The role of Kubernetes in cloud security

LACEWORK®

The landscape is changing at an increasing rate

Containerization has been widely accepted in the IT community as the go forward, cutting edge solution for new deployments. Take a look at GitHub, and the top category is guaranteed to be a game-changing tool like Kubernetes. Organizations are increasingly taking advantage of the on-demand pricing and incremental scalability of the cloud. While leaders and engineers alike know that Kubernetes is the future – a simple Google search of adoption rates is extremely telling – adoption has skyrocketed in the last few years, but knowledge of the platform is still lacking. Most organizations that are running trials or have no current usage cite that a lack of functional knowledge or ability to implement as their primary reasoning on why they have failed to adopt the tool further. Companies have to keep up or will be left in the dust by their competition. They can't do it alone, and need to implement tools, complete solutions, to solve their issues.

IT software implementation isn't the only part of the organization that has evolved over time. We've moved from a world where code is uploaded into a repository and then a process starts to review changes manually.

Looking at tools like Kubernetes - there aren't years and years of incremental updates to the software like you'd find in Windows, Linux or VMware. There's not a society of professionals who have built their careers on working with the software. The tool is new, and the experience level for those implementing the tool is limited. Additionally, if you look deeper at surveys from this result, you find that more and more of the community implementing the tool is not from a classic IT, but a developer background. The implementation of new tools is only increasing, and the likelihood of having a resource with truly deep expertise is becoming less likely, and often less needed. Kubernetes 1.0 released in July 2015, there are few people on the planet with years of experience with the system.

The safety nets are going away

So why are IT teams perhaps hesitant to implement this new technology and learn on the job? This question brings us to the history of information technology in business. Looking back to classic implementations – there was a very defined and controlled path. An IT leader or project owner would call up their VAR (value added reseller) and say that they are looking for a tool to solve a problem. Maybe they'd see the tool at a lunch or at a conference and want to learn more. From there, the vendor would come do a "dog and pony" show. They would roll out the red carpet for the potential of having a new customer. You'd get a team of people on site showing you everything the tool could do and a security blanket of people to call with questions. You'd tell them, "I need it to be secure", it has to pass my ISO/SOC/PCI audit compliance needs, it needs to cover both my build time and run time needs and they would work with you in making that a reality. Most likely, a proof of concept and terms would be defined and you'd try out the tool on some subset of your equipment for a month or two. Services teams would be called in to help implement, and account managers and support personnel would keep you happily using the new tool and new features as an ongoing security blanket.

Chances are, whatever hardware it took was already available to test on. You had additional capacity in your datacenter on old equipment to test out before you retired it – and there was always the safety of the implementation being behind your enterprise firewalls, in an internal VLAN, likely with physical access protecting the bottom line.

IT software implementation isn't the only part of the organization that has evolved over time. We've moved from a world where code is uploaded into a repository and then a process starts to review changes manually.

There's a manual QA process, a change management board review, and a human that actually deploys the code out through the environments. Releases frequently were a couple times a year affair, or maybe monthly in a forward-facing SaaS product. A security minded or titled individual was likely stationed as a barrier between new code and production, running vulnerability scans and penetration tests on internal environments prior to external facing environment approvals.

Looking back on it, it was a terribly slow and inefficient process but one that ensured that all of the I's were dotted and T's were crossed. We've evolved past what seems like ancient technology. There's a CICD pipeline releasing code through the various tiers of environments in an automated and repeatable fashion. It's no longer a once per month affair, many organizations are releasing multiple times per day. Automated QA testing has become a norm – but what about the security process? All of this speed in getting products out the door has left security behind, and the speed of business means it isn't a problem that can be solved with manpower. Tools are needed for security too.

"All of this speed in getting products out the door has left security behind..."

Flash forward to today, and those safeguards disappear. The age-old excuse of a system being insecure, and that being accepted because it is in the network, isn't valid anymore. What does a security focused leader do?

The creation of DevOps

In addition to the changes described, the walls between IT operations and development have started to decay. There is, and rightfully so, more and more focus on DevOps and building organizations that aren't a cost center but an organization that delivers towards the bottom line. We touched on one of the key tenants of DevOps earlier. To keep up, businesses are looking for faster delivery of new features. Environments need to maintain the highest levels of uptime, with limited required maintenance. Less maintenance leads to more innovation, and ultimately more time to solve business problems. Back to organizational survey results – the global consensus is that teams are more productive, happier, and more engaged with the shift to DevOps. With all these benefits the rigid structure of extremely functional IT roles has started to completely dissolve, depending on the organization. While this leads to greater opportunities – it also leads to hiring of more "jack of all trades types" than specific roles. The security of the product becomes a hat everyone wears, or sometimes a hat worn by no one at all.

The future

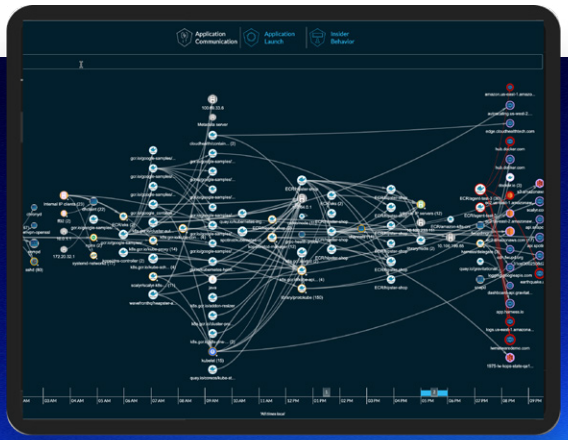
Reviewing the above, you end up with an organization that is evaluating business needs at a faster rate but is increasingly lacking deep knowledge and experience with the tools implemented. Many of the safeguards have dissipated and may not even be recognized as missing. The legacy systems and legacy organizational and bureaucratic structures of old won't keep a modern, continuous deployment, DevOps centric, cloud optimized organization safe. A quick feature release and solid code isn't going to matter to customers if it is insecure.

So, we've identified a need – but let's dive into solid security operations management starting with a comparison to a quality assurance organization. In quality assurance, bugs that make it past automated or manual testing out into production environments are typically referenced as escapes. Teams are measured on how many escapes they have and backlogs are formed for fixing these. Referencing easily searchable statistics, the cost of fixing an escape is, conservatively measured in 3-5 times the cost of fixing prior to release and these issues aren't necessarily blockers to impact critical business services. Take a look at a similar estimate for security related issues and the numbers are even more telling – most incidents are caused by an issue that could have been fixed for a fraction, a hundredth or thousandth, of the cost incurred resolving a security issue after the fact. A minor slip in a development environment can result in an exploited function in production. Some organizations base their whole business model on the security, or perceived security, of the system in place. Banks have vaults on display not for customer and employee convenience, but to showcase how safe your money will be in their care.

The tool must, therefore, be capable of being implemented in development and become as fundamental in the process of getting new features out the door as checking in code and running any other functional testing. It is essential that the tool is effective in build time –when code is being complied. Information security experts will always push for security to be involved as early as possible in the process. With the right tooling it not only can be, but must be a requirement. After all, the customer data and the organization depend on it.

Next, we've discussed the breakdown of organizational barriers and the buildout of teams responsible for more and more. The security role has dissolved further, and so the tool must fill in. Just as businesses are looking for CI/CD to deliver product faster, so must the security tools in place. A solution is needed that not only can maintain and manage itself over a long term with little intervention, but can also benefit and update itself automatically. Zero-day vulnerabilities are real, and a solid solution must receive updates in real time from a vast array of sensors. If a new vulnerability is discovered, it must react in run time. It must be capable of detecting suspicious actions prior to becoming a known vulnerability.

What separates good from great, and ultimately defines a world class solution, is one that meets all of these needs in a single service. While business should have a solution, they certainly don't want to add more overhead in managing a myriad of tools to get things done. A tool should be just as effective in build time as it is in run time – just as capable of meeting the developer's security need as the administrators. Unfortunately, most solutions and vendors are a point solution --something that solves one of these issues for a very specific functional area of the business. Finding a solid solution is like finding a needle in a haystack but they do exist. With the right solution in place, an organization should be able to state confidently that their systems are secure and safe, without having a dedicated individual or team reporting from a suite of tooling.



Ready to chat?

Request a demo