PROGRAMMING

GitHub Copilot Generated Insecure Code In 40% Of Circumstances During Experiment

SEPTEMBER 02, 2021 0

Researchers published a scholarly paper examining the security implications of GitHub Copilot, an excellent AI system now being used for code completion in Visual Studio Code. **In various scenarios, some 40 percent of tested projects were found to include security vulnerabilities.**

GitHub Copilot is defined as an "AI pair programmer" whose advanced AI system from OpenAI, called Codex, is trained on high-quality code repos on GitHub. So it works like a super-charged IntelliCode. Codex is an improvement on OpenAI's Generative Pre-trained Transformer 3 (GPT-3) machine language model that utilizes deep learning to generate human-like text.

"OpenAI Codex has a broad knowledge of how people use code and is significantly more capable than GPT-3 in code generation, in part, because it was trained on a data set that includes a much larger concentration of public source code," GitHub CEO Nat Friedman said in a June 29 **blog post.**"

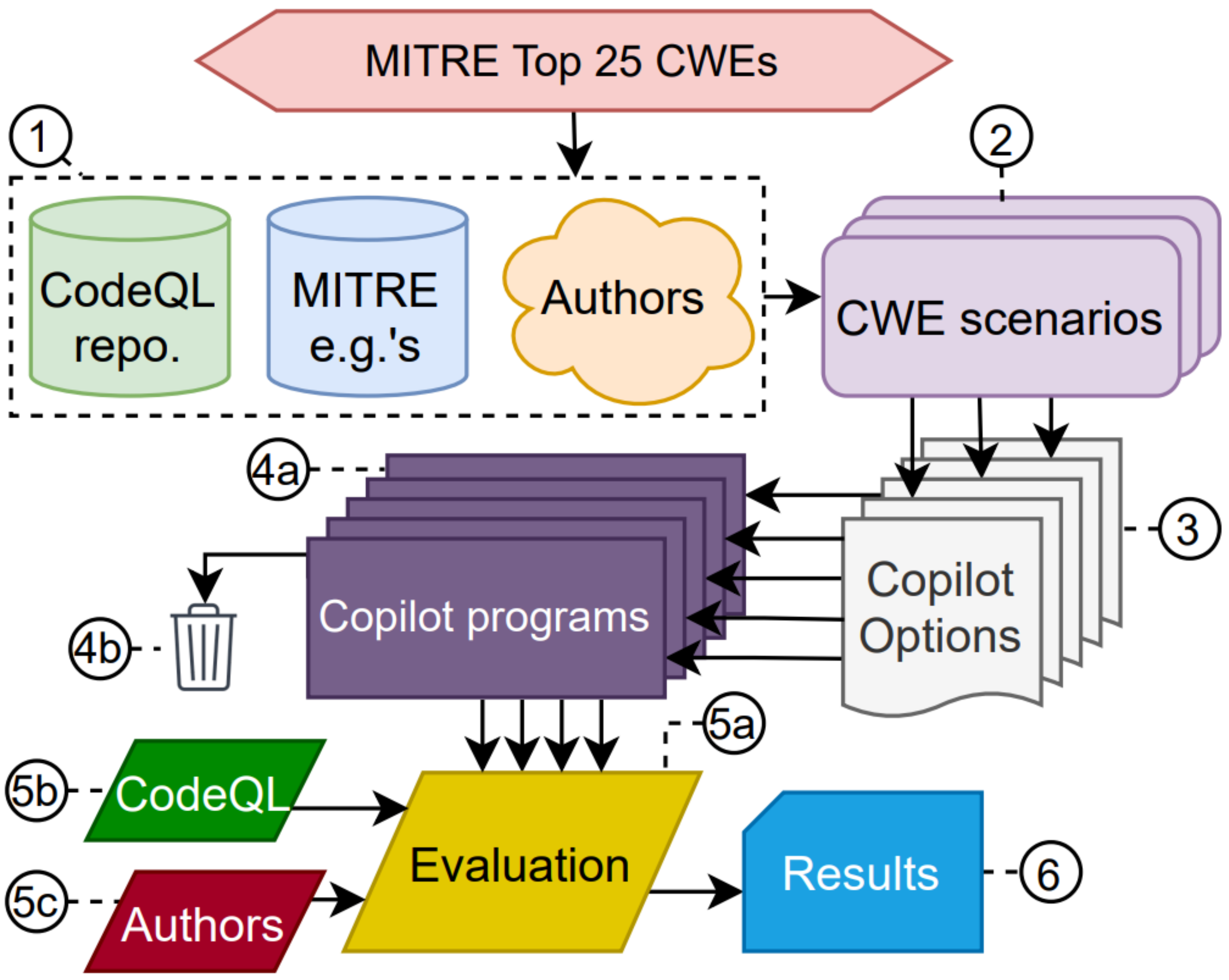
GitHub Copilot works with a broad set of frameworks and languages, but this technical preview runs very well for Python, JavaScript, TypeScript, Ruby and Go."

The project instantly stirred up controversy along various aspects,
with hints surrounding the quality of code, legal and ethical concerns, the possibility of replacing human developers, and the potential to advance security vulnerabilities.

It's that last item, security, that is the locus of the new scholarly paper, titled "An Empirical Cybersecurity Evaluation of GitHub Copilot's Code Contributions." The study aimed to identify the tendency of Copilot to generate insecure code, providing a gauge for the amount of scrutiny needed on the part of users to guard against security issues.

Using rigorous and detailed scientific analysis, the study concluded that upon testing 1,692 programs generated in 89 different code-completion scenarios, 40 percent were found to be vulnerable.

The situations were related to a subset of the top 25 high-risk Common Weakness Enumeration (CWE), a community-developed list of software and hardware weakness types managed by the not-for-profit MITRE security organization.



The study traced Copilot's behavior along three dimensions:

- **Diversity of domain,** its response to the domain, i.e., programming language/paradigm
- **Diversity of weakness,** its propensity for generating code that is susceptible to each of weaknesses in the CWE top 25, given a scenario where such a vulnerability is possible

- **Diversity of prompt,** its response to the context for a particular scenario (SQL injection)

"Overall, Copilot's response to our scenarios is mixed from a security standpoint, given the large number of generated vulnerabilities (across all axes and languages, 39.33 percent of the top and 40.48 percent of the total options were vulnerable)," the paper said.

"The security of the top options is particularly important -- novice users may have more confidence to accept the 'best' suggestion. As Copilot is trained over open-source code available on GitHub, we theorize that the variable security quality stems from the nature of the community-provided code. That is, where certain bugs are more visible in open-source repositories, those bugs will be more often reproduced by Copilot."

The scholarly paper joins another one titled **"Evaluating Large Language Models Trained on Code"** that investigated security along with legal and other implications.

"Codex has the potential to be useful in a range of ways," says that paper, published last month. "For example, it could help onboard users to new codebases, reduce context switching for experienced coders, enable non-programmers to write specifications and have Codex draft implementations, and aid in education and exploration. However, Codex also raises significant safety challenges, does not always produce code that is aligned with user intent, and has the potential to be misused."

GitHub Copilot was also criticized by the Free Software Foundation, which proclaimed that it was **"unacceptable and unjust"** in calling for yet more papers to be published to address philosophical and legal issues around the project.

It also stirred up existential anxiety among some developers who are worried that it and other advanced AI systems could replace human coders.

"There is no question that next-generation 'auto-complete tools like GitHub Copilot will increase the productivity of software developers," the authors (Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri) say in conclusion.

"However, while Copilot can rapidly generate prodigious amounts of code, our conclusions reveal that developers should remain vigilant ('awake') when using Copilot as a co-pilot. Ideally, Copilot should be paired with appropriate security-aware tooling during both training and generation to minimize the risk of introducing security vulnerabilities. While our study provides new insights into its behavior in response to security-relevant scenarios, future work should investigate other aspects, including adversarial approaches for security-enhanced training."

There was no mention of enhanced functionality to safeguard against the influx of security vulnerabilities, so possibly more papers and studies are in the works.

Next Read:

- **70+ Jetpack Compose Tutorials for Beginners, Intermediate and Advanced Android Developers**
- **70+ Flutter Projects For Beginners, Intermediate And Experienced Developers**
- **70+ Python Projects For Beginners, Intermediate & Advanced Developers With Source Code**

Programming

SHARE

RELATED POSTS