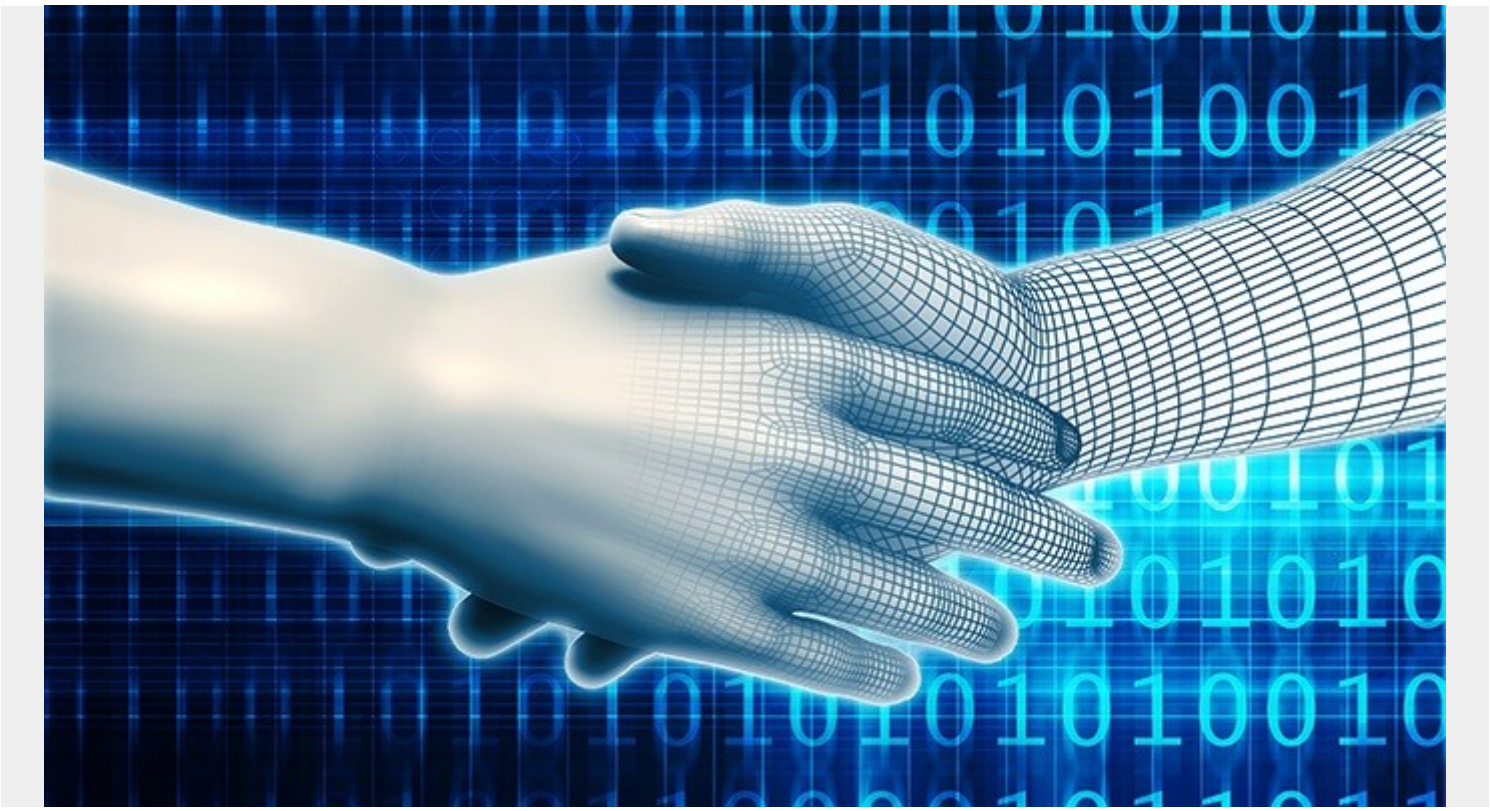
SHIFT LEFT CODE SCANNING WITH BMC DEVXAND VERACODE



You know those action movies where the bad guy and the good guy are fighting it out on the roof of a moving train? That's essentially what it's like to be a software developer. They are tasked with fixing immediate problems while dodging ongoing hurdles. There is constant pressure to develop code quickly and to get it right the first time. Organizations like banks, which have real-time operations and critical customer requirements, cannot afford to have exploitable software. There is no room for error and no time left for edits.

That's why we have paired BMC AMI DevX Workbench for Eclipse with Veracode – to discover security risks in mainframe applications early in the development lifecycle. Although the current integrated development environment (IDE) of DevX Workbench for Eclipse already edits and debugs code, the Veracode integration allows developers to shift left and scan code for security defects earlier in the development lifecycle (SDLC), where not only is it easier and less costly to fix, it's vital.

The integration between two technologies delivers both minimum viable compliance and continuous compliance since code can be checked in the pipeline to ensure compliance across the board. This in turn allows greater security for individual developers while also satisfying the office of the CISO whose mantra is "all code must be invulnerable" or "all vulnerabilities must be exposed and fixed." Such actions are essential in order to avoid the misguided belief in "security by obscurity" as well as the very real dangers of zero-day flaws.

An expert pair of eyes

Software engineers who write code are not always going to be able to know if that code is vulnerable, especially if they are new to the job, or if they are using code snippets pulled from templates. There are many ways to create something that is vulnerable, not only on the mainframe, but something that is prone to cross-site scripting later on. It can happen inadvertently—a vulnerability that's not syntactically incorrect and in which the logic of the program does what is expected, but which is still a vulnerability.

Veracode is like an expert pair of eyes that can be called on automatically, like developer guard rails. It's an automated expert that is brought into an integrated development environment to help developers avoid writing vulnerable code, without having to rely on a later pull request or peer review that itself might not detect the anomaly.

It is very possible for a large organization to produce vulnerable code without being aware of it. For example, a developer might make a database lookup call to pull up information based on user input without sanitizing the input for SQL database commands. That could result in exfiltration or destruction of data.

It is also possible for a spam email to deliver a payload that gets inside the bank's code and gets distributed, perhaps via SWIFT or via a mobile phone app. The vulnerability can go far beyond the confines of the mainframe. That's where Veracode truly shines since it isn't just about mainframe. Code that goes on to a mobile phone app, for example, can slip through an app marketplace's own controls.

These types of situations lead to more than just a vulnerability being exploited. There is also significant reputational damage, especially for a bank or financial institution. Customers will suffer, and organizations that supplied the code will develop a reputation for faulty product.

The goal with Veracode is to ensure no vulnerable code gets past the earliest stages of the SDLC. It's scanned all the way through, including the repository. This is shift left in action. Veracode and the mainframe boosts the hygiene factor, the engineering rigor that is needed by non-mainframe developers and mainframe developers alike, especially as both groups race to keep pace with the demands of the distributed marketplace.

Leveraging the open-borders approach

The integration between Veracode and BMC AMI DevX Workbench for Eclipse is part of BMC's open borders approach, which allows organizations to leverage their existing footprint, especially those that already have Veracode in their organization. BMC's process has always been to use the industry-leading solutions in an organization's DevOps toolchain while also ensuring that the toolchain delivers a best-in-class, best-in-breed, or "open borders" approach.

The bottom line is that the BMC DevX-Veracode integration gives developers the ability to shift left in their security testing. They can produce code in our Topaz IDE and then immediately test it for security vulnerabilities or compliance issues in the code they have written and fix it in the SDLC—a more timely and necessary solution. Additionally, entire code bases can also be scanned, using an automation pipeline, so that any code vulnerabilities cannot get introduced when merges occur.

To learn more about the BMC open borders approach and our integrations with best-in-class partners, check out the BMC AMI DevX Workbench for Eclipse on our BMC mainframe integrations

webpage and look for the Veracode tile. To see how the integration works, watch this short demo video.