James Chad Ballay

**Cryptography in the Cloud**

*“Cloud computing is often far more secure than traditional computing, because companies like Google and Amazon can attract and retain cyber-security personnel of a higher quality than many governmental agencies.”*

* Vivek Kundra, Executive Vice President at Salesforce.com

*“The best way to get management excited about a disaster plan is to burn down the building across the street.”*

* Dan Erwin Security Officer, Dow Chemical Co.

In the earlier days of the project life cycle the physical limitations of infrastructure would act as a chokepoint on delivery. A 6 week delivery date for getting a new db server racked and wired on the raised floor wouldn’t be that unexpected. To try and work past this, people would overbuy and over provision and hope to make things up on the backend. This often would lead to projects mired in stasis while waiting for the hardware to be setup. (Kim, Willis, Debois, Humble, & Allspaw, 2016, p. 6). With the physical limitations abated the next limitation was the project methodology choice. Waterfall was found lacking and after much trial and error Agile has risen to dominance. This will change over time but the underlying advancement is to quicken the pace of delivery by removing inefficiencies.

These two critical leaps in approach laid down the emergence of modern Continuous Integrations and Continuous Development pipeline. Constant yet minor improvements and automated, regular deployments of these changes to foster a near real time monitoring and adjustment of the project’s direction. Freed from the limitation of infrastructure, developers are able to create adhoc servers automatically. Instant feedback on the success of a change minimize the disconnect between thinking about a requirement and implementing the requirement. (Forsgren, Humble, Kim, 2018, pp. 41-42)

This confluence yields an unintended consequence. Security needs have increased due to project resources moving outside of the walled garden of our internal infrastructure. Language libraries, OS patches, externally hosted SaaS products, etc, etc…. All of these are tools that allow for the modern product lifecycle but they do not play well with standard security best practices. Developers on a whim can pull in a Node.js library with malicious code in it. OS patches need to be applied continuously but how do we know that the patch is a valid patch? Never mind the loss of control of physically hosting your own data and now trusting a third-party. The very nature of a cloud provider is to provide ubiquitous access to all their customers. Yet we \*must\* trust them to only allow access to correct data to the correct people.

Take this CICD pipeline as an example. (Garros)

CI/CD Pipeline for Software Development
Code Build Test Deploy Monitor
Dev
CI
Continuous Integration
CD
Continuous
Deploym...

**Github/GitLab** – Source Code repository. Can be hosted internally but many software repositories are externally hosted. Without cryptographically secured access, unauthorized changes to code could introduce vulnerabilities into your code or the libraries/tools/etc you rely on.

**Jenkins/Selenium/Load Runner** – Automation Server used for testing that helps to shift security checks to the left in your CI/CD pipeline. Best practices can be tested here. Also relies on cryptographically secured access.

**Docker/Vagrant** – lightweight developer environment. These are all mechanisms that compartmentalize developers. OS updates, A/B testing, abstraction of credentials, etc, etc…. These all require cryptography.

**Terraform/Puppet/Ansible** – Automatation configuration tool that replaces the custom bespoke setting up of an environment and instead is replaced with unmanned deployments. Access is authenticated and secured using cryptography.

**Grafana/ElasticSearch/Dynatrace** – Monitoring visualization tools. Who watches the watcher aptly applies here.

At no point throughout this whole cloud enabled pipeline does cryptography not play a crucial part of securing and protecting it. For each of these stops in the pipeline there exist multiple tools and approaches that implement the same functionality. No person/team/company can master each of those technologies as well as the underlying reliance on cryptography. For this the economy of the cloud comes into play.

Validating ACL(Access Control Lists) is a minor part of my role on my team. Maybe 10-15 mins a week will they come up. I don’t have the spare cycles to deep dive into learning this. But I don’t have to. Instead we’ve stood up a single team with the responsibilities that cover ACL requests. The same can be extended to cryptography. I cannot spend enough focus to understand and keep up with the field of cryptography. That just isn’t a core competency for my role. It’s a core requirement so to accomplish this we ship that off as a deliverable for other stakeholders. No source code repository tool gets past an initial look if it isn’t demonstrated as being secure. No provisioning or configuration tools can be used if they don’t meet compliance requirements for authentication and logging. Etc, etc….

Without cryptography we go back to the days of walled gardens and the inherent slow delivery timelines and limited production rates. That Iron Age of computing is no relegated to being only used in extreme corner cases. The Cloud Age of computing is the status quo and cryptography is a non-negotiable for it.