Wednesday Room 3 15:00 - 16:00 Talk (60 min)

Sandboxing .NET assemblies for fun, profit and of course security!

In our current way of developing .NET applications we rely a lot on third-party libraries developed by others. This of course has a lot of benefits from productivity perspective because there is no need to write needed functionality from scratch.

But by using in a third-party library you also pull in it's issues and possibly security problems that are found over time. What does the library do? And what type of other libraries and/or functionality does it rely on? What do the projects/people behind it do for security?
If we develop a .NET application using external libraries can we improve our security posture? Other new technologies like WebAssembly introduced a concept of nano-process, which allows the developer to limit the capabilities available for an external module by creating a restricted sandbox for it. Could we maybe do the same in .NET? In the old days we could use AppDomains and Code-Access Security (CAS) to achieve that, but with the introduction .NET Core there only is a single AppDomain and CAS has been deprecated. Luckily with .NET Core we did get more internals exposed on AssemblyLoadContext and in this session we're going to create a sandbox using that. A restricted sandbox that limits the functionality available that will improve the security posture of our application!

Niels Tanis

Niels Tanis has got a background in .NET development, pentesting and security consultancy. He also holds the CSSLP certification and has been involved in breaking, defending and building secure