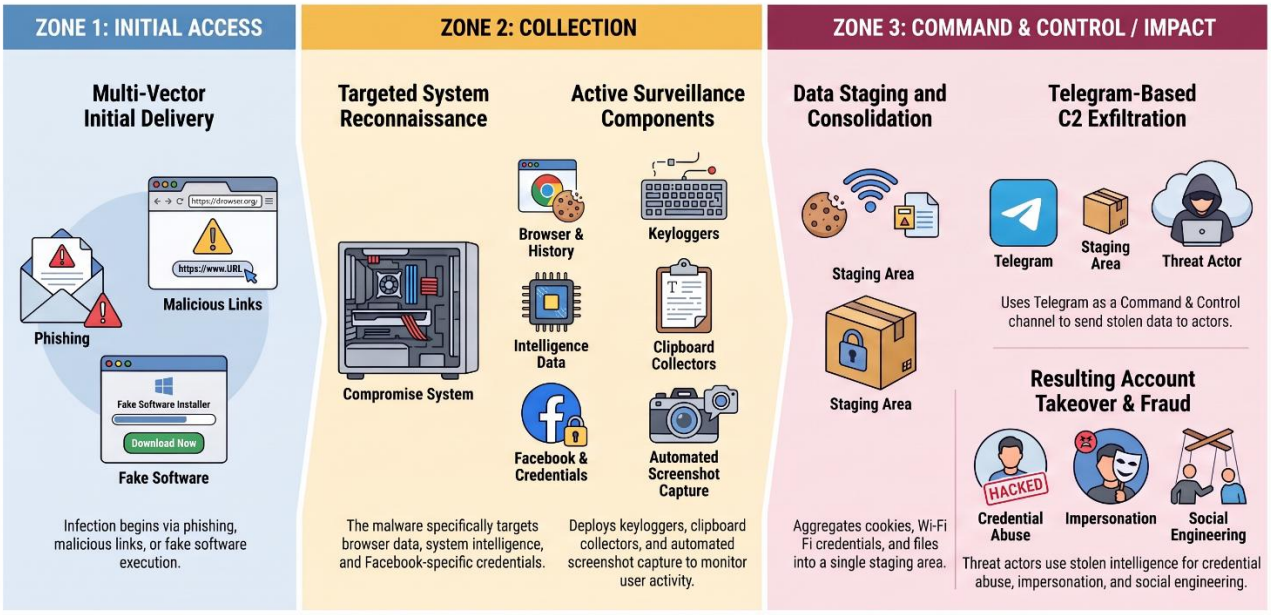



***Figure: Complete NodeStealer Infection and Data Theft Chain***

## Python-Based Malware Architecture

The use of Python provides attackers with access to a large ecosystem of libraries for:

- Keyboard interaction
- Browser/file processing
- HTTP communication
- Telegram communication
- Screenshot capture
- System information collection
- Encryption/encoding
- File compression

This modularity allows capabilities to be added without redesigning the complete malware.

The malware's architecture can therefore be represented as:

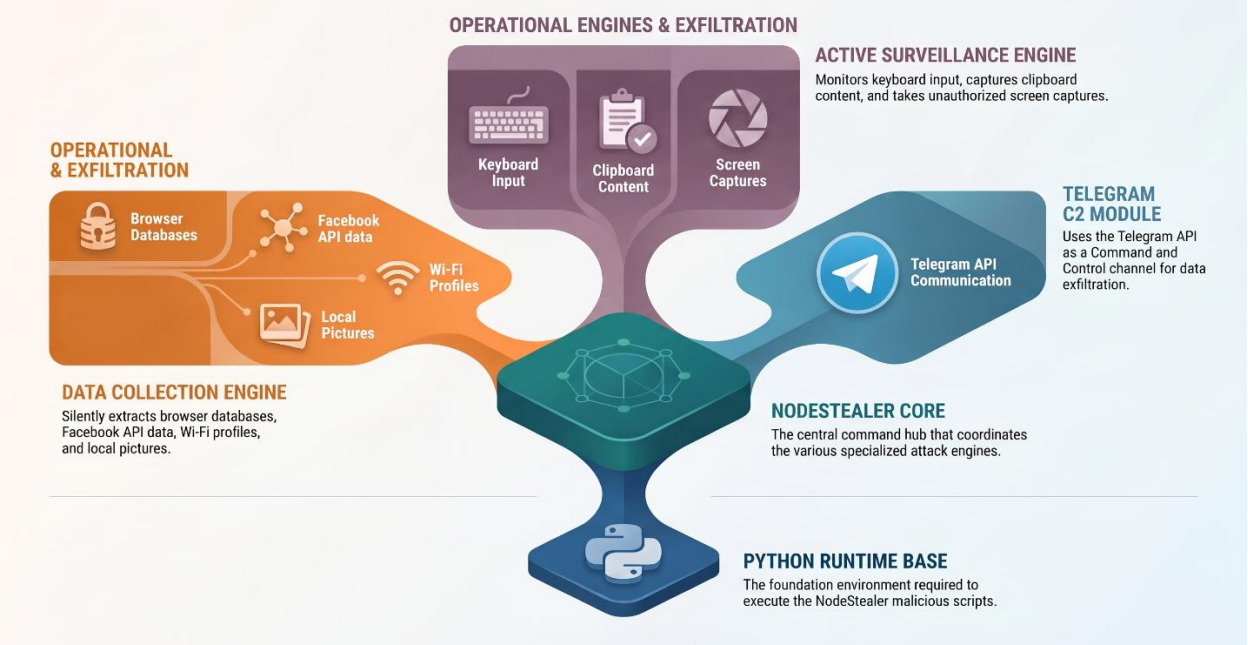



***Figure: Python NodeStealer Internal Architecture***

## Keylogging

One of the most important additions is a persistent keylogging capability. The latest NodeStealer uses the Python pynput library to monitor keyboard input. Captured keystrokes are written to a temporary file using a naming structure reported as: `keylog({ip}).txt`

A separate Python thread periodically sends the collected keylogging data to the Telegram-based C2 infrastructure. The reported interval is 120 seconds, after which the local contents are deleted. The operational flow is:

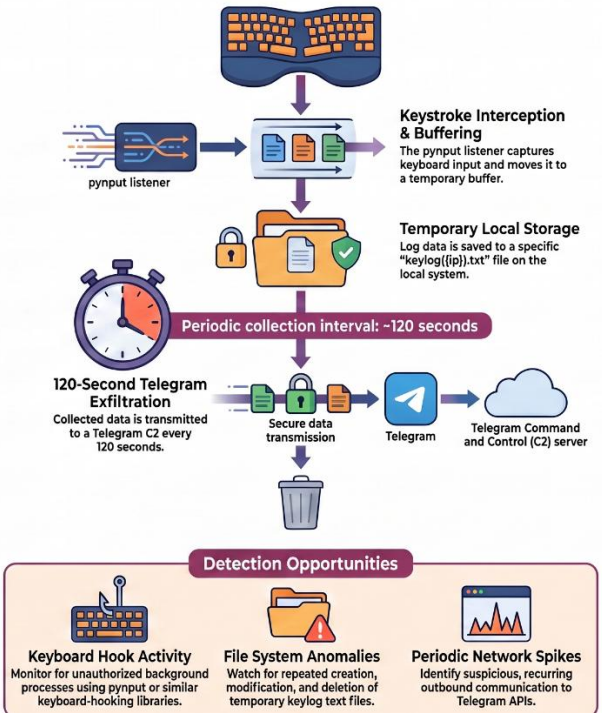



***Figure: NodeStealer Keylogging and Periodic Exfiltration***

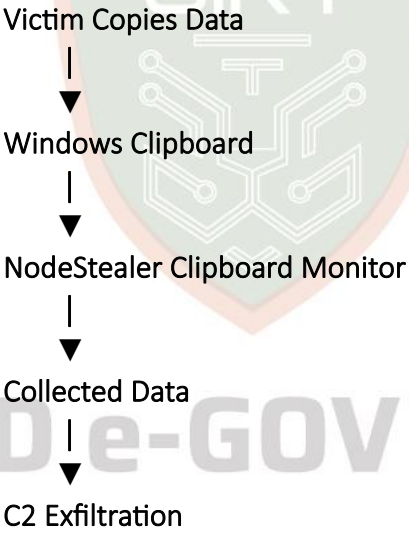
Consequently, password resets alone may be insufficient if an active keylogger remains on the endpoint.

## Clipboard Monitoring

The malware also introduces clipboard monitoring. A clipboard collector can capture information copied by the victim, including:

- Passwords
- API keys
- Tokens
- URLs
- Bank-account information
- Cryptocurrency addresses
- Internal documents
- Sensitive text

The potential attack chain is:






For organizations, clipboard monitoring is particularly important on systems used for:

- Banking
- MFS administration
- Software development
- Cloud administration
- Network management
- Database administration
- Cryptocurrency operations

## Screenshot Capture

The latest variant can capture screenshots of the victim's desktop. This expands the malware from data theft to visual surveillance. A screenshot may expose information that never exists as a file or browser credential, such as:

- Internal dashboards
- VPN portals
- Financial applications
- MFA prompts
- Administrative consoles
- Confidential documents
- Customer records
- Internal chat
- Security alert

## Browser Credential Theft

NodeStealer retains its historical focus on browser data.

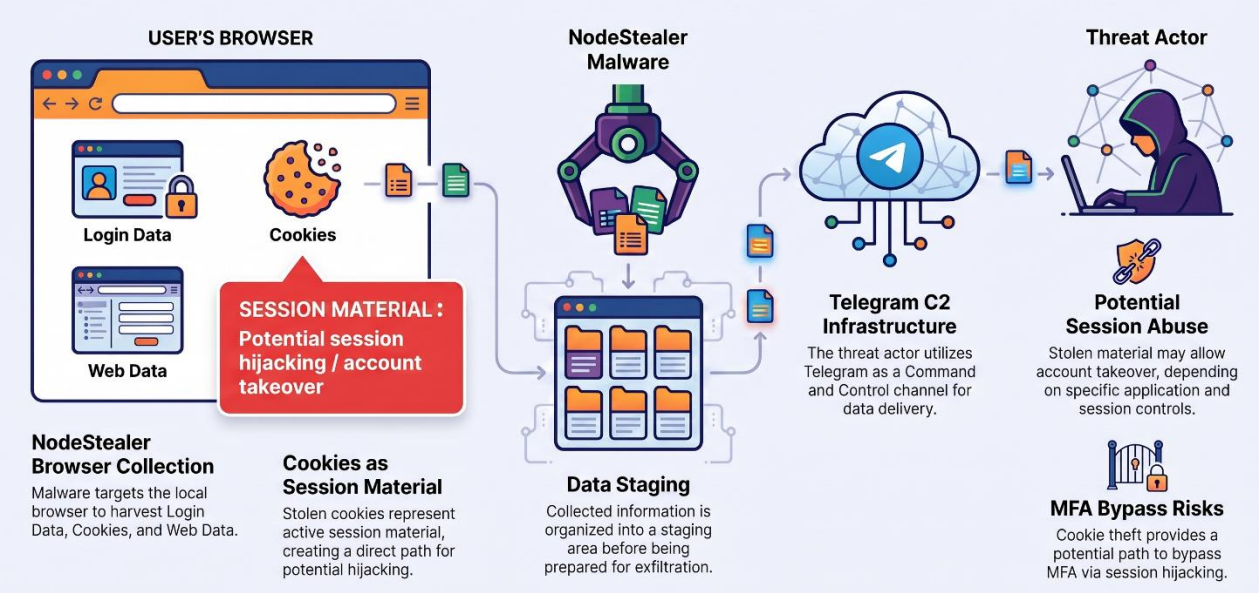



***Figure: Browser Credential and Session Theft***

Potential targets include:

- Stored credentials
- Cookies
- Session information
- Browser databases
- Autofill information
- Facebook-related credentials
- Financial-service information

The risk associated with browser cookies is particularly significant because **active session material can sometimes provide access without requiring the original password**.

A compromised browser therefore represents an identity-security problem, not merely a local endpoint problem.

## Wi-Fi Credential Theft

The latest variant also expands collection to Wi-Fi passwords. This creates additional risks in enterprise environments. In poorly segmented environments, recovered wireless credentials could potentially facilitate unauthorized network access or further reconnaissance.

Organizations should therefore avoid treating endpoint wireless credentials as low-value information.

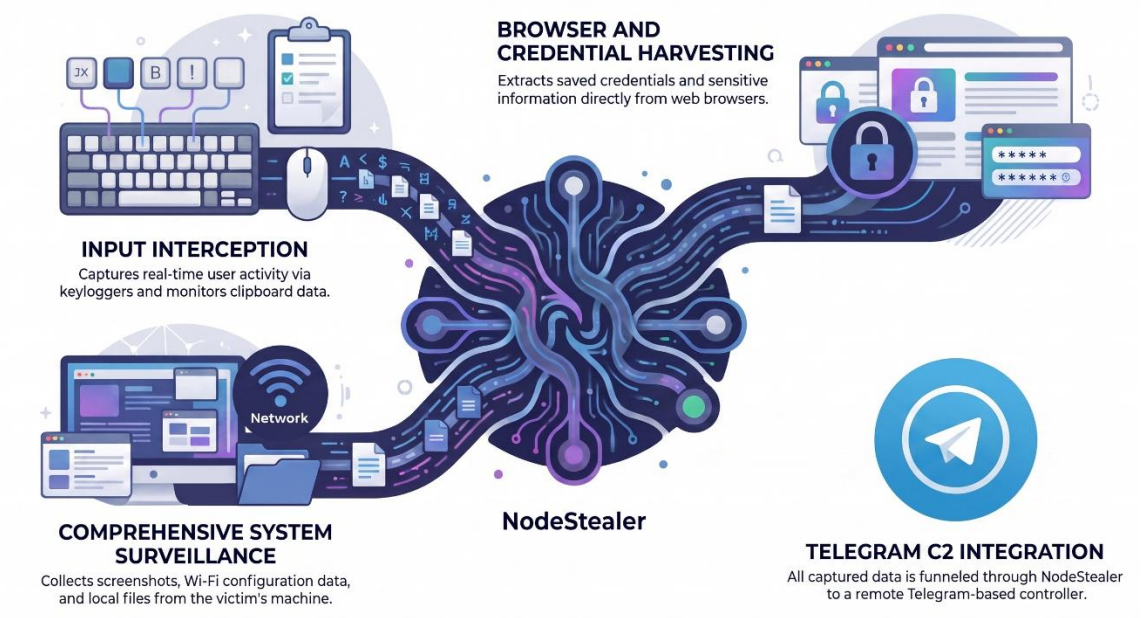



*Figure: Multi-Channel Surveillance Architecture*



## Pictures Directory Collection

The malware additionally expands its collection scope to the victim's Pictures directory. This can expose:

- Screenshots
- Scanned documents
- Identity documents
- Photographs
- Business documents
- Diagrams
- Credentials captured as images

The capability is important because sensitive information frequently exists outside conventional enterprise document repositories.

## Facebook Intelligence Collection

One of the most significant technical changes concerns Facebook. Earlier NodeStealer versions queried only **two Facebook Graph API endpoints**.

The new version queries **more than 20 endpoints**, allowing attackers to construct a much more comprehensive profile of the account operator.

The collection scope includes categories associated with:

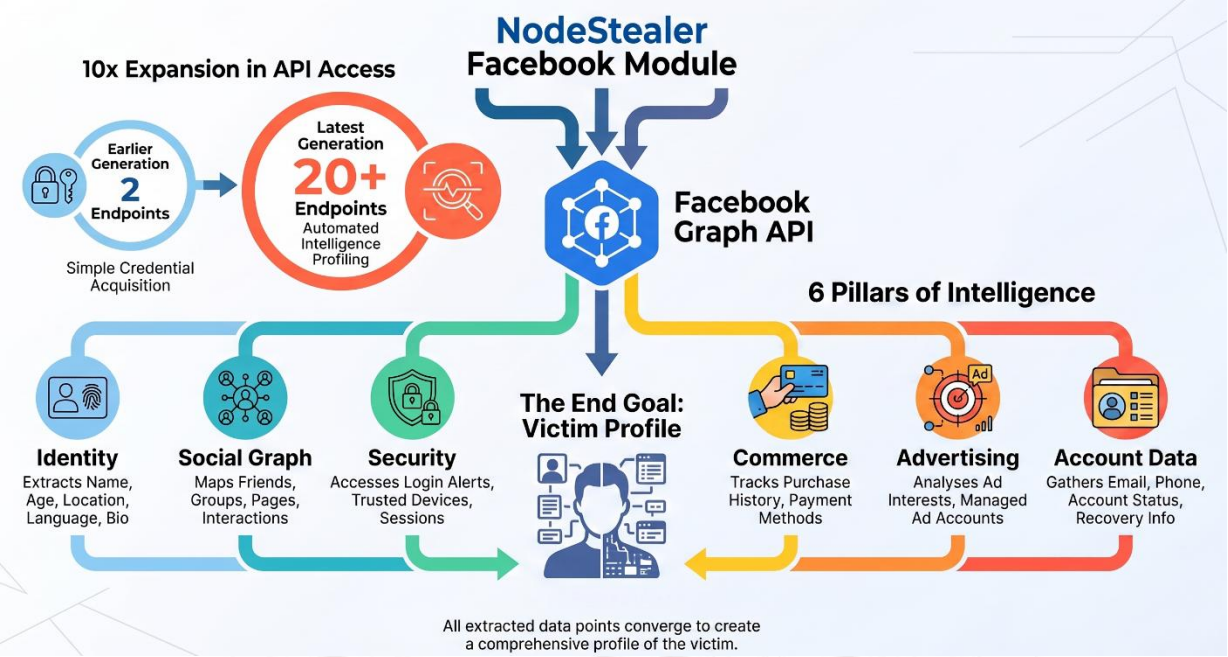



***Figure: Facebook Intelligence Collection***



## Facebook Account Profiling

The expanded collection can potentially allow threat actors to correlate:

- User Identity
- Friends / Social Connections
- Account Security
- Business Activity
- Advertising Activity
- Commerce Information

This information can support:

- Account takeover
- Impersonation
- Targeted phishing

- Business fraud
- Social engineering
- Advertising-account abuse
- Higher-value data resale

The source specifically identifies the potential for cross-platform account takeovers, impersonation scams and higher-value data resale.

## Facebook Ads Manager Risk

The malware's historical targeting of Facebook Ads Manager accounts remains particularly relevant to businesses and organizations that manage advertising campaigns.

Compromise could potentially enable an attacker to:


***Figure: Facebook/Business Account Takeover Scenario***

This could result in:

- Unauthorized advertising
- Financial loss
- Reputation damage
- Abuse of corporate identities
- Access to business information

Organizations using social-media advertising should therefore treat endpoint compromise as a potential **business-account security incident.**

## Telegram-Based C2 Architecture

The latest NodeStealer variant employs a **dual-bot Telegram command-and-control architecture**.

Conceptually:

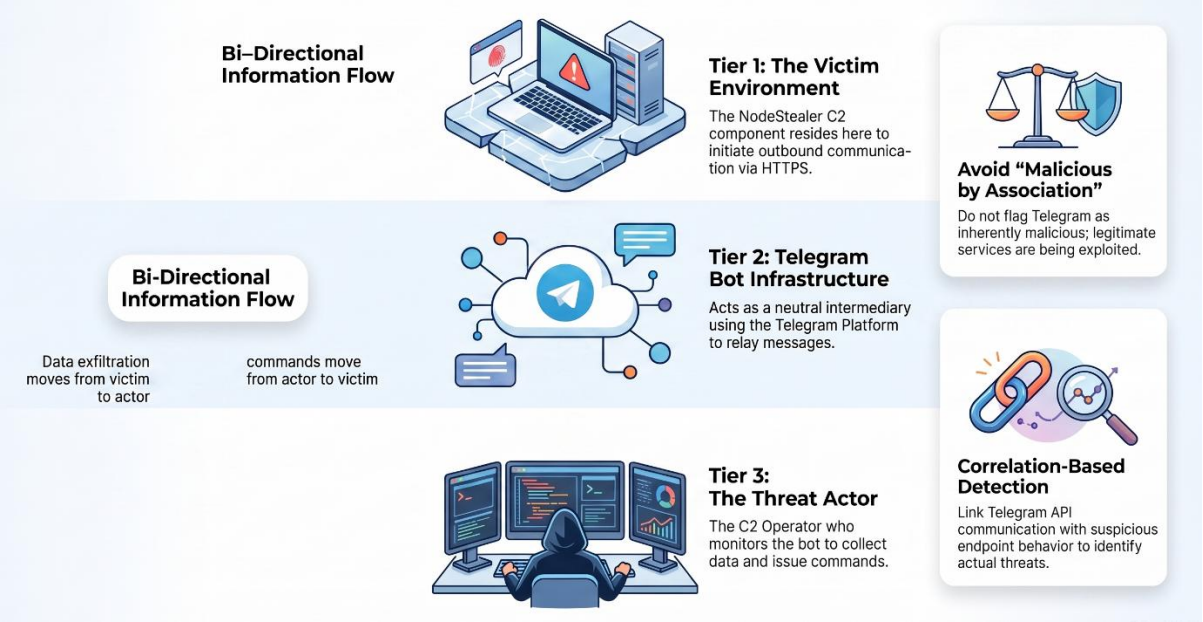



***Figure: Telegram-Based C2 Architecture***

Telegram provides several attractive characteristics to attackers:

- Widely available infrastructure
- Encrypted transport
- API-based automation
- Flexible bot functionality
- Easy message-based command handling

Consequently, simply blocking conventional malware domains may not be sufficient.

## Periodic Data Exfiltration

The keylogger demonstrates a periodic collection model.

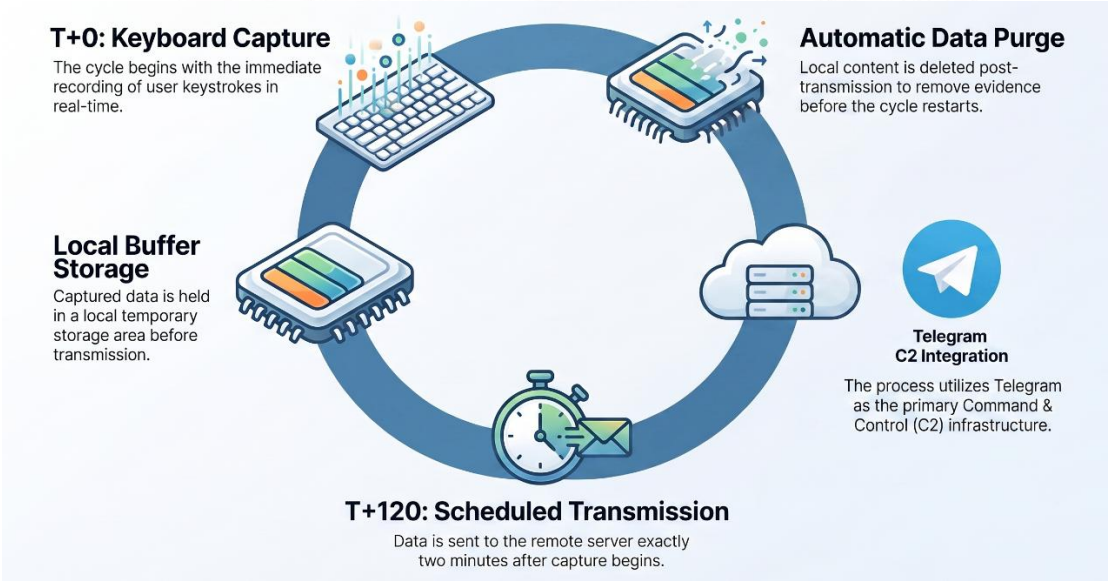



***Figure: Keyboard Data Exfiltration Cycle***

This creates a relatively small and repetitive network footprint.

Network defenders should therefore investigate **small recurring outbound connections** rather than focusing exclusively on high-volume data transfers.

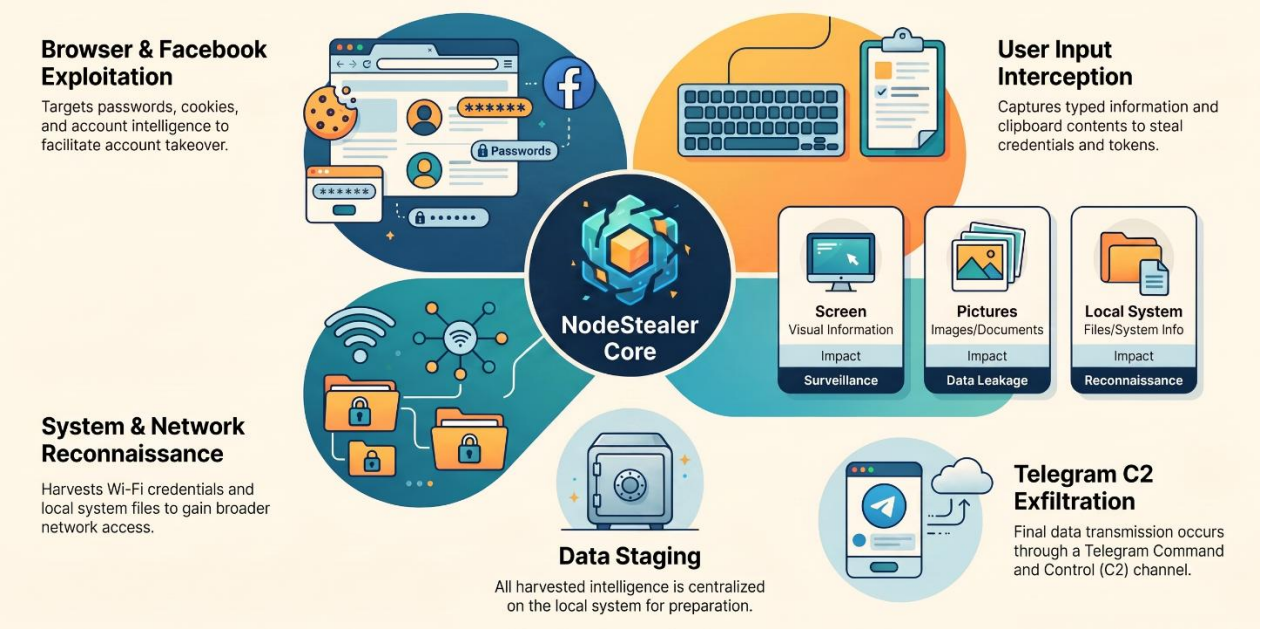



***Figure: NodeStealer Data Collection Matrix***

## AI-Assisted Malware Development

An unusual characteristic of the latest code is the systematic use of decorative emojis in output/logging.

The research assessed these characteristics as potentially consistent with **LLM-assisted coding**. The researchers noted that this behavior was absent from earlier NodeStealer variants. This development has broader implications.

AI-assisted development can potentially allow threat actors to:

- Rapidly add malware features
- Generate boilerplate code
- Modify existing modules
- Create platform-specific functionality
- Produce multiple variants
- Accelerate debugging and development

However, **AI-assisted attribution should not be treated as an IOC**. Code style alone cannot establish that AI was used. The practical defensive lesson is that defenders should expect **shorter malware development cycles** and more **frequent variant changes**.

## Bangladesh-Specific Threat Scenario

A potential Bangladesh-focused attack could target an employee through:

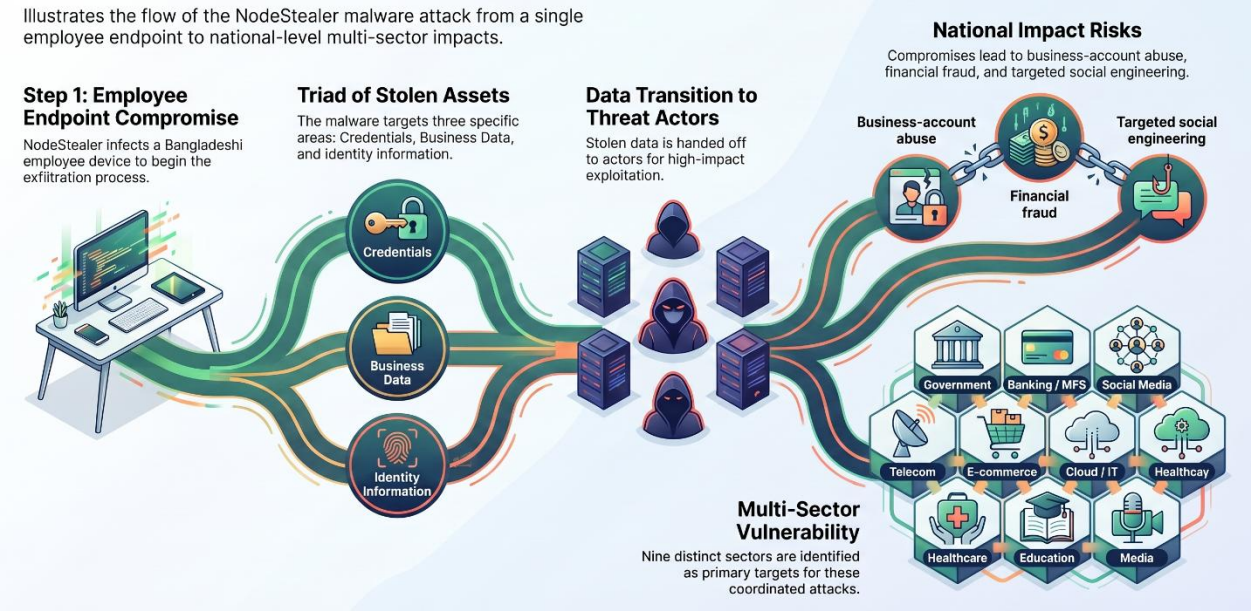



***Figure: Bangladesh Risk Scenario***

Again, this represents a **defensive threat model**, not evidence that this exact chain has been observed against Bangladesh.

## MITRE ATT&CK Mapping

| Tactic | Technique | ID |
|---|---|---|
| **Initial Access** | Phishing | T1566 |
| **Execution** | User Execution | T1204 |
| **Execution** | Command/Scripting Interpreter | T1059 |
| **Credential Access** | Credentials from Web Browsers | T1555.003 |
| **Credential Access** | Input Capture: Keylogging | T1056.001 |
| **Discovery** | System Information Discovery | T1082 |
| **Collection** | Screen Capture | T1113 |
| **Collection** | Clipboard Data | T1115 |
| **Collection** | Data from Local System | T1005 |
| **Collection** | Archive Collected Data | T1560 |

| | | |
|---|---|---|
| **Command & Control** | Application Layer Protocol | T1071 |
| **Command & Control** | Encrypted Channel | T1573 |
| **Exfiltration** | Exfiltration Over C2 Channel | T1041 |

## Indicators of Compromise (IoC)

### SHA256

- `f00be13bbfe31b19483bfeed8b53756f2bf3fc3d050d94cb079fa7365b322364`
- `7cfae56d84a39867048c9b44b4dcedd625661ce733959a3095838da470ad1892`
- `3e11bb1a3d00c271547b06fe49e47739b0dfcc71a7bed5336e75ab4b94a41f46`

### Telegram Bot Tokens

- `7094444204:AAFoaWZVfCF4ZyHvMpuAY0U15D3JlzxhNYg`
- `8842354283:AAFklEDGu7yjXAZajQG4cOaggcYtVtmqnsc`
- `7538597440:AAEsJistNXUaiM43icWerxRW1ZCROiqcx6Q`

### Telegram Chat IDs

- `5512231206`
- `7300493307`
- `6182684172`

### Artifacts

- `C:\Users\Public\number.txt`
- `C:\Users\Public\id.txt`
- `%TEMP%\keylog({ip}).txt`

## Detection and Monitoring

#### Endpoint Detection

Monitor for:

- Unexpected Python processes
- Unknown Python executables
- `python.exe` launched from user-writable directories
- Suspicious scripts in `%TEMP%`
- Keyboard-hook activity

- Unexpected screenshot APIs
- Clipboard monitoring
- Browser database access
- Access to Wi-Fi profile information
- Unexpected Telegram API communication

### Process-Tree Detection

High-risk examples include:

```
Outlook
  └── python.exe
    └── nodeStealer.py
```

or:

```
Browser
  └── python.exe
    └── suspicious script
```

or:

```
Explorer
  └── python.exe
    └── network connection
```

These parent-child relationships should be investigated in context.

### Browser Monitoring

EDR teams should monitor unusual access to browser profile directories. Examples include:

- Chrome User Data
- Edge User Data
- Firefox Profiles
- Browser Cookies
- Login Data
- Web Data
- Local Storage

A Python process accessing multiple browser databases should generate elevated scrutiny.

**Detection logic**:

Unknown Python Process + Browser Profile Access + Network Connection = High-Risk Event

### Network Detection

Monitor endpoints making connections to:

- Telegram infrastructure
- Newly observed external IP addresses
- Unusual HTTPS destinations
- Unknown API endpoints
- Repeated low-volume outbound connections

However, Telegram traffic alone is not sufficient to identify NodeStealer because legitimate users may use Telegram.

1. Detection should correlate:
2. Telegram Connection
3. Suspicious Python Process
4. Browser Data Access
5. Credential Collection

### Windows Event and EDR Telemetry

SOC teams should prioritize:

- Process creation
- Network connection events
- File creation
- File access
- PowerShell activity
- Script execution
- Persistence mechanisms
- Credential-access telemetry



Useful Windows telemetry includes:

| Event ID | Purpose |
|---|---|
| **4688** | Process creation |
| **4104** | PowerShell Script Block Logging |
| **7045** | Service installation |
| **4698** | Scheduled task creation |
| **4624** | Successful authentication |
| **4625** | Failed authentication |

EDR telemetry should supplement Windows event logs because much of the malware behavior occurs at the **process, file-access and network layers.**

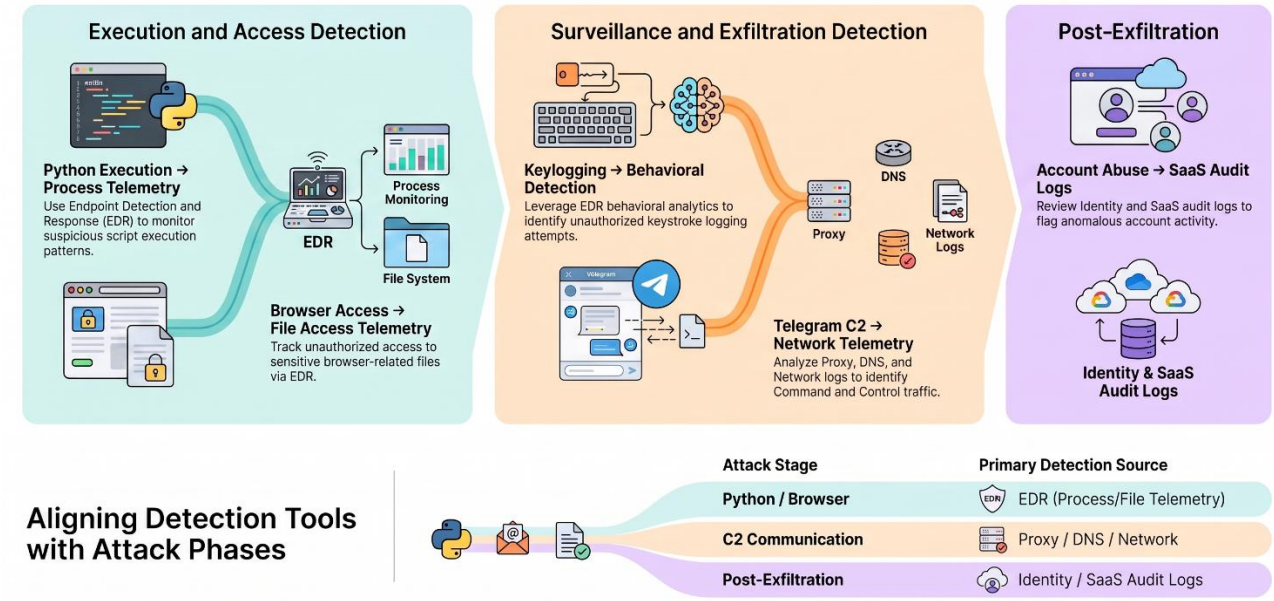



***Figure: Detection Opportunities Across the Attack Chain***

## SIEM Correlation Rules

Rule 1: Severity: HIGH
Unknown Python + Browser Database Access + External Network Connection

Rule 2: Severity: CRITICAL
Python Process + Keyboard Hook + Telegram Connection

Rule 3: Severity: CRITICAL
Python + Clipboard Access + Screenshot Capture + Outbound Connection

Rule 4: Severity: HIGH
Browser Credential Access + Facebook Account Access + New External Network Destination

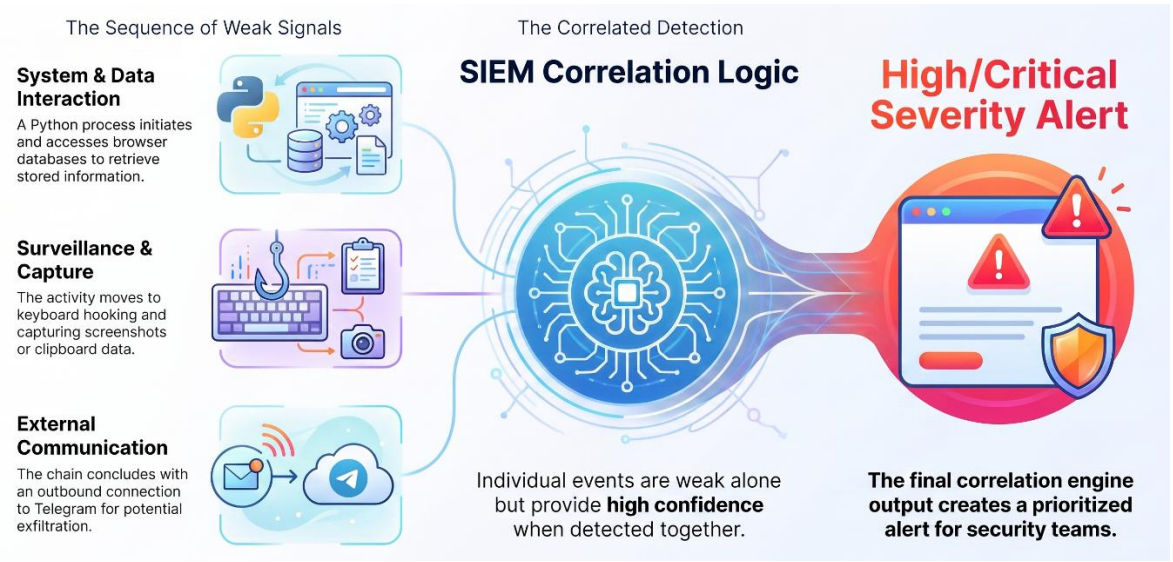



***Figure: SIEM Correlation Model***

# Incident Response Guidance

If NodeStealer infection is suspected:

## Immediate

- Isolate the endpoint from the network.
- Preserve volatile and forensic evidence.
- Identify the malware process and persistence.
- Determine the infection time.
- Identify all accounts accessed from the endpoint.

## Credential Response

Because the malware can capture keystrokes and browser information:

- Reset affected passwords.
- Revoke active sessions.
- Revoke browser sessions.
- Rotate API keys.
- Rotate VPN credentials.
- Reset privileged credentials.
- Review MFA sessions.
- Review social-media sessions.

## Business Account Response

For Facebook/business accounts:

- Revoke unknown sessions.
- Review administrators.
- Review business managers.
- Review advertising accounts.
- Review payment methods.
- Review unauthorized campaigns.
- Review recent account changes.

## Enterprise Investigation

Search the environment for:

- Same file
- Same process
- Same persistence
- Same C2
- Same Telegram communication
- Same browser-access pattern

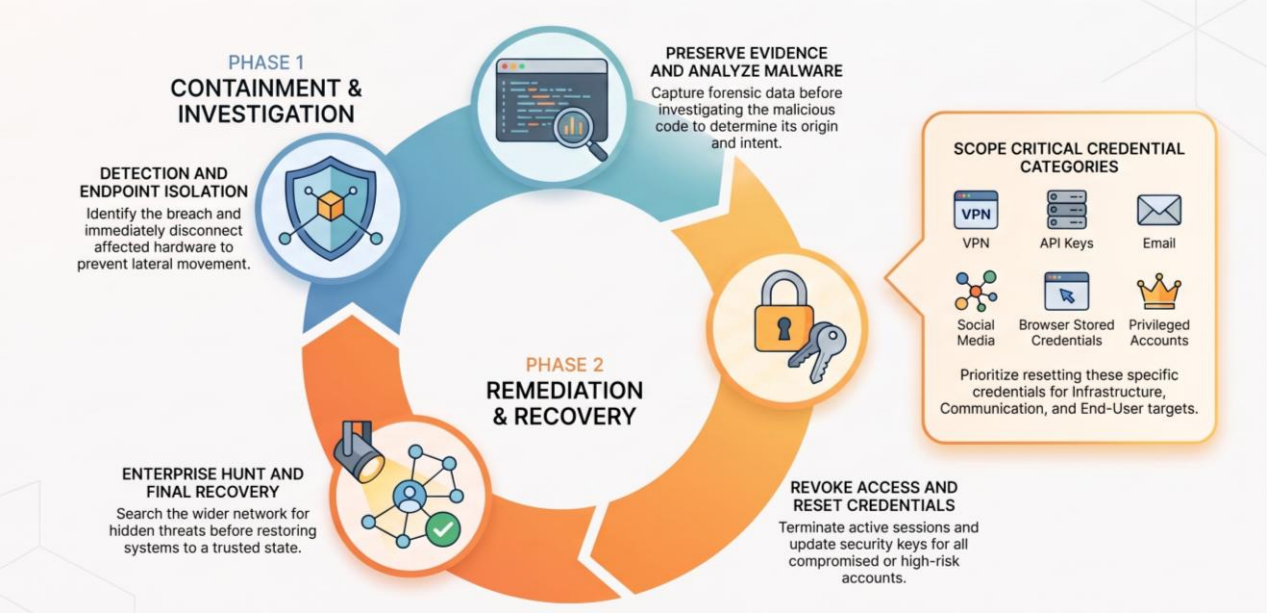



***Figure: Incident Response Workflow***

## Recommended Actions

- Immediately hunt for unauthorized Python-based executables and scripts.
- Monitor non-browser processes accessing browser credential databases.
- Deploy EDR detection for keylogging, clipboard monitoring and screenshot capture.
- Correlate suspicious Python execution with Telegram network activity.
- Review Facebook and business-account sessions from potentially compromised endpoints.
- Revoke active sessions when browser or cookie theft is suspected.
- Reset credentials entered while a suspected keylogger was active.
- Rotate API keys and privileged credentials associated with compromised endpoints.
- Review financial, advertising and payment accounts for unauthorized activity.
- Implement application allowlisting and restrict execution from user-writable directories.
- Strengthen phishing awareness, particularly for executable files and unsolicited business documents.
- Conduct environment-wide threat hunting when one confirmed infection is identified.
- Preserve forensic evidence before remediation where operationally possible.

The reported expansion of Python NodeStealer into a multi-capability spyware platform substantially increases the risk of credential theft, session hijacking, business-account compromise, targeted social engineering and financial fraud. Organizations in Bangladesh should prioritize behavior-based detection and identity/session protection, because file hashes and conventional malware signatures are unlikely to provide sufficient coverage against rapidly changing Python-based variants.

## References


- https://www.netskope.com/blog/python-nodestealer-ai-assisted-to-full-spyware
- https://unit42.paloaltonetworks.com/nodestealer-2-targets-facebook-business