TLP:WHITE | UNRESTRICTED DISTRIBUTION

THREAT INTELLIGENCE REPORT

Contagious Interview Campaign

Independent Analysis of the StegaBin Wave

Text Steganography, Dead-Drop C2, Typosquatting TTPs and Pivot Analysis

Attributed to Famous Chollima / Lazarus Group

Field	Value
Report ID	CTI-001 | StegaBin wave
Report Date	March 9, 2026
Analyst	SerapHim
Analyst Role	Independent Cyber Threat Intelligence
Campaign	Contagious Interview (ongoing since mid-2025)
Wave / Moniker	StegaBin (designated by Socket Security, Feb 27, 2026)
Threat Actor	Famous Chollima (Lazarus Group)
Attribution Confidence	High
Malware Family	Cross-Platform RAT and 9-Module Credential Stealer
Wave Discovery	February 25-26, 2026
Infrastructure Status	Taken down. npm packages removed, Pastebin pastes offline
Analyst Implementation	Independent Rust-based decoder (partial, pre-takedown)
TLP Classification	TLP:WHITE, Unrestricted Distribution
MITRE ATT&CK	T1195.001, T1027.003, T1102.001, T1059.004, T1547, T1555.003, T1552.001, T1041, T1056.001

1. Executive Summary

This report documents an independent analysis of the StegaBin wave, one iteration within the ongoing Contagious Interview supply chain campaign operated by Famous Chollima, a threat cluster attributed to the DPRK and tracked under the broader Lazarus Group umbrella.

The StegaBin wave was first detected on February 25-26, 2026. Its defining technical characteristic is the use of character-level text steganography embedded in Pastebin pastes to conceal command-and-control infrastructure addresses. This method replaces earlier Contagious Interview delivery mechanisms that used direct Bitbucket-hosted payloads, and it represents a measurable increase in evasion capability.

This analysis was conducted independently against live infrastructure in the approximately 90-minute window before full takedown on March 8, 2026. A Rust-based decoder was written and executed during this window. The steganographic structure was confirmed, the length marker and character substitution pattern were documented directly from fetched content, and a decode was attempted. The attempt failed due to an offset calculation error that was identified post-analysis and is documented in full in Section 4.6.

Section 10 contains original pivot analysis across all primary artifacts from this campaign. No equivalent analysis was found in prior public reporting at time of publication.

2. Campaign and Threat Actor Background

Famous Chollima is a DPRK-affiliated threat cluster operating under the Lazarus Group umbrella. The group has a documented history of financially motivated operations, with particular focus on cryptocurrency theft, blockchain-adjacent targeting, and software supply chain compromise.

Contagious Interview is an ongoing campaign by this actor, active since at least May 2025. It targets software developers through malicious packages published to public registries, primarily npm. The social engineering pretext typically involves fake technical interview scenarios or coding assessments that lead the target to install a malicious package as part of the interaction.

The StegaBin moniker refers specifically to this wave of the campaign, named for its use of steganographic Pastebin pastes as a dead-drop resolver. It does not refer to the broader campaign.

3. Wave Overview

3.1 Timeline

Date	Event
Feb 25-26, 2026	26 malicious npm packages published across throwaway accounts
Feb 26, 2026	Kieran Miyamoto (kmsec.uk) independently discloses 17 packages with steganographic decoder walkthrough
Feb 27, 2026	Socket Security publishes full disclosure, names the wave StegaBin, identifies all 26 packages, captures 9-module toolkit from live C2
Mar 2, 2026	The Hacker News coverage
Mar 8, ~02:30	Independent analysis begins. Fetch 1: primary URL (CJ5PrtNk) accessible, length marker 00972 identified
Mar 8, ~02:45	Fetch 2: Unicode inspection of first 10 characters, CRLF boundary at index 5-6 documented, essay body confirmed starting at index 7
Mar 8, ~03:00	Fetch 3: decode attempt. Offset error produces interval 0, output is 972 zero characters
Mar 8, ~03:00+	Fallback URLs 0ec7i68M and DjDCxcsT return HTTP 404. Full takedown confirmed
Mar 8, ~04:xx	All three Pastebin pastes offline. Total analysis window approximately 90 minutes
Mar 9, 2026	Report published

3.2 Package Distribution Strategy

26 malicious packages were published across 26 separate npm accounts over two days. Each package is a typosquat of a widely-used library. Eight of the 26 packages carry a -lint suffix, positioning them as plausible developer linting tooling.

Each malicious package declares the legitimate package it impersonates as a listed dependency. This means the victim project continues to function after the package is installed, which delays detection while the malicious install script runs in the background.

Socket identified three account persona clusters across the 26 accounts:

- christopher.smith.*47

- andrew.*walker*
- joni*

The remaining 11 accounts appear as singletons with no common pattern.

4. Technical Analysis

4.1 Infection Chain

- **Stage 1.** The npm install lifecycle hook executes install.js automatically on package installation. This loads vendor/scrypt-js/version.js, a path constructed to appear as a vendored cryptographic dependency.
- **Stage 2.** version.js is obfuscated using RC4 string encryption, array rotation, anti-debug routines, and control flow flattening. It fetches Pastebin URLs and extracts C2 addresses from character-level substitutions embedded in essay-format text.
- **Stage 3.** A platform-specific shell payload is fetched from a Vercel C2. macOS uses curl piped to sh, Linux uses wget piped to sh, Windows uses curl piped to cmd.
- **Stage 4.** A single-use token is appended to the second request. Node.js 20.11.1 is installed if absent. parser.js and package.json are downloaded from C2.
- **Stage 5.** parser.js connects to 103[.]106[.]67[.]63:1244. The C2 deploys a 9-module credential stealer automatically.

4.2 Dead-Drop Resolver Structure

The loader hardcodes three Pastebin URLs in a fallback chain. If the primary paste is removed, the loader falls through to the fallback URLs. All three pastes resolve to the same set of 31 Vercel-hosted C2 domains.

```
Primary:   hxxps://pastebin[.]com/raw/CJ5PrtNk  (user: davidsouza23, 353
views)
Fallback1: hxxps://pastebin[.]com/raw/0ec7i68M  (user: Edgar04231,   15
views)
Fallback2: hxxps://pastebin[.]com/raw/DjDCxcsT  (user: Edgar04231,   19
views)
```

4.3 Paste Structure and Length Marker

The following was obtained directly from a live fetch of the primary paste prior to takedown.

```
[*] fetch: hxxps://pastebin[.]com/raw/CJ5PrtNk
[+] body length: 15191 chars
[*] inspect first 10 chars (Unicode):
  [0] U+0030  0
  [1] U+0030  0
  [2] U+0039  9
  [3] U+0037  7
  [4] U+0032  2
  [5] U+000D  CR
  [6] U+000A  LF
  [7] U+0054  T
  [8] U+0068  h
  [9] U+0065  e
```

Indices 0-4 form the string 00972, a zero-padded 5-digit length marker encoding the count of characters hidden in the essay body. The CRLF at indices 5-6 separates the header from the carrier text. The essay begins at index 7.

4.4 Character Substitution Pattern

Analysis of the first 200 characters of the essay body identified single-character substitutions at non-random intervals. The essay title is "The Evolution of Programming Languages: From Machine Code to Modern Frameworks". Substitutions observed:

Observed	Expected	Substituted char
evelution	evolution	v
Progxamming	Programming	x
Languagts	Languages	t
Machin-	Machine	-
Moderc	Modern	c
Phogramming	Programming	h
Langeages	Languages	e
undecgone	undergone	c
remarkakle	remarkable	k
transformatdon	transformation	d
diwn	dawn	w
computinn	computing	n

4.5 Decoding Algorithm

- Strip zero-width Unicode characters (U+200B, U+FEFF) from the full response body.
- Read the 5-digit length marker from positions 0-4. In this sample msg_len = 972.
- Skip the CRLF at indices 5-6. The carrier text begins at index 7.
- Compute interval: carrier_length = total_body_length minus 7 = 15184, interval = floor(15184 / 972) = 15.
- Extract one character at each interval position throughout the carrier text.
- Concatenate extracted characters, split on ||| separator, terminate at ===END=== to produce an array of 31 Vercel C2 domains.

The CRLF boundary at indices 5-6 must be excluded from the carrier length. Implementations that include it will compute an incorrect interval and produce garbled or null output.

4.6 Independent Analysis: Offset Error

```
[*] fetch 3: hxxps://pastebin[.]com/raw/CJ5PrtNk
[+] length after strip: 15191 chars
[*] header: "00972"
[+] N (msg_len): 972
[+] carrier_length: 5     <- ERROR: header length passed instead of body
length
[+] interval: 0           <- ERROR: floor(5 / 972) = 0
[+] output (100 chars):
    0000000000000000000000000000000000000000... (972 zeros)
```

The variable carrier_length was set to 5, the length of the header string "00972", rather than 15184, the actual length of the essay body after the CRLF offset. With interval at 0 the decoder sampled the same index on every iteration and produced 972 zero characters. The correct value is 15184, yielding interval 15. This error was identified after the analysis session. By that point the infrastructure was offline and the corrected decode could not be attempted.

4.7 Fallback Fetch Results and Takedown Confirmation

```
Fallback 1: hxxps://pastebin[.]com/raw/0ec7i68M  -> HTTP 404
Fallback 2: hxxps://pastebin[.]com/raw/DjDCxcsT  -> HTTP 404
```

The primary URL remained accessible across all three fetch attempts during the session. Both fallback URLs were already returning 404 by the time they were first tried. Full takedown of the primary URL was confirmed at approximately 04:xx on March 8, 2026.

5. 9-Module Infostealer Toolkit

On RAT connection to 103[.]106[.]67[.]63:1244, the C2 automatically deploys nine modules. The following module details were captured by Socket Security via a simulated compromised client against live C2 infrastructure prior to takedown.

Module	SHA-256 (trunc)	Function
01 vs	ddbb527b...	VS Code persistence via malicious tasks.json. Uses runOn:folderOpen. 186-space whitespace pushes malicious command off-screen in the VS Code task viewer. Cross-platform.
02 clip	ce80100a...	Keylogger, mouse tracker, and clipboard stealer. Encrypts captured data with AES-256-CBC. Exfiltrates every 10 minutes to port 1244/clipup.
03 bro	714f8903...	Python browser credential stealer. Payload is Base85-encoded and XOR-obfuscated with key Xt3rqfmL.
04 j	1b15f730...	Node.js crypto wallet and browser stealer targeting 86 wallet extension IDs including MetaMask, Phantom, Coinbase, Binance, Trust, Exodus, and Keplr. Also targets iCloud Keychain on macOS.
05 z	ba3e520b...	Filesystem sweep using 17 glob patterns targeting *.wal, *.key, *.mne, *.env, *.kdb, *.sol and others. Uses HMAC-SHA256 deduplication to avoid re-uploading files.
06 n	78c8b904...	Interactive RAT with persistent WebSocket connection to port 1247. Supports FTP bulk exfiltration via basic-ftp and can kill security processes on command.
07 truffle	4e6a7bf3...	Downloads TruffleHog v3.92.5 from GitHub, scans the full home directory and all drives, uploads results to port 1244/uploads, then deletes evidence.
08 git	978e8f16...	SSH key and Git credential theft. Collects all .ssh files, parses .git-credentials, and extracts tokens from remote URLs. Scans to depth 20.
09 sched	da1775d0...	StegaBin loader redeployed as a persistence mechanism. Identical SHA-256 to vendor/scrypt-js/version.js, confirming same file reused.

Shared identifiers across all modules:

- Member identifier: THKASDFOWG
- Encryption password: Angelisbadguy@#!
- Upload endpoints: /clipup and /uploads

6. Indicators of Compromise

6.1 Network Infrastructure

Type	Indicator	Notes
IP	103[.]106[.]67[.]63	C2 RAT port 1244, WebSocket port 1247 (AS23470, ReliableSite.Net LLC)
Pastebin	pastebin[.]com/raw/CJ5PrtNk	Primary dead-drop. User: davidsouza23. 353 views.
Pastebin	pastebin[.]com/raw/0ec7i68M	Fallback 1. User: Edgar04231. 15 views. 404 at analysis time.
Pastebin	pastebin[.]com/raw/DjDCxcsT	Fallback 2. User: Edgar04231. 19 views. 404 at analysis time.
Domain	ext-checkdin[.]vercel[.]app	Active Vercel C2 at time of Socket analysis
Domain	cleverstack-ext301[.]vercel[.]app	Vercel C2, inactive at analysis time
Domain	brightlaunch-ext742[.]vercel[.]app	Vercel C2, inactive
Domain	primevector-ext483[.]vercel[.]app	Vercel C2, inactive
Note	+27 additional Vercel domains	Full list in Socket Security disclosure. 31 domains total.

6.2 Malicious npm Packages

Package	Package	Package
argonist@0.41.0	bcryptance@6.5.2	bee-quarl@2.1.2
bubble-core@6.26.2	corstoken@2.14.7	daytonjs@1.11.20
ether-lint@5.9.4	expressjs-lint@5.3.2	fastify-lint@5.8.0
formmiderable@3.5.7	hapi-lint@19.1.2	iosysredis@5.13.2
jslint-config@10.22.2	jsnwebapptoken@8.40.2	kafkajs-lint@2.21.3
loadash-lint@4.17.24	mqttoken@5.40.2	prism-lint@7.4.2
promanage@6.0.21	sequelization@6.40.2	typoriem@0.4.17
undicy-lint@7.23.1	uuindex@13.1.0	vitetest-lint@4.1.21
windowston@3.19.2	zoddle@4.4.2	

6.3 File and String Indicators

Type	Indicator	Notes
File path	vendor/scrypt-js/version.js	Primary malicious payload. Present in all 26 packages.
File path	scripts/test/install.js	npm lifecycle hook that triggers the loader.
File path	tasks.json (VS Code)	Persistence target. Written to VS Code user config directory.
SHA-256	da1775d0fbe99fbc35b6f0b4a3a3cb84...	version.js and sched module share this hash, confirming same file.

Type	Indicator	Notes
String	THKASDFOWG	Shared member identifier across all 9 modules.
String	Angelisbadguy@#!	Shared encryption password across all 9 modules.

7. MITRE ATT&CK Mapping

ID	Technique	Observed
T1195.001	Supply Chain Compromise: Software Dependencies	26 malicious packages published to npm registry
T1027.003	Obfuscated Files: Steganography	C2 addresses encoded via character substitution in Pastebin essay text
T1102.001	Web Service: Dead Drop Resolver	Pastebin used as intermediary to resolve 31 Vercel C2 domains
T1059.004	Command Interpreter: Unix Shell	Platform-specific shell scripts fetched from Vercel C2
T1547	Boot/Logon Autostart Execution	VS Code tasks.json persistence via runOn:folderOpen
T1555.003	Credentials from Web Browsers	Chrome, Firefox, Edge, Brave, Opera targeted. iCloud Keychain on macOS.
T1552.001	Unsecured Credentials: Files	TruffleHog weaponized to scan filesystem for API keys and secrets
T1041	Exfiltration Over C2 Channel	WebSocket to port 1247, FTP bulk upload, HTTP POST to port 1244
T1056.001	Input Capture: Keylogging	WH_KEYBOARD_LL on Windows, xinput on Linux, CGEventTap on macOS

8. Analysis Limitations

This report is published with full transparency regarding the boundaries of the analysis conducted.

- **Partial decode.** The steganographic structure was confirmed directly from fetched content. The decode attempt failed due to an offset calculation error documented in Section 4.6. The corrected decode was not re-attempted before infrastructure went offline.
- **Infrastructure window.** The analysis window was approximately 90 minutes. Both fallback URLs were already offline before they were first attempted. The primary URL was accessible throughout the session.
- **Module analysis.** Section 5 documents the 9-module toolkit as captured by Socket Security via a simulated compromised client. This was not replicated independently.
- **Pivot scope.** Section 10 pivot analysis used passive open-source intelligence only. No active probing of infrastructure was performed.

9. Defensive Recommendations

- **npm audit.** Review installed packages for typosquatting against well-known libraries, particularly those using -lint, -token, or -ation suffixes. Automated tools such as Socket.dev can assist.

- **Network monitoring.** Alert on outbound connections to pastebin.com initiated by Node.js processes or development tooling. Pastebin is a legitimate service and is unlikely to be blocked at the network perimeter, which is what makes it viable as a dead-drop.
- **VS Code task audit.** Review the global tasks.json file for unexpected runOn:folderOpen entries. The 186-space whitespace pattern pushes malicious commands off-screen in the VS Code interface.
- **Secrets hygiene.** Do not store API keys, SSH private keys, or plaintext credentials in filesystem locations reachable by development tooling. TruffleHog is a legitimate scanning tool that is weaponized in this campaign to locate and exfiltrate exactly these files.
- **Developer awareness.** Unsolicited technical interview requests arriving via LinkedIn, GitHub, or Reddit that require package installation should be treated with suspicion. This is the documented initial access pattern for Contagious Interview.
- **ASN awareness.** Connections from development environments to AS23470 (ReliableSite.Net LLC) on ports 1244 or 1247 should be investigated.

10. Pivot Analysis

This section contains original analysis conducted after the campaign infrastructure was taken down. All pivot work used passive open-source intelligence. No equivalent analysis was found in prior public reporting at time of publication.

10.1 C2 IP: 103[.]106[.]67[.]63

The IP is hosted under AS23470, registered to ReliableSite.Net LLC, a US-based dedicated server provider headquartered in Miami, FL. The ASN was registered under ARIN on August 10, 2018, and operates approximately 60,945 IPv4 addresses across 177 announced prefixes.

Scamalytics rates AS23470 at a fraud score of 20 out of 100, placing it in the medium-risk category. Approximately 20% of observed traffic from this ASN is assessed as potentially fraudulent. This is consistent with a commodity shared-hosting environment where legitimate customers coexist with abusive tenants.

No prior public documentation was found linking AS23470 to Famous Chollima or Lazarus Group infrastructure before this campaign. This absence is worth noting. The actor appears to have selected a provider with no established reputation for hosting nation-state tooling, which reduces the likelihood that outbound connections to this ASN would trigger reputation-based blocks at the network perimeter.

The C2 operated on ports 1244 and 1247. These are non-standard ports that avoid common inspection points targeting 80 and 443, but are not unusual enough to trigger generic high-port alerting in most enterprise environments. This is a deliberate positioning choice.

Deeper infrastructure pivot against this IP would require access to historical passive DNS, certificate metadata, or port scan history from platforms such as Shodan or Censys. That data was not available at the access level used for this analysis. Historical relationships between this IP and other domains or certificates may exist and are not ruled out.

10.2 Pastebin Personas: davidsouza23 and Edgar04231

The dead-drop chain used two Pastebin accounts. The primary paste was registered to davidsouza23 and accumulated 353 views before takedown. Both fallback pastes were registered to a single account, Edgar04231, with 15 and 19 views respectively.

The view count difference is significant. The primary paste at 353 views received substantially more traffic than either fallback, which is consistent with the primary URL being broadly distributed to victims or ingested by automated analysis systems, while the fallback URLs remained mostly untouched. The fallbacks appear to have served a redundancy function that was never needed.

Both fallback pastes sharing a single account is a notable pattern. If that account were identified and suspended before takedown, both fallback URLs would have been neutralized simultaneously. This contradicts the redundancy design intent and represents a minor operational security inconsistency relative to the otherwise methodical structure of the delivery chain.

Both account names follow a firstname-surname-number format consistent with synthetic Western developer personas. No prior Pastebin activity was found for either account in public sources, which suggests they were registered specifically for this campaign rather than repurposed from prior infrastructure.

10.3 Member Identifier: THKASDFOWG

The string THKASDFOWG appears as a shared identifier across all nine post-compromise modules. No prior occurrence of this string was found in public threat intelligence reporting, malware databases, or general web search at time of analysis.

The string is 10 characters, all uppercase Latin, with no numeric or special characters. It does not correspond to any known English word, acronym, or documented identifier. It does not appear to be a hash fragment or encoding artifact.

The most probable function is a campaign-specific or operator-specific session token used to tag exfiltrated data on the C2 side. A shared identifier across all nine modules allows the operator to correlate data received through different exfiltration channels back to a single victim. This is a common design pattern in modular infostealer frameworks.

The character set (T, H, K, A, S, D, F, O, W, G) draws entirely from the QWERTY keyboard home row and adjacent rows. This is not a standard keyboard walk, but the composition is consistent with either a semi-random internal codename or a manually typed string with no particular semantic intent.

If this identifier or a derivative form appears in future Contagious Interview samples, it becomes a behavioral link between campaign waves. It is recommended that analysts monitoring npm packages or analyzing DPRK-attributed malware flag this string in automated detection rules.

10.4 Encryption Password: Angelisbadguy@#!

The string Angelisbadguy@#! is used as the encryption password across all nine modules, including AES-256-CBC for clipboard data and XOR for the browser credential stealer. This artifact has not been analyzed in depth in any prior public reporting found at time of publication.

The password is structured as a readable English phrase: "Angel is a bad guy" followed by a special character suffix. The @#! suffix is a common complexity-padding pattern used in manually authored passwords. The substantive portion of the credential is the phrase itself.

The use of a proper name, "Angel", at the start of an operational encryption password is uncommon in malware authored for production use. Credentials of this kind are typically randomized or generated. A human-readable phrase with a name in it suggests the string was manually chosen, possibly referencing an internal operator handle, a target, or an internal reference with meaning within the team that created it.

SentinelOne has previously documented that Contagious Interview operators use identifiable human names in internal testing infrastructure, including persona names found in artifact strings. The presence of "Angel" in this credential is consistent with that behavioral pattern, though a direct connection cannot be confirmed from open-source data alone.

If this password or structural derivatives such as angel_badguy, angelisbad, or similar appear in future DPRK-attributed malware or in credential material from compromised infrastructure, it would constitute a high-confidence link between campaigns and potentially between individual operators.

10.5 npm Account Persona Patterns

Of the 26 npm accounts used, Socket identified 15 as clustering into three named persona patterns. The remaining 11 appear as singletons.

Pattern	Count	Notes
christopher.smith.*47	~5 accounts	Firstname, surname fragment, numeric suffix. Consistent format.
andrew.*walker*	~5 accounts	Firstname, variable middle segment, surname. Wildcard insertion point.
joni*	~5 accounts	Single given name prefix. Simpler pattern.
11 singletons	1 each	No shared pattern. Likely deliberate to reduce pattern-based detection.

The clustered personas follow formats consistent with scripted account generation using a name template with variable fields. The christopher.smith.*47 pattern is particularly notable because of the consistent *47 numeric suffix. If additional packages from this actor are discovered carrying a *47 suffix in the npm username, that suffix becomes a reliable tracking artifact independent of the account name.

The singleton accounts serve a functional purpose within the distribution strategy. Pattern-based detection that successfully identifies and blocks the christopher.smith.* cluster would not affect accounts with no matching pattern. The singletons act as fallback delivery points that survive pattern-based filtering.

All identified persona names use Western English names, consistent with the social engineering pretext of the campaign, which involves fake Western recruiter or developer identities.

10.6 File Path: vendor/scrypt-js/version.js

This file path is present in all 26 packages as the primary malicious payload, and reappears as the sched persistence module, where it carries an identical SHA-256 hash (da1775d0...), confirming it is the same file reused in both roles.

The path is constructed to resemble a vendored cryptographic dependency. scrypt-js is a real, widely-used npm package that implements the scrypt key derivation function. Vendoring dependencies by copying them into a /vendor directory is a legitimate and common JavaScript development practice. The filename version.js is plausible within that context, as version information is often separated into its own file. The combination passes casual inspection and would not stand out in a standard code review.

This path is a reliable file-based IOC. Its consistent presence across all 26 packages, combined with its reuse as the persistence module, makes it suitable for automated detection. A pre-install audit rule targeting the combination of scripts/test/install.js and vendor/scrypt-js/version.js in the same package would have blocked this campaign at the installation stage.

10.7 Summary

No direct infrastructure overlap was found between the artifacts in this campaign and prior publicly documented Contagious Interview waves. This absence is consistent with the actor's documented pattern of aggressively rotating infrastructure between campaigns to prevent cross-campaign attribution through IP or domain reuse.

The artifacts that carry the most value for long-term tracking are THKASDFOWG and Angelisbadguy@#!, both of which appear to be manually authored rather than generated. If either string or a derivative appears in future DPRK-attributed samples, the link would be high-confidence. The *47 npm account suffix is a secondary tracking artifact worth monitoring in automated package scanning.

The operational security of this campaign is generally high at the infrastructure and delivery level, but shows inconsistency in artifact management. Human-readable credentials, consolidated fallback accounts under a single persona, and a consistent malicious file path are the points where the actor left durable tracking material.

11. References

- [1] Socket Security, Burckhardt P. and van der Zee P. StegaBin: 26 Malicious npm Packages Use Pastebin Steganography to Deploy Multi-Stage Credential Stealer. socket.dev/blog. February 27, 2026.
- [2] Miyamoto K. (kmsec.uk). DPRK Text Steganography. kmsec.uk/blog/dprk-text-steganography. February 26, 2026.
- [3] The Hacker News. North Korean Hackers Publish 26 npm Packages Hiding Pastebin C2 for Cross-Platform RAT. thehackernews.com/2026/03. March 2, 2026.
- [4] Socket Security. Contagious Interview Supply Chain Attacks Tracker. socket.dev/supply-chain-attacks.
- [5] MITRE ATT&CK. Lazarus Group (G0032). attack.mitre.org/groups/G0032.
- [6] SentinelOne Labs. Contagious Interview Threat Actors Scout Cyber Intel Platforms. September 4, 2025.
- [7] Scamalytics. AS23470 ReliableSite.Net LLC Fraud Risk Assessment. scamalytics.com.
- [8] ARIN WHOIS. AS23470 ReliableSite.Net LLC. rdap.arin.net. Accessed March 2026.