Enterprise software security risk mitigation

7 Elements was engaged by an enterprise software company which provides accounting and ERP solutions to large financial organisations. The purpose of the engagement was to perform an application security assessment of a new cloud-based version of its widely deployed Windows application.

The client had developed a custom Object Relationship Mapper (ORM) in order to transport complex business objects via SOAP HTTP messages to its cloud-based ASP.NET web application.

Throughout the engagement, 7 Elements worked closely with the client in order to ensure the scope of the assessment would meet the goal of risk identification and mitigation. As the client was moving from a traditional client/server, local network deployment model to an internet based web application there were a number of threat vectors we brought to their attention. These challenged some of their assumptions regarding their application and security model, and enabled the development team to take this in to account.

Through a series of meetings with the Project Manager and IT Development representatives we quickly established a pragmatic approach to the assessment, delivering a risk focused testing schedule. Our client took the security of their product very seriously and the original scope was to test all data input fields on all forms in the application, totalling many hundreds. Through discussion with the client as to the nature of the inputs, we suggested an alternate risk-based approach. We ascertained with the development team that there was significant commonality in the backend code and we therefore recommended that specific inputs be identified which would give total coverage of the core code. This was well received and allowed us to reduce the total effort on that area of the assessment by approximately 60% while maintaining the same overall assurance for the client.

The reduction in expected effort on the data input side afforded the client more time for 7 Elements to explore the custom ORM layer. Traditional web application testing tools could not read or manipulate the data in transit due to the use of an unsupported compression method, Fastinfoset. We developed a bespoke plugin combining Java and C# code in order to decode and re-encode SOAP messages sent between the thick client and the cloud-based web application; this allowed us to send attack payloads without the constraints of the Windows application itself. This directly led to the discovery of a number of security issues for which our client was able to quickly develop fixes.

Close collaboration with the client, and the development of custom code to enable an attack on the application, allowed us to deliver maximum value to the client. From the initial pre-assessment scoping phase, through the engagement itself and then in subsequent wash-up calls with all stakeholders, we combined efficient use of automation to test for common weaknesses with robust manual techniques focused on the client's particular application and threats. Efficient internal resource allocation also allowed us to deliver the full assessment and report against a very tight timeline and drop-dead delivery date of the new software to pilot customers.