Insecure Deserialization

Insecure deserialization is still a threat in cyberspace and it is often related to insufficient access control. Explaining roughly serialization and deserialization, the former process converts an object into byte streams and the latter does it in the other way around (F5DevCentral, 2018) (Mitre, 2017). The relationship to weak access control can be understood in thinking about the moment when deserialization happens. This can be at the moment when data leaves a database but also when data will be converted back by the CPU while moving out of its instruction cycle (= an output given by the machine). So, if there is only little access control (no differentiation between a normal “user” and an “admin” for example) when data is requested (a “load” statement) then the system is much more sensitive to attacks (Cassou et al., 2016) (Pillai, 2017) (Shaji, 2020).

Beside increasing access control (“authorization”) it should also be mentioned that improving authentication can also reduce the risk of insecure deserialization. In other words, displaying (loading) information to the user will only lead to a result when authentication is valid (a finger print for example) (Pillai, 2017) (F5DevCentral, 2018) (Shaji, 2020). As a final note, it should fairly be said that not only deserialization difficulties can occur while handling data but also serialization is far from being secure when not effectively protected. A common threat here is that a malicious user sends input data that can not be converted by the machine, ending in an almost never-ending process which costs a load of CPU capacity and can overload the system. In this case, it is important that user inputs will always be validated before they become serialized (for example, making a type check and when the type is not valid then return an error message) (Shaji, 2020).

Reference List:

Cassou D., Ducasse S., Fabresse L., Fabry J. & Van Caekenberghe S. (2016) *Enterprise Pharo a Web Perspective.* N.P.. Lulu Press

F5DevCentral (2018) 2017 OWASP Top 10: Insecure Deserialization. Available from: https://www.youtube.com/watch?v=nkTBwbnfesQ [Accessed 29 June 2022]

Mitre (2017) Weaknesses in OWASP Top Ten.

Pillai, A.B. (2017) *Software Architecture with Python*. Birmingham, UK. Packt Publishing Ltd

Shaji, S.B. (2020) Using Python's pickling to explain Insecure Deserialization. Available from: https://medium.com/@shibinbshaji007/using-pythons-pickling-to-explain-insecure-deserialization-5837d2328466 [Accessed 29 June 2022]