**Key Principles for writing secure code**

Cybersecurity has become the major thread of any application. Hackers. phishers, malicious attackers are everywhere and their attacks can result in collapse of the largest organizations. According to data from BBN Times, organizations will spend an estimated $101.6 billion on cybersecurity software, services and hardware by 2020.

Being an application developer, part of duties is to write code securely and develop application layers with vulnerabilities top of mind. There are a lot of different way to write secure code, but based on OWASP, Open Web Application Security Project, there are 10 principles for securing coding. In order to summarize them, I put them in different catalogs:

Minimize complexity of application layers

* Principle #1: Minimize attack surface area
  + Complexity of architecture in source code and related functionality creates hole for developers to ignore security measure.
  + The more complex the system is, the more opportunity security hole exist.
* Principle #2: Don’t trust services
  + Validate input from any data sources, especially external one
  + Be suspicious of command line arguments, SOA, environment variables, properties files, etc.
  + Also, sanitize data that sending to other subsystem to make sure any distrustful data that pass the validation will not pass to other system like database
* Principle #3: Keep security simple
  + Following KISS principle (Keep It Simple, Stupid) approach
  + Keep the design as simple and small as possible
* Principle #4: Fix security issues correctly
  + Once the bug is found, find out the root cause, and make sure to screen the application thoroughly to look for the similar issue in it

Maximum depth for defense

* Principle #5: Principle of Defense in depth
  + Using multiple defensive strategies for risk management, therefore the application will not come down by only 1 layer of defense
* Principle #6: Separation of duties
  + The key to valid fraud is the separation duties amongst multiple user roles; while administration users are able to maintain the application but not able to use it; normal users are able to use the application but not able to maintain it
* Principle #7: Fail securely
  + When application fail for any reason, by using try-catch code, the application should halt under it’s own flow, insert error to error log, rollback all partial updates
  + If the application is able to continue with the failure, make sure the data is secure before continue

Enhanced rules for permission, whitelisting, and Obscurity

* Principle #8: Establish secure defaults
  + Don’t put user experience before security, even that will improve user experience
  + Unless user perform to take the risk, don’t lower the default security level
* Principle #9: Principle of Least privilege
  + Any user only needs the least privilege to perform the process, don’t provide extra premission
* Principle #10: Avoid security by obscurity
  + Hiding the information and processes of the application is not effective for security.
  + Adopting a secure coding standard, apply and develop application base on it is far better