orca
optimizing application ecosystems

Success Stories: Detecting and Correcting Drift in a Complex Environment

Snapshot

Organization:
Large Financial Services Company.
Application, Middleware & DevOps teams.

Location:
USA

Challenge:
To automatically find and fix non-compliant middleware configuration changes without requiring expensive and time-consuming IT staff training and learning new languages.

Environment:
JBoss, Apache, IIS, WebSphere Application Server

Solution:
Orca Drift Detection & Correction

Impact:
- Out-of-compliance configuration changes are now detected and corrected immediately.
- Zero (0) new scripting languages to learn.

> **"With Orca we can see the environment moving and see when someone does something nasty. We can keep rogue developers from doing something out of bounds. Orca saves time troubleshooting. It detects changes right away and it tells us what that change is, and who made it. We have Orca set up to automatically smack down any unauthorized changes."**

Background:

A multi-billion dollar financial services company had an ongoing challenge with developers and others making well-intentioned, but non-compliant configuration changes. Policies, procedures and training could only go so far. Unauthorized changes were still occurring. As the pace of software releases and other environmental changes increased this was understandable but still not acceptable. The consequences of bad configuration changes can include performance problems, outages or compliance problems. So the company needed a viable technology solution.

Alternatives Considered:

> **"Staying with Puppet was not an option since it would require several weeks of training to learn the environments and builds. We would have been faced with returning to manual builds. There were risks in relying on a tool that only one person had mastered."**

One option was to continue using Puppet. After all, the company had installed it and used it for a time. Unfortunately it was too difficult and time-consuming to train the rest of the team on the tool. Staying with Puppet was not viable. The company also considered Chef but discovered that it was "too much like Puppet" – still too complicated. Initial setup with Puppet had taken a month. And training staff would require several weeks of dedicated effort to get them to administrator level of proficiency. This was time that no one really had, but yet they had to decide on something.

Choosing Orca: "Ease of use was our key criteria."

The team was very close to pulling the trigger on yet another tool that was not a perfect fit, but then read up on Orca's drift detection capabilities. Orca sounded too good to be true. According to their team leader, "Ease of use was our key criteria." When it came to the product demo and proof of concept, they had no pre-conceived notions going in. Within hours, the team began using Orca. Within days the team was comfortably operating their complex environment using Orca. The team was also impressed that Orca only required one connection to their firewall, "rather than punching 300 holes in my network."

> **"My team could even use Orca during the setup process in the POC. Orca has a very well laid out GUI. You don't have to think. You don't have to ask how to execute commands. In the realm of automation tools, this is a first. Orca is actually easy to use. The learning curve is very small."**

Using Orca: Detecting middleware configuration drift their way

Supporting a large financial services company, the team has some applications that don't require a lot of security and others that require very high security. "With Orca we can treat them differently. With Puppet it was one-size-fits-all. It was checking every 15 minutes. We decided to have Orca check for changes every 5 minutes to make it easier to notice changes."

Results: ☑ Drift detection time: Immediate ☑ Drift correction time: Immediate

What's my drift detection ROI?
https://www.orcaconfig.com/roi-calculator/

Does Orca work in my environment?
https://www.orcaconfig.com/request-demo/