BLOG POST

Moving to a cloud, not a storm

Avoiding common problems when moving to the cloud.

Nigel C

In my previous blog post on cloud security, I said you should concentrate on controlling data access, rather than worrying about the underlying security of the public cloud. In this blog post, I'll outline how you can avoid some common pitfalls, and why you should simplify your permissions model to get the most out of the cloud. Moving to the cloud can be a complex process, and shouldn't be made more complex than necessary.

Don’t stop at 'lift and shift'

'Lift and shift' is the practice of replicating an existing local system, but in the cloud. It’s a straightforward way to migrate, but it's number 4 in the NCSC's list of anti-patterns ('Building an 'on-prem' solution in the the cloud'), and should therefore be avoided. The cloud version of the system will retain all the security complexity of the original one, but will add the complexity of forcing cloud security systems to emulate local ones. There is a risk that this emulation will be incomplete, and won’t offer the security it is supposed to, while not achieving the security benefits offered by the cloud.

This 'lift and shift' approach can also prove costly. For example, by replicating a packet firewall device when packet filtering is available for free as part of the network configuration.

I suspect a lot of cloud migrations begin with good intentions; the intent is to 'lift and shift' as just the first step. But the impetus (and budget) to refine the

migration runs out after that first step. Instead, you should try to redesign or refactor the system to use the managed services available. It's fine to start small and look for 'quick wins', where an existing cloud service can easily replace part of the system being migrated, but **make that next step**. This will reveal the advantages of migrating further, towards a more 'cloud native' architecture, in terms of cost or complexity reductions.

You should try to reduce the complexity of the system before moving it to the cloud.

Don't make data public unless you want it to be 'public'

Making data or a service 'public' means making it available to anyone. This is probably the easiest mistake to make when moving to the cloud.

'Public' is similar to 'world readable', something addressed in Andrew A's 'My cloud isn't a castle' blog post. On a local system, we're used to the idea that making something 'world readable' means making it available to everyone on the local system. Moving to the public cloud, it's easy to associate making something 'world readable' as making it available to **everyone** in the organisation's cloud account.

Unfortunately, the word 'public' is there for a reason; 'world readable' is more literal, and includes everyone on the internet.

Moving to the cloud requires a bit more care to ensure that access control is enforced, particularly in areas that didn't previously need it. This is another reason why lift-and-shift is bad; it's easy to overlook new security requirements introduced by the cloud.

To summarise: don't make something world readable unless it really doesn't matter if **anyone in the world could read it** (such as data used by static web pages that anyone can read anyway).

Replace parts of the system with managed services

You should favour managed services (and serverless functions) over custom virtual machines. Or, in cloud terminology, favour SaaS and PaaS solutions over IaaS solutions. The cloud provider manages some or all of the maintenance of managed services and serverless functions, whereas (for IaaS solutions) the customer has to maintain their custom virtual machines. This is in line with the government's 'Cloud First' policy.

This is a smaller scale version of the lift-and-shift problem; it's lifting-and-shifting something like a database, effectively duplicating in a virtual machine something that the cloud provider already supports natively (and more efficiently). As with lift-and-shift, it's not making efficient use of the cloud.

There are bound to be occasions where a custom database implementation or a bespoke virtual machine is the right choice, but confirm this rather than make assumptions. As much work as possible should still be handed over to managed services and serverless functions to reduce the maintenance burden, so you can concentrate your effort on other areas.

Use role-based access control

Public cloud providers typically support role-based access control (RBAC), which can simplify how access control is managed. Instead of a user accumulating all the permissions they need, they are granted the permissions of the role they need to perform, and lose those permissions when they change roles. This makes it easier to restrict access to just the rights actually needed, since they only need to be set for the role rather than for every user needing access. This is included in the NCSC's Cloud Security Principles (9. Secure user management).

Even with RBAC, it's still a good idea to monitor access controls so out-dated permissions are removed (both in the sense that the role might have permissions it no longer needs, and users might have access to roles they no longer need).

Simple local security features might not be simple in the cloud

Another smaller-scale example of 'lift-and-shift' thinking involves migrating local security features verbatim into the cloud, since some local security features don't migrate easily. If the benefit of a security feature to the cloud is hard to articulate, it may be a sign that it doesn't work in the cloud. For example, having a single machine act as an internet gateway is a common security feature, even though it risks a 'crunchy on the outside, soft on the inside' security anti-pattern (criticised at least as far back as 1994, and also touched on in our Preventing Lateral Movement Guidance). But it's less useful in the cloud.

In the physical world, adding a second internet gateway to a network would be a significant amount of work. A new administrator unfamiliar with the system would likely recognise that it can't be the right approach, and a rogue administrator would be at risk of detection. In the cloud, an administrator can easily add a new connection to the internet, deliberately or accidentally, with just a few clicks. That doesn't make such gateways useless, but it's unsafe to **rely** on all connections going through a single gateway unless the network is monitored to detect and remove new gateways as they appear (or unless everyday administrators are not granted permission to create internet gateways).

Complex local security features might not be complex in the cloud

The converse is also true; some security features that would be complex for a local system are simple in the cloud. The cloud often supports global monitoring or templated features that would mean a lot of work to implement on a local system.

For example, putting a basic packet firewall on every network link and collating the results would be an administrative nightmare on a local system, but cloud systems usually come with this capability by default. Adding a few rules can improve network segmentation and limit the damage if an attacker gets in.

That said, don't go overboard and create so many rules that they become difficult to understand and maintain!

Cut the cruft!

Keeping security simple has always been good advice. More than 20 years ago, Bruce Schneier noted that "the worst enemy of security is complexity". The more complex the security, the harder it is to understand, and the greater the likelihood it's not doing what you think it's doing[1].

It should always be clear **who** has access to any resource. Equivalently, it should always be clear why someone does **not** have access to a resource. Moving to the public cloud is a good time to simplify the permissions model; look for opportunities to remove any cruft that has accumulated over time and any 'special cases' that turn out not to be so special after all. Pruning unnecessary complexity makes any holes in the security easier to spot and fix.

Use Infrastructure-as-Code

Infrastructure-as-Code refers to a variety of techniques to create infrastructure based on templates or scripts. Using Infrastructure-as-Code, ideally to implement the entire system, encourages thinking in terms of commodity computing, and has several security benefits (such as reducing errors). Think of it as a combined checklist and documentation for setting up the system.

This is a lot to learn about when moving to the cloud; it's understandable that it's left until later, and it's why it's the last suggestion in this blog. But there are a lot of benefits to getting to grips with infrastructure-as-code, and a time when you're already cataloguing everything on the existing system is the ideal time to deploy it. Doing so encourages you to find ways of simplifying the system, and it saves doing it all again sometime in the future.

Infrastructure-as-code lets you recreate your whole system from scratch. This is clearly valuable for disaster recovery, but it also makes it easy to set up duplicate systems for resilience, or to temporarily make a copy of the system to test system changes or upgrades. If a system needs to be moved to another cloud, an Infrastructure-as-Code description is a place to start from, and will reduce the work involved (though the current proprietary nature of public clouds means there's some vendor lock-in whichever is chosen).

Conclusion

Moving an existing system to the cloud is a major undertaking. The temptation is to take the simplest route, but a bit of planning and preparation can make the difference between a successful cloud migration and an unsuccessful one.

Nigel C.
Platform Security Research

[1] The NCSC Secure-by-Default Principles cover expand on this:

- security should never compromise usability – products need to be secure enough, then maximise usability
- security should not require extensive configuration to work, and should just work reliably where implemented
- security should not require specific technical understanding or non-obvious behaviour from the user

WRITTEN BY

Nigel C
Platform Security Research

PUBLISHED

21 October 2020

WRITTEN FOR

Small & medium sized organisations

Public sector

Large organisations

Cyber security professionals

PART OF BLOG