Rugged Software Development

Are We Doing Enough?

Many of us can admit that security is often either an oversight or an afterthought when we're developing our applications. Raise your hand if you agree. Due to a number of reasons, mainly time and budget, the focus is mostly on coding new features and then getting them deployed as soon as possible. Our pipeline technologies contain a number of tools that help us to identify the vulnerabilities in our code (SAST), in the open source components that we use, (SCA) and in a running app (DAST). We rely heavily on these tools to catch as many potential vulnerable areas in our code as possible ***but it's not enough!***

We need to take additional steps to ensure that our applications are going to run as intended, regardless of the conditions it encounters when it is deployed. A big part of this is obviously security but that isn't the only thing. We also have to take into account reliability, survivability and a variety of other factors. The Internet, and networks in general, are dangerous places where we need to be conscious of both malicious actors as well as other unintentional adverse conditions. This is where the concept of "*Rugged Software Development*" comes into play.

Rugged Software Development?

In a nutshell, "*Rugged Software Development*" describes software development teams that have a culture of rapidly evolving their ability to create available, survivable, defensible, secure, and resilient software. This is exactly the kind of culture that we need to strive to be here as we continue to develop applications for our internal and external customers.

It’s a Mindset

There’s been a growing movement among the developer community and it has culminated with the creation of the *Rugged Manifesto*. The *Rugged Manifesto* is a set of statements or better yet, a "mindset" that each of us should consider, should be thinking about, should strive to put into practice every day as we are writing our code.

Basically, the Rugged Manifesto states:

- I am rugged and, more importantly, my code is rugged.
- I recognize that software has become a foundation of our modern world.
- I recognize the awesome responsibility that comes with this foundational role.
- I recognize that my code will be used in ways I cannot anticipate, in ways it was not designed, and for longer than it was ever intended.
- I recognize that my code will be attacked by talented and persistent adversaries who threaten our physical, economic, and national security.
- I recognize these things - and I choose to be rugged.
- I am rugged because I refuse to be a source of vulnerability or weakness.
- I am rugged because I assure my code will support its mission.

- I am rugged because my code can face these challenges and persist in spite of them.
- I am rugged, not because it is easy, but because it is necessary and I am up for the challenge.

These are simple statements, a mindset as described earlier, that we should all be saying to ourselves, thinking about, as we are developing our applications.

In closing

As software developers we have an incredible opportunity to move our world into the future and as such we have an ethical and moral responsibility to ensure that our code is clean, bug free and void of anything that can be exploited from both internal and external malicious sources. It is encouraged that all of us take the above statements and put them into practice – to help encourage and foster a culture that promotes rugged development. You can read more into this by visiting: https://ruggedsoftware.org/