Hive Pro

Threat Level

Amber

HiveForce Labs

THREAT ADVISORY

ATTACK REPORT

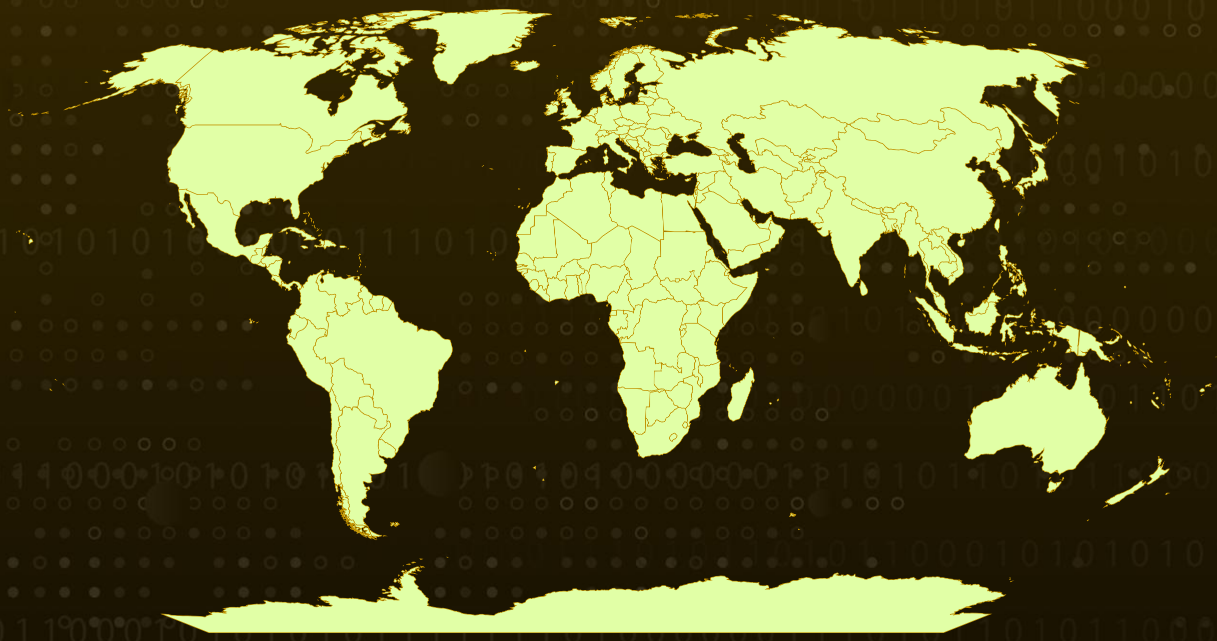
Typosquatted npm Packages Execute Stealthy Credential Theft Operation

Date of Publication	Admiralty Code	TA Number
October 31, 2025	A1	TA2025334

Summary

Attack Discovered: July 4, 2025
Targeted Region: Worldwide
Affected Platforms: Windows, Mac, Linux
Attack: A cluster of ten malicious npm packages, secretly published by a threat actor known as andrew_r1, carried out a deceptive supply-chain attack that blended social engineering with advanced obfuscation and cross-platform data theft. Masquerading as legitimate libraries, such as TypeScript and Discord, these packages tricked developers with fake CAPTCHAs and realistic installation prompts before unleashing a 24MB information stealer capable of harvesting credentials from system keyrings, browsers, and authentication services across Windows, Linux, and macOS. With over 9,900 downloads, the campaign showcased how a single npm install could silently fingerprint victims, deploy an obfuscated payload, and exfiltrate valuable data all under the guise of a trusted open-source dependency.

Attack Regions



Powered by Bing
© Australian Bureau of Statistics, GeoNames, Microsoft, Navinfo, Open Places, OpenStreetMap, Overture Maps Fundation, TomTom, Zenrin

Attack Details

#1 Between July 4, 2025, and the months that followed, uncovered ten malicious npm packages orchestrating a sophisticated, multi-stage credential theft campaign. Each of these packages, disguised with typosquatted names imitating trusted libraries like TypeScript and Discord, amassed over 9,900 downloads before a petition led to their removal from the npm registry. The threat actor, identified as andrew_r1, leveraged npm's postinstall mechanism to execute malicious scripts immediately upon installation, masking their intent beneath layers of obfuscation and a deceptive sense of legitimacy.

#2 The attack chain begins with social engineering. Once installed, the malicious package.json file triggers an install.js script that detects the victim's operating system and launches an obfuscated payload in a new terminal window. This approach allows the malware to execute independently from the npm installation process, displaying a brief flash of the terminal before clearing it to minimize suspicion and evade basic monitoring tools. Behind this stealthy execution lies a carefully engineered JavaScript payload protected by four layers of obfuscation, including a self-decoding eval wrapper, XOR-based encryption with dynamic keys, URL encoding, and control-flow confusion using switch-state logic and mixed-base arithmetic, all designed to hinder reverse engineering and automated analysis.

#3 The attack chain begins with social engineering. Victims are presented with a convincing fake CAPTCHA, giving the illusion of bot protection and legitimacy. During this phase, the malware also sends the victim's IP address to the attacker's server for fingerprinting and geolocation, allowing selective targeting. Once the CAPTCHA is completed, the next stage unfolds as the malware downloads a 24MB cross-platform information stealer—packaged via PyInstaller to run seamlessly without requiring Python. This stealer is specifically built for Windows, Linux, and macOS, enabling broad reach across developer environments.

#4 The downloaded information stealer conducts a thorough reconnaissance of the infected system, harvesting credentials and sensitive data from multiple sources. It targets system keyrings, extracting stored credentials for email clients, VPNs, and cloud storage; browsers, collecting saved passwords, cookies, and session tokens; and authentication services, stealing OAuth and JWT tokens to gain persistent access to online accounts and APIs. By combining these theft vectors, the malware ensures maximum data value while maintaining a low detection footprint.

#5 Finally, the stolen credentials and data are compressed into ZIP archives and exfiltrated to the attacker's command-and-control infrastructure. This seamless combination of social engineering, obfuscated delivery, and cross-platform credential theft makes the campaign one of the more intricate npm-based attacks observed in 2025. Its use of deceptive CAPTCHAs, professional C2 operations, and a powerful multi-OS information stealer underscores the growing risks within the open-source ecosystem, highlighting the urgent need for developers to validate package authenticity and monitor installation behaviors for any signs of compromise.

Recommendations

Immediately Remove Suspicious Packages: If you've installed any of the malicious npm packages or those with similar typosquatted names, uninstall them right away. Review your project dependencies carefully and delete anything unfamiliar or unnecessary.

Verify Package Authenticity Before Installing: Always double-check package names, publishers, and download counts on the npm registry. Malicious actors often publish lookalike packages with minor name variations to trick developers.

Inspect Post-install Scripts: Before installing a new package, open its package.json file and look for suspicious postinstall or install scripts. These scripts should raise a red flag if they launch new terminals, download binaries, or execute encoded code.

Monitor Network Activity and System Behavior: Keep an eye out for unusual network connections, new processes, or unknown executables appearing after npm installs. These may indicate hidden malware activity.

Enhance Endpoint Protection: Deploy next-generation antivirus (NGAV) and endpoint detection & response (EDR) solutions to identify and block malware. Leverage behavioral analysis and machine learning-based detection to spot suspicious activity.

Potential MITRE ATT&CK TTPs

TA0001 Initial Access	**TA0002** Execution	**TA0005** Defense Evasion	**TA0006** Credential Access
TA0007 Discovery	**TA0009** Collection	**TA0010** Exfiltration	**TA0011** Command and Control
T1195 Supply Chain Compromise	**T1195.002** Compromise Software Supply Chain	**T1027** Obfuscated Files or Information	**T1027.002** Software Packing
T1204 User Execution	**T1204.002** Malicious File	**T1059** Command and Scripting Interpreter	**T1059.007** JavaScript

T1059.004 Unix Shell	T1059.006 Python	T1555 Credentials from Password Stores	T1555.003 Credentials from Web Browsers
T1555.001 Keychain	T1539 Steal Web Session Cookie	T1552 Unsecured Credentials	T1552.001 Credentials In Files
T1552.004 Private Keys	T1071 Application Layer Protocol	T1071.001 Web Protocols	T1041 Exfiltration Over C2 Channel
T1560 Archive Collected Data	T1560.001 Archive via Utility	T1027.009 Embedded Payloads	T1140 Deobfuscate/Decode Files or Information
T1082 System Information Discovery	T1083 File and Directory Discovery	T1036 Masquerading	T1614 System Location Discovery

Indicators of Compromise (IOCs)

TYPE	VALUE
Malicious Packages	deezcord.js, dezcord.js, dizcordjs, etherdjs, ethesjs, ethetsjs, nodemonjs, react-router-dom.js, typescriptjs, zustand.js
IPv4	195[.]133[.]79[.]43
SHA256	80552ce00e5d271da870e96207541a4f82a782e7b7f4690baeca5d411ed71edb
Email	parvlhonor@gmx[.]com

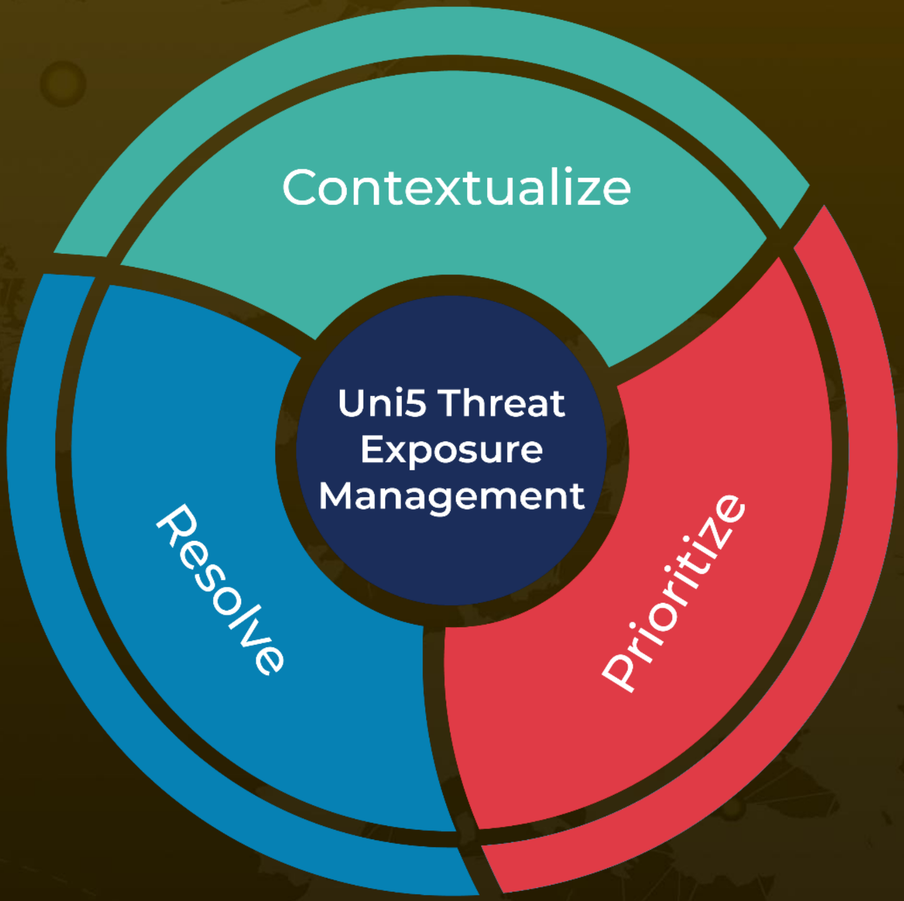
References

https://socket.dev/blog/10-npm-typosquatted-packages-deploy-credential-harvester

What Next?

At **Hive Pro**, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a free demo with **HivePro Uni5**: Threat Exposure Management Platform.



REPORT GENERATED ON

October 31, 2025 • 8:00 AM

More at www.hivepro.com