Firewall Policy Management in the Cloud

By Chris Brenton

For the last 20 years firewall rule management has changed very little. Sure we’ve seen enhancements get added and GUI’s become prettier, but how we go about architecting and organizing a rule base has remained pretty much the same. With the introduction of IaaS cloud however, we’re poised to see a rethink of how we structure and manage rules within our firewall policies. In this post I’ll talk about why.

Figure 1 shows a classic Gen2 network infrastructure. There are two firewalls in use, one between the Internet and the internal network, and another between the internal network and high asset value servers.

When we look at the hierarchical structure of how rules are managed, it looks something like this:

- Physical location of the firewall
  - Logical location of the assets (which subnet)
    - Function of the server (required network services)

So I may logon to a specific firewall in order to manipulate some portion of the rules I need to manage. Within that firewall, rules are most likely organized based on the logical segregation of the network. For example the first group of rules may be “traffic entering the DMZ”, the second group may be “traffic leaving the DMZ”, the third group may be “traffic leaving the
internal network”, and so on. Within these groups I may be leveraging additional subgroups to further simplify management (as an example one rule that defines a specific traffic pattern for all Web servers). While on the surface this management may seem complex, a proper structure is critical as it is not uncommon to have hundreds, possibly thousands of rules that must be managed, even in a simple environment such as the one shown in Figure 1.

**Public Cloud**

Obviously once we start adding in public cloud hosting, firewall rule management can become even more unruly. An example is shown in Figure 2. If we continue to apply the above hierarchical structure, management can quickly turn into a nightmare. This is because we’ve increased the number of points on the first level of structure, namely the number of individual firewalls that must be managed. The complexity quickly runs down the structure, increasing the number of unique rules that must be managed.

Further, I may be forced to manage these rules across multiple firewall products. For example I may be using Cisco firewalls internally, but the first public cloud vendor may only support Check Point firewalls. The second may only support Juniper. So now I not only have a whole slew of rules to manage, but I need to do it across multiple vendor interfaces.
This setup also introduces a new problem, namely security becoming an inhibitor to bursting. Assume I have a server running on my DMZ that I wish to burst out to one of the public vendors. How do I handle security change management? It is most likely a manual process to delete the rules associated with that server on the perimeter firewall, and then add them in on the public provider’s firewall solution. Does this mean the server cannot be bursted until the changes are cycled through the firewall management group? Will this introduce unacceptable down time and latency? What if the burst is a high priority due to a sudden increase in traffic patterns?

The Impact of Mobile Computing

As I’ve written in the past, mobile and cloud computing are becoming intertwined. Public cloud has become the delivery mechanism of choice for offering mobile content, applications and storage. This market shift is making network security less relevant and changing the way we deploy customer facing resources.

For example, to deal with high load in the Gen2 days, we would deploy some servers at high bandwidth co-location centers, install a few load balancers, perform a bit of geo-location DNS trickery, and we were off to the races. The goal was to get the resource as logically close to the consumer as possible. Today however, we have the ability to be far more surgical in our resource deployment.

For example, let’s say I’m about to hit the US market with a mobile application. If the goal is to get our resources as close to the consumer as possible, does it not make sense to spin up servers on Verizon and AT&T’s public clouds in order to service each provider’s mobile customer base? If I next shift my release out to Asia, I simply follow the same formula of leveraging the public cloud offerings of local TelCos. With some good back end management, I can be extremely precise in identifying exactly how many servers I need to have available within each cloud at any given time. This permits me a much higher level of optimization efficiency than was possible under the old Gen2 model.

Oh but wait a minute, what about security? With all of these servers spinning up and down, how do I scale my firewall rule management to compensate? Note that as we scale out the number of data centers, attempting to manage a security choke point at each location becomes an immediate inhibitor. So we have DevOps saying “This solution makes us agile and scalable” while the security team is saying “You want to do what???”.

Resolving The Issues of Firewall Policy Scaling

So how do we remain agile and scalable while continuing to enforce the appropriate amount of risk management? One of the easiest ways is to simply chop off the first few levels of our rule base’s hierarchical structure. We then scale out the lower levels as required.

For example, assume we need to secure a public facing Web server. Regardless of whether that server is located on the DMZ, Vendor A’s cloud or Vendor B’s, it is going to require the same level of risk management. In fact, we probably have multiple public facing Web servers that all require this same level of risk mitigation.

If they all need the same level of risk mitigation, does it really matter where the VM is being executed? In other words, if we switch from controlling network traffic on the wire, to controlling
network traffic in and out of the VM, we now have a portable security solution that scales as needed and is location independent. We can now set a single firewall policy that can be enforced regardless of execution location. Firewall rules can be set once, and no longer need to be updated if the server switches locations.

So how do we create portable firewall rules? The simplest implementation is to leverage the firewall built into each operating system. If we define our rules within the host based firewall, we now have a risk mitigation policy that is not dependent on logical location, and can scale as required. All we need is a decent tool for managing the built in firewall on multiple systems simultaneously. Luckily these are just starting to hit the market.