How generative AI improves software security

The technology that takes the guesswork out of fixing unsecure code before applications are deployed

Robert Larkin

Veracode

GENERATIVE AI TOOLS are becoming increasingly prevalent, providing interactive experiences that captivate the public's imagination. These tools are accessible to anyone, offering a unique opportunity to engage and explore the creative possibilities enabled by AI technology. The technology doesn't just train a model to recognize patterns. It can create things that are easy to understand: images, text, even videos. Sometimes the results are hilariously wrong, but other times the results are quite impressive, such as clear, concise answers to complex questions. Generative pre-trained transformer (GPT) technology, such as ChatGPT, has opened the doors for everyone to be an evaluator because the output is accessible and easy to critique.

How generative AI relates to the security space is a little less of an open forum, however. We have specific problems that need targeted solutions, so the ways we try to leverage the technology are more focused than free-form. Some organizations are using models to gain a real-time understanding of threats in a network. Veracode chooses to focus on the application code that powers systems and protects data, making us a code-driven rather than event-driven security provider. Specifically, we use generative AI to help developers write secure code — something they often don't know how to do.

MAPPING BAD CODE TO GOOD PATCHES

Forrester published an excellent report on this topic titled "Show, Don't Tell, Your Developers How to Write Secure Code." Forrester researchers found that 70% of applications fail their first static security scan because of violations of the top 10 recognized vulnerabilities and flaws in development and design. The report notes that software developers are not taught cybersecurity concepts in college, while people who have even less training are now writing code via low-code platforms.

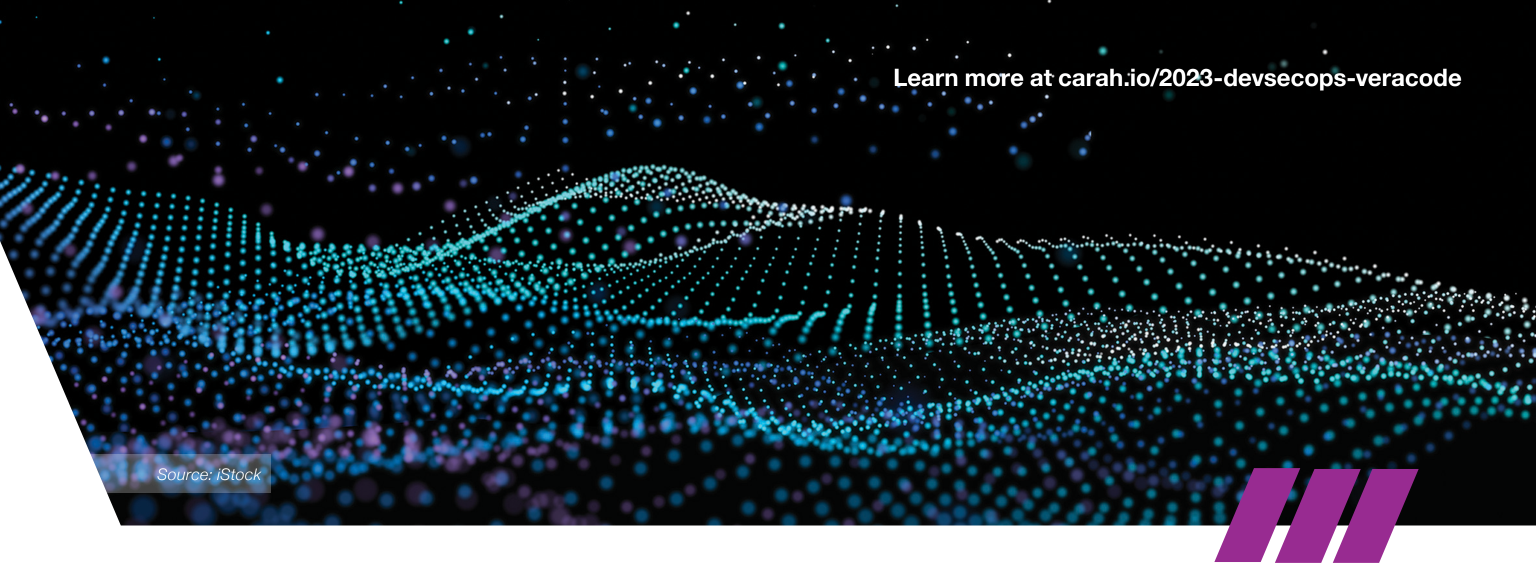
> "Veracode chooses to focus on the application code that powers systems and protects data, **making us a code-driven rather than event-driven security provider.**"

Veracode is helping developers learn some key security techniques with a new tool called Veracode Fix. It takes the next step from scanning code for flaws and vulnerabilities to showing developers how to fix the code before deploying it in an application. The tool is built on GPT technology, but it has not been trained on customer code gathered during our 17 years in business. First of all, we don't store customer code. Second, we already know that 70% of the world's code is unsecure, and training a model on bad input leads to bad results.

Instead, we start with a scan that identifies a flaw and then train the model to recognize what kind of flaw it is. Mapping the code to known bad examples is the GPT part, and those examples are curated by our research team. Each set of known bad flaws comes with known good reference patches. Rather than simply spitting out code, we are conducting a targeted exercise to match bad code with good patches, and that's how we produce actionable, secure results.

A SUPERVISED LEARNING MODEL ON A CURATED DATASET

No customer code references are used in the process, and no customer code from the process is used for future



Source: iStock

fixes. That's different from other services that mine data from the questions being put to it or other types of machine learning that have produced distressing results because of bias in the training data.

We avoid those problems by using a supervised learning model on a curated dataset. The approach ensures that we can provide our government customers with reliable fixes they can easily implement. Those efforts put Veracode in the vanguard of cloud-based software development tools that meet the government's security and modernization requirements. ■

Robert Larkin is senior solutions architect at Veracode.