Phases of the Microsoft SDL

Megan Leonard

ISEC 475

7/2/2023

Microsoft SDL is another framework that we can see is split into different phases and practices. SDL is split into 7 different phases that it follows with 5 inner phases having 3 practices each while the first and last phase each have one practice. The first phase is training which has the practice core security training where employees are taught about security awareness and anything that is needed for them to complete their roles.

The next phase is requirements which is the time to determine the necessities for the project security and determining risks. The first practice is establishing security requirements, which is finding what security methods are needed to comply with company standards. The second practice is creating quality gates and bug bars. This is to create a base acceptable security level and a line between what needs fixed and what bugs are fine. The third practice is security and privacy risk assessment which goes over the possible risks and how the company will handle them.

Then we have the design phase which is the start of determining what the design should look like and how it should work. The first practice is establishing design requirements, which is choosing the design aspects based on the company standards and the security needs. Then we analyze surface attacks which looks into how outside attacks affect the system and vulnerabilities within the design. Finally, we have threat modeling where we look at the different attacks that could happen to the software and see how they impact and the damage they cause.

Implementation is the next phase which starts adding in the chosen designs and requirements. The first practice is using approved tools which means that when implementing the design, it has to be done with tools that fall under the requirements that were given and follow the standards of the company. Then there is deprecating unsafe functions where any function that is not up to the safety standards is deactivated. Then we have static analysis which looks over the structure that was just implemented to make sure it is following the given requirements and is safe.

Verification follows where we start running more tests against the system to see how it holds. First, we have dynamic analysis which looks at the system while it is running to make sure that the checks from the static analysis hold up when it is running as well. Fuzz testing follows where injections with unexpected or invalid inputs are entered to see how they are handled by the system or if there are vulnerabilities in place. Then we have attack surface review where we look over the previous analysis of the attack surface and see how it works with our new design and if problems have been handled or need to be looked at.

Release follows with the final checks to the structure before it is released. Incident response plan is a plan created to look at different possible risks that could be encountered and how the company will deal with them. Then we have the final security review to make sure that all problems have been addressed and it runs as expected. Finally, we have released the archive which is launching the system and releasing it.

The final phase is response which is executing the incident response plan as needed. The seven phases each play a role within the development and are all needed in their own way. The practices help split up the phase topic to make sure the different aspects are worked on.

Citation:

AI, S. C. (2019, April 4). *How to approach security development lifecycle (SDL)*. Medium. <https://medium.com/@SAPCAI/how-to-approach-security-development-lifecycle-sdl-7a3002a534f3>

BruceCowper. (n.d.). *SDL security bug bar (sample) - security documentation*. Security Documentation | Microsoft Learn. <https://learn.microsoft.com/en-us/security/sdl/security-bug-bar-sample>

Gluckd. (n.d.). *Microsoft Security Development Lifecycle (SDL) - microsoft service assurance*. Microsoft Service Assurance | Microsoft Learn. <https://learn.microsoft.com/en-us/compliance/assurance/assurance-microsoft-security-development-lifecycle>

Mead, N. R., & Woody, C. C. (2017). *Cyber Security Engineering: A practical approach for systems and software assurance*. Addison-Wesley.

SDL in practice - owasp foundation. (n.d.). <https://owasp.org/www-pdf-archive/SDL_in_practice.pdf>

*What is Fuzz Testing and how does it work?*. Synopsys. (n.d.). <https://www.synopsys.com/glossary/what-is-fuzz-testing.html#:~:text=Definition,as%20crashes%20or%20information%20leakage>.