Anthony Quintero-Quiroga

Mark Bates

Intro to Go

October 3, 2021

My History in Technology

I first got into technology when I was twelve years old. I learned how to build my own computer for the purpose of building a gaming PC. In high school I took a tech support class for three years. During this time, I had an internship with a local PC store, which involved building computers for school districts and providing onsite IT support. I always had an interest in building applications, so I pursued a bachelor’s in computer science to learn more about computers and how to build applications. Upon graduating college, I started my career working for NIWC Pacific. Although I was hired for software development, I majority of my tasking has been cyber security related.

One recurring task that I get is to conduct audits on ships, which involves going onsite and verifying that all applicable Security Technical Implementation Guides (STIGs) have been applied to the system. Normally this is a long and tedious process where we would have to manually check hundreds of different STIGs. However, my team has developed a tool help automate most of this process. Audits that once took a week, or longer, can now be completed within a few days. This tool, however, has some flaws and runs on old legacy code. We plan on modernizing and improving the tool. Go fits into our future because we have chosen to modernize our tool using the Go language. We chose Go because it is a compiled language that supports building for multiple platforms. Since the different ships all have different environments and are different from our development environment, it is essential that we can build our application for a specific environment beforehand. By doing so, we can guarantee that our tool will work on the system. Our tool was originally built for Linux Operating Systems. Using Go’s cross compiling capabilities, we will now be able to use the tool on Windows computer and any other possible Operating System. Go’s built-in concurrency should improve the functionality of our tool as it will allow us to run concurrent system checks simultaneously. Finally, Go’s built-in server capabilities will also allow us to add additional features to our tool that will make the tool easier to use with better reporting capabilities. The fact that Go offers so many features without dependencies is crucial since making changes to any ships baseline is disallowed for security purposes.

In addition to modernizing the auditing tool, I may also use Go to modernize a web application that I currently support. The website is essentially a Dropbox for the entire Navy. Dropbox utilizes Go for some of its capabilities. Therefore, I figure our website could also benefit from using Go. Again, I believe that goroutines could prove to be useful in speeding up some of our website’s features. Since Go was designed with scalability in mind, modernizing to Go should ensure that our website continues to scale up without jeopardizing speed. Finally, Go’s simplicity should make the transition from our previous programming language to Go much easier for our developers. Overall, Go seems promising for my future endeavors, and I look forward to learning more about it.