Hive Pro

Threat Level

Red

HiveForce Labs

THREAT ADVISORY

ATTACK REPORT

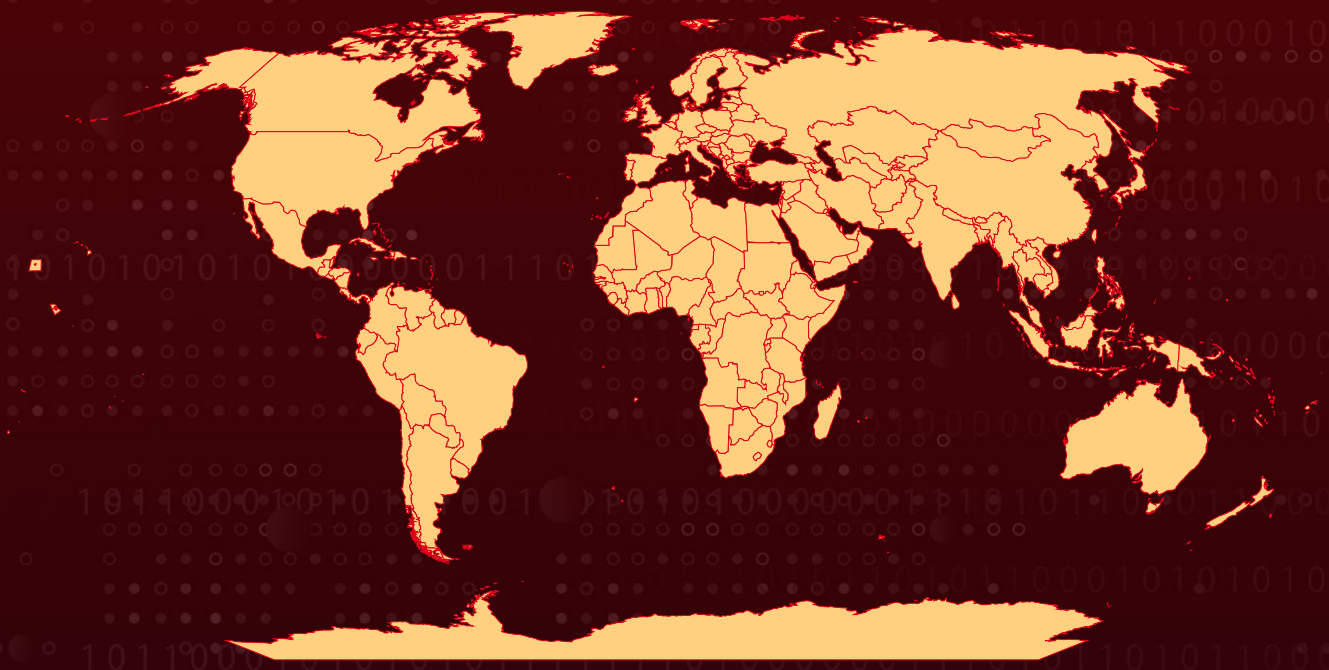
Popular npm Package ‘rand-user-agent’ Compromised in Supply Chain Attack

Date of Publication	Admiralty Code	TA Number
May 13, 2025	A1	TA2025145

Summary

First Seen: May 5, 2025
Targeted Countries: Worldwide
Attack: The widely used npm package rand-user-agent, known for generating random browser user-agent strings, was compromised in a supply chain attack. Malicious versions (1.0.110, 2.0.83, and 2.0.84) were published to the npm registry, containing obfuscated code that installed a Remote Access Trojan (RAT). This malware established a connection to a command-and-control server, enabling attackers to execute shell commands, upload files, and harvest system information. The malicious code was absent from the project's GitHub repository, indicating a targeted attack on the npm distribution channel. Although the compromised versions have been removed, affected users are advised to conduct thorough system scans, as simply downgrading the package does not eliminate the RAT.

Attack Regions



Attack Details

#1

The supply chain attack on the rand-user-agent npm package marks a sophisticated example of how trusted open-source libraries can be weaponized to distribute malware. On May 5, 2025, security researchers discovered that a Remote Access Trojan (RAT) embedded in recent versions of the package, specifically versions 1.0.110, 2.0.83, and 2.0.84. This package, which had over 45,000 weekly downloads, is commonly used to generate random browser user-agent strings, seemingly harmless functionality that made it a perfect cover for malicious activity.

#2

The attacker did not compromise the project's GitHub repository but instead pushed the malicious versions directly to the npm registry. The malicious code was hidden in the dist/index.js file, disguised with excessive whitespace to avoid detection in code viewers and editors. This obfuscation concealed a payload that was designed to be reconstructed and executed at runtime. By manipulating string order and using disguised functions, the attacker made the injected code difficult to analyze without deep inspection.

#3

Once the malicious code was executed, it established a connection to a command-and-control (C2) server. The malware harvested basic system information, including the hostname, username, and operating system, and sent it to the C2 server. From there, the attacker could issue commands to the victim's system, including executing shell commands, changing directories, and uploading files. This level of access allowed for full remote control of the compromised machine.

#4

To maintain stealth, the malware created a hidden directory (~/.node_modules) and extended the module.paths so that it could load required packages like axios and socket.io-client without raising immediate red flags. This use of standard libraries and environment manipulation helped the attack blend into normal development workflows, making detection more difficult.

#5

The malicious versions were quickly removed from npm once the breach was identified, but users who installed the affected releases remain at risk, as simply downgrading does not remove the RAT from compromised machines. The incident underscores the ongoing risks associated with open-source supply chains, where compromised packages can infiltrate thousands of systems through automated dependency management.

Recommendations

Audit and Remove Malicious Versions: Ensure that versions 1.0.110, 2.0.83, and 2.0.84 of rand-user-agent are not present in your projects. Even after removal, perform comprehensive system scans to detect any residual malicious code or backdoors introduced by the compromised package.

Implement Dependency Management Best Practices: Lock dependencies to specific versions using tools like package-lock.json to prevent automatic updates that might introduce vulnerabilities. Adopt scoped packages (e.g., @your-org/package-name) to minimize the risk of name collisions and dependency confusion attacks. Set up .npmrc files to define trusted registries and prevent unintentional installations from unverified sources.

Enhance Security During Package Installation: Use the --ignore-scripts flag during installation to prevent the execution of potentially malicious pre/post-install scripts. Utilize tools and practices that verify the integrity and authenticity of packages before installation.

Strengthen Authentication and Access Controls: Activate 2FA for accounts associated with package publishing to add an extra layer of security. Apply the principle of least privilege by granting only necessary permissions to users and services interacting with your codebase.

Potential MITRE ATT&CK TTPs

TA0002 Execution	**TA0003** Persistence	**TA0011** Command and Control	**TA0001** Initial Access
TA0005 Defense Evasion	**T1071** Application Layer Protocol	**T1082** System Information Discovery	**T1059** Command and Scripting Interpreter
T1195 Supply Chain Compromise	**T1027** Obfuscated Files or Information	**T1036** Masquerading	**T1071.002** File Transfer Protocols
T1195.001 Compromise Software Dependencies and Development Tools	**T1140** Deobfuscate/Decode Files or Information	**T1071.001** Web Protocols	

Indicators of Compromise (IOCs)

TYPE	VALUE
IPv4	85[.]239[.]62[.]36
URLs	hxxp[://]85[.]239[.]62[.]36, hxxp[://]85[.]239[.]62[.]36:27017/u/f, hxxp[://]85[.]239[.]62[.]36:3306, hxxps[://]www[.]webscrapingapi[.]com/

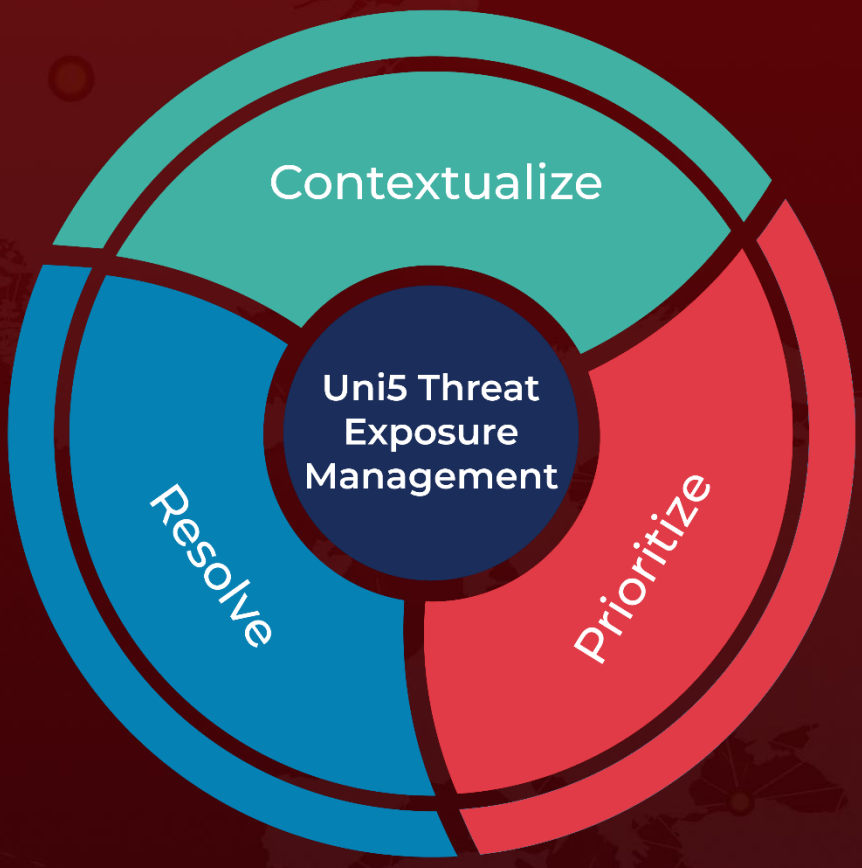
References

https://www.aikido.dev/blog/catching-a-rat-remote-access-trojian-rand-user-agent-supply-chain-compromise

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

May 13, 2025 • 3:30 AM

More at www.hivepro.com