CLOUD SECURITY NIRVANA AT AWS

My server! So near...
Yet chowned. Data not found. Note:
Next startup, use cloud.

Having your data center next door under lock and key, seeing the neat row of blue LEDs blinking, and hearing that reassuring hum day after day would lull any Operations Manager into a false sense of security. The fact of the matter is, if those systems are on the Internet (which they probably are), then your confidential intellectual property on those systems, your key data and corporate assets, are no more or no less secure in that physical data center than they are in the AWS cloud.

Back in the old days of cloudless data centers, we used to examine seven key areas during the course of any environment security audit. These were:

1. The Perimeter: Traditionally the domain of your carriers, firewalls, IPS/IDS (and before that, the bastion hosts)
2. Network Security: Internal routers, switches, trunks and VLANs
3. Systems Security: Operating systems and core stack
4. Apps Security: Fun things like cross site-scripting (XSS), SQL injections and buffer overflow
5. Physical Security: Dealing with man-traps and mouse-traps
6. Policies and Procedures: Who has access to what, is it monitored, backed up and documented
7. External Factors: Information which may or may not be fully in your control (registrar and public records, social media postings of employees) which could be harvested to compile your online profile

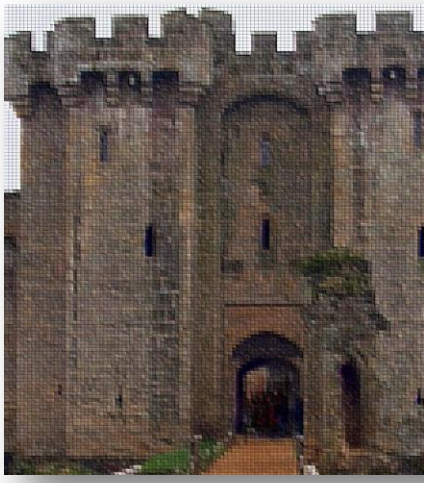
There's been a persistent stigma associated with cloud computing security. It is true that there are some differences in how each of these areas is treated in a cloud-based environment; however, the notion that the cloud is inherently less secure is no longer true. The reality is that AWS provides a ton of security features and functions at low-to-no cost that are just not available in a traditional data center environment. Just like everything else in the world of systems, these need to be understood, applied and used in a holistic fashion to achieve security nirvana. In case you have been sitting out the cloud computing paradigm shift so far, here are some interesting parallels:

CONTROL ACCESS TO YOUR (CLOUD) DATA CENTER

The AWS console is essentially the doorway of your cloud data center. You get access to it when you first set up your account. At this point you are building the walls of your monkey cage or your data center - you and only you have the "root access" keys. AWS console access can be protected by Multi-Factor Authentication (MFA) using hardware tokens or a Time-based One-Time Password (TOTP) mobile app like Duo or Google Authenticator.

AWS provides a facility called Identity and Access Management (IAM) which allows you to select folks on the team who have access to the console. You can create groups of users and assign them different sets of policies, effectively building additional walls within your data center to limit access for individuals or groups to perform a limited set of tasks, or to work on specific systems. These IAM policies provide a more flexible and granular level of control than you can achieve with physical walls. Entry into the AWS console for IAM users can be configured through a custom URL. Individual pairs of secret and access keys can be generated for users who prefer CLI over the console GUI, and the AWS STS (Security Token Service) facility can be used to obtain temporary limited-privilege credentials from within your code. After all, the cloud is all about programmatic deployment.

THE PERIMETER AND NETWORK ARCHITECTURE – DESIGN IT RIGHT

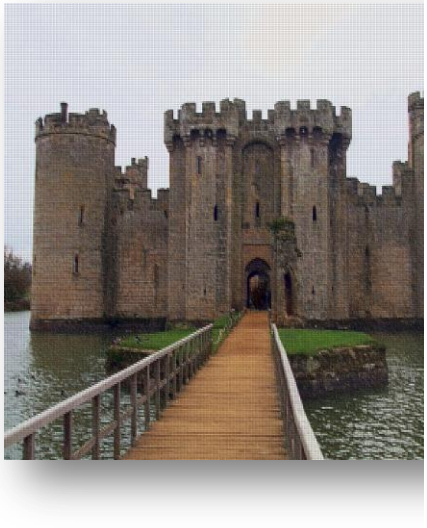


Remember all the white-boarding sessions to carve up your network into multiple subnets with DMZs, firewalls, routers, core switches, etc.? Cloud security still mandates good design (along with good white-boarding skills), but implementing it has become a whole lot easier. AWS has the concept of Virtual Private Clouds (VPC) that you can quickly configure in your environment with your own private IP address schema. Think of this as your locked core, but without the bother of the actual switching gear which would require its own security set-up in the physical world. VPCs can be deployed in a multi-zone architecture across an entire region (e.g. the US-East-1 region, which comprises multiple zones or data centers). HQ can access all resources within a VPC through a site-to-site VPN that's set up with your FW/Router or VPN concentrator (e.g. Cisco ASA) at your office, and multiple VPCs can talk to each other through a controlled VPC peering arrangement. More elaborate infrastructures can be designed by employing solutions from vendors in AWS's ecosystem such as Palo Alto Networks, Fortinet, Brocade, and OpenVPN Technologies.

You can carve up multiple subnets within the VPC and set up routing tables to designate private and public subnets, with individually configured NACLs to control the type of traffic allowed in that subnet. Instances spun up in the public subnet can be automatically assigned a public IP address with a direct route out to the Internet, whereas instances initiated in the private subnet would only get internal addresses from the subnet block, and would need a NAT gateway to communicate out over the Internet. Public IP addresses are typically non-contiguous, adding to security by obfuscation, making it difficult to guess non-DNS'ed services, and reducing the likelihood of a planned DDoS attack targeting a corporate IP block. Having your systems at AWS doesn't mean that you cannot work with your favorite DDoS vendor like CloudFlare or use CDNs like Akamai; external services like these can be overlaid on top of your architecture.

Just like most other services at AWS, the entire VPC-based infrastructure can be scripted and set up (or torn down) with the push of a button, using AWS's CloudFormation.

SPIN UP THE REST



AWS provides standard “machine images” or AMIs of most popular Linux and Windows flavors that you can “spin up” in your environment. There are some important security elements that come into play here. For starters, you need to identify the VPC and the subnet where you want your new instance spun up. If the instance is not providing any direct public-facing services, then it really does not need a public IP address, and should be spun up in a private subnet. Even web servers do not need public IP addresses if they are set up in a farm, fronted by an AWS Elastic Load Balancer (ELB); the ELB is the only component that needs to be externally available. Other load balancer solutions, e.g. F5’s BIG-IP are available through the AWS Marketplace.

The next piece of configuration has to do with additional data drives or volumes that you may need to attach to your instance. These can now be optionally encrypted, and you should choose to do so; the same is true of RDS instances.

Next up are Security Groups - a collection of custom policies specific to the VPC that get applied to each individual instance and define exactly who has access and to which port or service on that instance.

The final piece of configuration is assigning a key-pair to access the instance. Passwords alone are woefully inadequate to secure any system, so AWS provides an easy mechanism to associate and use RSA-based keys for systems access. You can create key pairs for each individual system, but it’s probably easier to share keys between similar groups or clusters of servers.

Your instance is now ready to be spun up. And if it is an internal app server, then congratulations! You probably have your new instance in a private subnet, accessible only through the VPN by selected IP addresses, for the use of selected services, and administered by a select group of people.

You have put up your defenses around the system, but you do need to harden it on the inside as well. If you chose to go through the hardening exercise yourself, then you can easily create your own private set of secure AMIs to use across your environment. Keep in mind that your instances still need to be maintained and patched on a regular basis, which should be automated through tools such as Chef, Ansible or Puppet. Alternately, you have the option of picking pre-hardened AMIs provided by organizations such as CIS through the AWS Marketplace. The AWS Marketplace is a great resource to find and deploy AMIs and cloud-based solutions from your favorite technology vendors like Barracuda, Trend Micro and Cisco to address specific requests or requirements.

KEEP IT RUNNING SECURELY

Beyond architecture and implementation, there are several operational elements that deal with maintaining a stable and secure infrastructure.

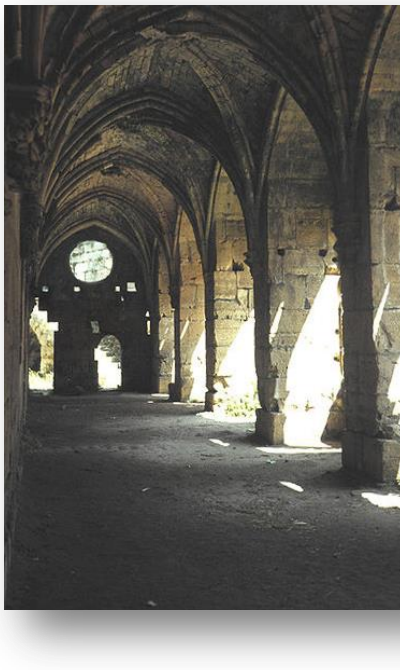
Logging, Monitoring, Alerting

As an experienced IT professional, you are not going to fall for any phishing or social engineering exploits, right? But sometimes you do pass stickies and “temporary” passwords in clear text emails. A corporate entity can put strict policies in place to discourage such practices, but it is difficult to police individuals to this extent. What is needed, however, is a rock-solid monitoring system that can alert in the event of an unauthorized access attempt or systems anomaly. AWS provides some useful monitoring tools like:

- CloudTrail and Config: For tracking all AWS user activity, inventory and configuration changes
- Trusted Advisory Reports: For providing on-demand analysis on the status and security of your AWS account and environment
- Billing Alarms: For providing alerts if charges or service usage exceed defined thresholds

However, you still need to set up your own real-time monitoring facilities using time-tested tools like Nagios or Zabbix (or alternately commercial tools such as Datadog or Dynatrace). You can set up a monitoring server on your internal subnet; use CloudWatch to monitor the monitoring server itself, and use an external monitoring facility (e.g. Pingdom) to test the user experience and collect metrics on any public facing services.

And yes, you still need to monitor all those logs for signs of forced entry or other malicious activity. You can set up a centralized logging facility with the Elastic Stack, Loggly or Splunk to analyze your logs; alternately, if you don't want to run a 24x7 shop then consider utilizing a service like Alert Logic to watch over your instances. Tie in alerting and incident reporting with PagerDuty or OpsGenie to set up your on-call rotations and escalations. Remember that there is no magic bullet for poorly designed or obsolete apps, so it is critical to run vulnerability scans across your environment, at least as frequently as warranted by the regulatory compliance requirements of your industry - another area where Alert Logic can potentially help.



Build Recoverability

"Backups" is one of the dirtiest words in any respectable Ops Manager's run book, but one of the most important functions in the safety and security of your environment. AWS provides the ability to create on-demand snapshots of your EBS volumes, and automatic snapshots of RDS instances with point-in-time recovery. Snapshots are fast, inexpensive, and programmable. They can be copied and stored off-region to provide an effective pilot-light disaster recovery solution. For real-time DR / replication capabilities, there are more sophisticated solutions from companies such as CloudEndure that are available through the AWS Marketplace.

IN SUMMARY

Creating a secure AWS environment needs to combine a multi-layer approach with a variety of building blocks and controls available in AWS's ecosystem in order to achieve security nirvana. Here's a summary of the lessons learned:

- Control access to the AWS console (your virtual data center): Use IAM, with MFA. Enforce password policies, and different roles & policies for the creation & deletion of resources.
- Control your perimeter and network security: Design a scalable VPC with a layered subnet architecture to accommodate multiple public and private subnets. Use NACLs to control the type of traffic allowed in any subnet. Limit the use of public IPs.
- Control your systems security: Use hardened AMIs - from the AWS Marketplace, or your own. Use Security Groups at the instance level to control access to known services from known hosts. Use encryption where you can (e.g. for EBS volumes).
- Standardize your builds: Use automated tools like AWS's CloudFormation combined with configuration management tools like Chef, Ansible, etc. to build and maintain your environment.
- Monitor your environment: Use AWS constructs like CloudTrail, Config, Trusted Advisory Reports, and Billing Alarms in addition to other logging and monitoring tools.
- Backup your environment: Use snapshots. Store a copy off-region or in a separate account.

RESOURCES

- AWS: Cloud Security Features
- Akamai: Cloud Application Delivery Solutions
- Alert Logic: Cloud Security for AWS
- Ansible: Automation Solutions for AWS
- Barracuda Networks: Web Application Firewall on AWS
- Brocade: VPN/Firewall/Router Products on AWS Marketplace
- Chef: Automation Solutions for AWS
- Cisco: Cisco Products on AWS Marketplace
- CloudEndure: Disaster Recovery Solutions on AWS
- CloudFlare: Security Solutions
- Datadog: Ops Monitoring
- Duo Security: Authentication Solutions
- Dynatrace: AWS Monitoring
- Elastic: Elastic (ELK) Stack
- F5 Networks: Application Delivery Services Platform on AWS
- Google: Google Authenticator
- Loggly: Log Management Solutions for AWS
- Nagios: Monitoring Products
- OpenVPN Technologies: VPN Products on AWS Marketplace
- OpsGenie: Alert Management Services
- PagerDuty: Alert Management Services
- Palo Alto Networks: Next Gen Firewall on AWS Marketplace
- Pingdom: Monitoring Services
- Puppet Labs: Automation Solutions for AWS
- Splunk: Data Analytics Products on AWS Marketplace
- Trend Micro: Security Products on AWS Marketplace
- Zabbix: Monitoring Software

CONTACT INFORMATION

TGIX is a certified Advanced Consulting Partner with AWS and has worked with many organizations, guiding them through cloud adoption strategy, architectures, implementations and support. Contact us for a complimentary evaluation of your current network and security architecture in AWS.

Address: 2 West 45th Street, New York, NY 10036
Web: http://www.tgix.com
Email: info@tgix.com