

Case Study

Automox



Brett A Haines

Vice President at Atlantic.Net, Inc.

- ✓ Review by a Real User
- ✓ Verified by IT Central Station

What is our primary use case?

We use Automox as part of one of our product offerings, our server management, where we use it for patch management.

As concerns the Atlantic.Net side, when we're selling to the marketplace, we sell mostly public cloud with dedicated hosts, or cloud VMs. We also sell hybrid cloud and private cloud, as well as co-location when that unique need arises.

Obviously, with so many servers out there, our clients are very big into compliance, examples being HIPAA, PCI, NIST, ISO certifications and the like. We need to be able to provide patch management and that's how we utilize Automox. That's what got us started looking for this back in 2017. We needed something as our in-house solution was not working very well in terms of what we wanted: visibility and up-to-date patching. At that point, we decided to explore

new open source options versus what is out there for purchase.

This is how we stumbled across Automox and started using it with our clients. We never truly made this kind of functionality into a formal offering before settling on Automox. It was more on an urgent-need basis. But, once we adopted Automox, we made it a formalized offering.

How has it helped my organization?

The biggest improvement to our organization is the man-hours it saves us. One of the reasons we shied away from selling these kinds of things in a formal product, prior to Automox, is simply that it would have taken two or three dedicated people just to do updates — and that was before we really started it. We had some formal things



for Windows in place, but because we've had so many different product offerings for operating systems, not just Windows, it was really tough to have everything covered under one solution, especially an open-source solution.

The savings in man-hours alone, in switching from that to Automox, means Automox has paid for itself. While I do not have exact figures, since then we've been patch managing five to seven times the number of servers. There's no way we would've been able to do that and still keep it cost-effective. How much time do we save? It depends on the load, since there is always that spike, given that Microsoft comes out with a new patch on Tuesdays. Not taking that into account, we probably spend about an hour a day, per person, meaning roughly five hours a week. That comes out to about 20 hours a month doing patch management through Automox, where in the past it was a full-time job for about two people.

Since we have scaled—and all the more so because in security compliance you have to test the updates, and do other things as well—we've probably saved hundreds of hours a month, at least. This amounts to significant savings.

What is most valuable?

The most valuable feature is probably the interface. Obviously, the work they do behind the scenes is important, such as: making sure that all the patches are there and making sure that everything is explained, such as what

requires a reboot and what does not. It saves us on much legwork by removing all that manual processing from our side. From our point of view, the interface is clearly super simple to use, super simple to get up and running. It also makes it very easy to digest the data.

When I look at the dashboard, I can see how many are scheduled for updates, how many are already fully up to date, and how many need attention. I can see if there are any exceptions that my people put in for the customer. It's one of those things where it's really easy for everyone to be on the same page.

I can go to the Control Panel and I can create different organizations. This way, not everything has to be under one single interface and account. We can split it out as we see fit. That was something that we wanted. While it was not a big deal, it is nice that I can now go in and see a customer who has 400 VMs with us in a single pane of glass. I can click and see where they stand, as opposed to having to go through thousands all mixed together.

What needs improvement?

We would like to see additional detailed reporting for Service providers like us. We had to build our own reports via their APIs to meet our needs.



For how long have I used the solution?

I have been using the product since June 2017.

What do I think about the stability of the solution?

Other than when maintenance is being performed, we have had no issues with the stability of the product, no downtime.

What do I think about the scalability of the solution?

The scalability has been great, especially now that we can segment out large customers.

We have significant plans for integrating it into our cloud portal. We can start with just the patching, which is what we are undertaking at present, and progress to offer it as a self-service model in our cloud portal.

How are customer service and technical support?

We have had no complaints about the technical support. We've used it twice in almost four years now and have had no issues, and no complaints from our side. It's been great.

Which solution did I use previously and why did I switch?

Prior to using Automox, we utilized open source and a Microsoft solution together. We would not consider the Microsoft solution to be a direct competitor of Automox. It was used solely for patching Windows systems, and the open source was used only for patching Linux-based systems.

There were a host of reasons that we decided to switch from this approach to utilizing Automox. The old systems were cumbersome, very hard to understand and to train on. They involved a long learning curve. Also, you were very limited in what you could and could not manage. You were very limited in terms of scheduling or the updates that you wanted to apply. For certain OS's, even with the open source solution, you could not patch Windows and you could not patch certain Linux flavors with it. Just that issue alone left us asking ourselves how we could offer a full-fledged product without even being able to do all the OS's that we offer.

That is when we started looking at a real system to replace it. We were not set on going with a cloud-native solution, but this being in the cloud reduced, by two VMs, what we would have had to manage, update, monitor and the like. That made life that much easier for my people.

We did make use of Automox's free trial prior to using it, which we considered extremely important. You may get promised the world concerning tech solutions, just to discover that



they often do not properly work. A sales representative may promise you the world and then not carry through. We actually got an extension on the trial because we were not yet ready. We then ended up paying for just one endpoint on a server. We were getting a little delayed. Once everyone received the approval from everyone buying in, we proceeded to push it live. If they still make available a free trial, definitely use it, because it makes life a lot easier.

How was the initial setup?

We required four staff in total for the deployment, involving two engineers who did the actual work, and two decision-makers. I found the initial setup of Automax very straightforward, very easy. My tech level is that I know what I'm talking about. I can kind of navigate pretty well, definitely not day-in and day-out, as I'm not in systems anymore due to my position. But, even if you told me to set it up today for a smaller-medium sized business that has 50 VMs, I could probably knock it out and do so without having too many headaches. That includes setting up policies, the VMs, and the hosts, and having everything set up and reporting back and making certain that we are hitting the criteria we set out to achieve. That is all pretty easy thanks to the interface.

Regarding the length of time of the initial deployment, we did not rush with the testing. We had a trial and then we asked for another

trial because we got pulled away due to what was going on with our customers and other systems. Once we decided to move forward with it, it took less than a week. This did not involve pushing very hard to get it done. It was more the case that my people had the time.

For our implementation strategy, we matched it up. The hardest part involved defining how we wished to put customers in at the time. That was before we had the ability to split them out into different segments. We had to figure out how to match what our other systems do, such as: customer number, customer name, and the relevant service policy. This is because some customers only want critical updates and some want all updates. Other customers want no updates to ever be applied. They just want to be notified of them and then they can update the patches themselves. We went in there, took our customer feedback and made it fit our needs.

What about the implementation team?

We did not work with any third-party integrator or a consultant to help out with the deployment. I had experience because I'm actually the person in charge of our Salesforce and its development. With Salesforce, you definitely need the help of a consultant no matter what. You can get away with a simple deployment of Salesforce. Once you start getting deeper into CPQ and the like, that this is where the devs come in. With this, there's no need for them. It's



very simple to deploy. We have two people maintaining it.

What was our ROI?

While I cannot supply you with specific numbers, the savings in man-hours alone has been a great ROI on the use of Automox. On the flip side, as we are heavily involved in the security compliance sector, we have to take into account HIPAA, PCI, SOC and the like. We need an offering for auto-patching. Whether or not the customer actually opts for it, the ability to provide him with the option at least gets us in the arena to bid on the deal. If we didn't have that as an offering, then we would have lost a lot more deals over the years.

What's my experience with pricing, setup cost, and licensing?

The pricing and licensing costs have been great for us. We got in early and then they had a little price rise. Overall for the price that they charge now, you get your money's worth on it.

For us, they had no additional costs beyond their standard licensing price. I don't know if they do setup now but we did all our own setting up and deployment. It solely involved licensing costs for each endpoint.

My advice to others who are evaluating or thinking of implementing Automox is to give it a

shot. If a free trial is still available, definitely use it, because it makes life a lot easier.

Which other solutions did I evaluate?

We did evaluate other options although I do not recall which ones off the top of my head. Most of them were not just patching solutions. They were a lot of other things too, and we did not need those extra pieces.

There are other differences between the other solutions and Automox, in terms of pros and cons. When it comes to solutions, we want to try to do open source first. We are very big on open source, our entire cloud platform having been built from the ground up by our development team. When comparing Automox to the open source solutions, we were finding many gaps where, perhaps, the open source could not support all the OS's that we wanted to support, or which involved much more manual labor, or involved delays in the patches being made available in the appropriate amount of time. There were a number of reasons that, once we started examining everything, it made much more sense to go with Automox.

What other advice do I have?

I recently asked my team about which areas of the product have room for improvement and for the moment we don't have any complaints, something I find anomalous. When I hear that



from my techs and engineers, it's always a great thing.

We're not heavy into the Worklets yet, so those may have room for improvement, as they are something new for Automox and for us. But overall, when it comes to patch management, the dashboard, ease of use, and the ease with which data is digested, have been great for us.

We're not trying to push what Automox is for. We use it for patch management as its key task and it's been fantastic for that. There have been no complaints by us so far.

Our staff who access it include systems administrators and systems engineers. Across our organization, we have around ten people who access it at least once per month. There is not a lot that needs to go into this. I haven't really had to do much work with it altogether during the past year or two. We really only need one or two guys in here, depending on whether Microsoft or, perhaps, Ubuntu drops a large patch. The amount of manual labor required of an employee is very limited, which is nice.

The biggest lesson that we have learned from using Automox is that as a managed service provider, it gives us a lot of insights into what our customers want. There are many customers who come in and state that they want everything patched up and up to date. Then, there are those who are the complete opposite. Most of our customers fall somewhere in the middle. Many of them do not have a proper understanding of why things need to be patched. But this has been very interesting

because once we went live, we started getting feedback from our customers and this gave us insight into other areas that our customers consider to be important.

The cloud-native aspect of the solution was not very important to us. We considered it to be a "take it or leave it" feature. Retrospectively, it is a great feature to have. It saves us from worrying about the servers or about updating Automox. So it does, in retrospect, make a lot of sense to go cloud native because we don't have to worry about it. We have had no issues, no downtime issues with Automox. There are no complaints in that regard. While being cloud native was not a key feature at the time, looking back it's pretty nice not to have to worry about it.

The solution provides us all the visibility we need, although we do not actually manage laptops. We only deal with what's inside data center walls. We actually use it for our employees' desktops and then on the server side. It gives us complete visibility.

While we don't have any macOS on here, the patch management definitely covers Windows, Linux, and even Unix. We love the overall patch management abilities. It's the main driver of why we adopted Automox, and it has definitely stood up to the test of time.

It is absolutely important to us that it provides a cross-platform patch management across Windows and Linux endpoints. This was one of the driving factors and among the decision-making criteria for us. We already had two different patch management systems in the past,



one of which could only handle Windows, the other only Linux. We were looking to try to unify that. It was such a pain since we were forced to bounce from one to the other. One of our older solutions capped out on how old an OS could be, since we have some customers who simply cannot move an application off an older OS. With Automox, the gamut is covered. That's what we want, that single pane of glass for patch management.

For the moment, we do not use the product for automation of patching. While it is definitely in our R&D pipeline to adapt it, we do not yet have it automated. They have the playbooks and they seemed very interesting to me. I'm just waiting for some R&D time on our side so that it may be integrated.

My people have been using Automox's Worklets for simpler tasks or those that are not overly complex. We're waiting to get these into our cloud portal. For the moment, we are making some use of them. We feel it is pretty important that Worklets enables us to enforce tasks across any managed endpoints. Especially in light of some of the vulnerabilities over the past two years, we want to enforce that our customers update. While we do not force updates, we want to make certain that the updates are covered and are applied in a reasonable amount of time. There are some delays with clients.

Consequently, this capability of follow up and notification, should it still be waiting, is very important to us. We're using some of the Worklets from the community, which is really

nice. One of them involves getting Windows Update events, and that is great because it's part of troubleshooting.

I've never had a complaint with Automox's speed. None of my people have had a complaint. We are satisfied with it.

While my technical level is not that of an engineer, I can set up VMware, Windows, and Hyper-V environments. I can do the basics and follow instructions. This solution was super easy. We just installed it on a server during the original testing and then had it phone home and that was simple.

We do not yet make use of it for API functionality, but this is something that we are looking into. It's part of our R&D plans to be able to push it from the cloud portal for customers. My people have already used it internally, but it's not yet for our clients.

Which deployment model are you using for this solution?

Public Cloud



Read 6 reviews of Automox

[See All Reviews](#)