* Define security and privacy requirements **(Makes it easier to identify targets and minimize disruptions in plans and schedules) SDL PRACTICE #2**
* Create quality gates/Bug Bars (Table of possible threats to our app, categorised from critical, important, moderate, low) **SDL PRACTICE #3**
* Security and Privacy Risk assessment (?) **SDL PRACTICE #4**
* Define design requirements **SDL PRACTICE #5**
* Attack surface analysis (Reducing the amount of code running, reduce entry points, eliminate services used by very few users. **SDL PRACTICE #6**
* Threat modelling – potential threats are to be identified, enumerated, prioritised all from an attackers point of view. Provides a systematic analysis to the probable attacker’s profile, the most likely attack vectors, and assets most desired. **SDL PRACTICE #7**
* Using approved tools (?) **SDL PRACTICE #8**
* Deprecating unsafe functions – Analyse all project functions and APIs and removing those which are unsafe to help reduce security bugs. Use code scanning tools which check code for functions on the banned list, replacing them with safer alternatives. **SDL PRACTICE #9**
* Static analysis – checking the code without actually running it. (Binary Static Analysis Flowchart) **SDL PRACTICE #10**
* Fuzz Testing. **SDL PRACTICE #12**
* Conduct final security review **SDL PRACTICE #15**