

Title

**Changing Security:
Breaking the Dependency on Kerberos using DSL**

Speaker(s)

Jason Partington / SourceCode

To whom is the presentation addressed?

Test Managers, Risk Managers, Release Managers, Testing Team Leads

Keywords

DSL, Test Cases, Integration, Kerberos, Distributed landscapes

Abstract

Security changes are among the most impactful and risky that we face from a QA perspective. Most changes impact their area and often have other impact areas, but security is a totally different kettle of fish. I came here last year and listened and enjoyed the presentations, of particular interest to me were a couple that dealt with 'what is being tested'. This question comes to me regularly from all directions and soon after the conference, November 2010 to be precise; I realized that we weren't doing a good job of answering that question: this meant that I wasn't doing a good job of showing the coverage and exactly what QA was doing. The bugs were visible, but the method was opaque.

Later that month it was announced that we were going to remove our dependency on Kerberos. Kerberos is an excellent and secure authentication mechanism that ensures that only trusted accounts are capable of transferring calls between machine boundaries. Kerberos is also – in Greek mythology – a three headed dog that guards the gates to Hades. This should give you a clue that number 1, it is difficult to configure and number 2 it will be difficult to remove a dependency on it. The problem from a QA side was multi-faceted: 1. we didn't have a set of authentication test cases, we'd always needed Kerberos. 2. The change was the most far reaching that we'd ever done, meaning that we'd need to employ multiple methods to test this. 3. We weren't 100% sure of the full scope an authentication change would cause, from a product perspective we integrate to all sorts of systems including, but not limited to: Windows, SAP, MS SQL, SharePoint, InfoPath, CRM, Excel, Word, Active Directory, etc. On top of that we do not dictate landscapes to clients, this means that different clients run on very different landscapes and are often distributed – creating the initial problem. Upgrades would need to work seamlessly for existing clients with Kerberos and new installs would need to work seamlessly without Kerberos.

The solution was multi-faceted too, including testing over existing distributed environments (all virtualized) and newly created distributed environments that had never heard of Kerberos. Development would be peer-reviewed at every step and unit tested thoroughly. On top of that we needed test cases. Many test cases. The total came to 3 981. Now, here's the next piece of the puzzle. That many test cases are very difficult to create and maintain. Enter DSL – Domain Specific Language. Now, this term usually applies to either software engineering (as in a programming language like C#) or the manufacturing industry (think a car assembly line), but by using terms found in our software (and general user interface development) we could quickly change our test cases from the traditional 'ask-a-question-format' to a granular, searchable title based method. In addition, in this instance, it reduced the reliance on repro steps too as the tester can see what needs to be tested by reading the title.

Given that the test cases were now all in DSL format there was a side benefit: pattern matching. Programming is essentially pattern matching and problem solving, so I could write a program to dump the test case information out into an easily digestible Excel format for reporting to senior and executive management. This meant that missing areas or too low coverage was easily identifiable and remedial action could be taken early enough to have a positive effect on the outcome. The release was on time and within budget. As of yet we had had only positive feedback.

Biography

I have been in the IT industry for 14 years. 12 of those were spent on the development side of the business, before I moved over into QA. I've been in QA for a relatively short time, only 2 years at this stage, but have tremendous respect for QA as an industry. For the last 2 years I have been the QA Manager at SourceCode Technology Holdings, where I have spent the last 6 years of my IT career. We are the vendors for 3 products: K2 blackpearl, K2 blackpoint and K2 connect.
