TGIX

AWS Security Architectures for Highly Regulated Industries

Achieving a best-practices security posture

An In-the-Trenches Viewpoint

Table of Contents

AWS Security Architectures for Highly Regulated Industries

AWS Security Architectures for Highly Regulated Industries

Achieving a best-practices security posture

Businesses around the world today need to comply with an ever-increasing roster of regulatory requirements to protect their customers, with highly regulated industries such as healthcare and finance facing especially stringent requirements just to keep the doors open much less remain competitive. Of all the compliance issues, the ones dealing with security aspects of IT infrastructure and operations are especially onerous, with increasing and rapidly changing complexity—and it's crucial to keep up to protect both the company and its customers. In this environment, understanding cloud security architectures and tooling for rapidly evolving risks is critical for the CISO or any other executive charged with managing regulatory compliance functions.

MANAGING DATA SECURITY

4 Areas of Focus

No matter the regulatory compliance body or standard that one needs to deal with, at the end of the day it's all about the data. Specifically:

Identifying all data that is core to the regulatory issue, be it PHI, PII, PCI, or other sensitive data, and the systems and networks where it resides.

Protecting sensitive data, whether at rest or in motion, via access controls, encryption, backups, and any other means necessary.

Monitoring sensitive data, who/what is accessing it, how is it being used, and recording the audit trails and access.

Alerting appropriate personnel as soon as protective measures are breached and sensitive data is exposed.

This whitepaper outlines the best-practices architectures and tools to specifically to address these data privacy and security requirements in an AWS environment. Following this approach will not only improve your security posture, but save you time and headaches when it's time to run your external audits and validations, whether for HIPAA, SOC 2, FedRAMP, etc.

AWS's Shared Responsibility Model

When dealing with AWS environments it is necessary to understand AWS's "Shared Responsibility Model," which establishes that Security and Compliance is a shared responsibility between AWS and the customer. Specifically, AWS is responsible for the security of the cloud (i.e. of the infrastructure building blocks), and the customer is responsible for security in the cloud (i.e. of their environment, data and applications running in the cloud).
AWS supports a large number of security standards and compliance certifications including PCI-DSS, HIPAA/HITECH, FedRAMP, etc (a comprehensive list is available at aws.amazon.com/compliance/programs/). Customers inherit the controls provided by AWS through which they are able to satisfy a large number of compliance requirements. There are additionally a number of security features and functions that AWS provides at low to no cost. When used in conjunction with third-party tools and techniques described below, you can design a holistic solution to meet your regulatory requirements and solidify your security posture.

AWS Multi-Account Architecture & Provisioning

Ensuring a rock-solid systems architecture foundation is a crucial precursor for protecting sensitive data. Gone are the days of single-account AWS architectures, which used multiple VPCs (and messy VPC peering) to enforce isolation between different environments such as production and development. Best-practices now dictate setting up a multi-account architecture under a single AWS Organization umbrella, with different environments isolated in their own accounts and connected via AWS's Transit Gateway. This architecture ensures that sensitive "live" data remains restricted to separate production accounts, which can be managed and configured with additional lockdowns. Separate AWS accounts should be set up for shared services and management functions, with the accounts accessed through a custom Single Sign-On (SSO) portal using AWS's SSO or an external IDMS such as Okta. (See our whitepaper on Multi-Account AWS Architectures for a more detailed look at this design, including security controls such as Service Control Policies (SCP) that can be centrally administered through the Master account.) Next, create groups of users through the SSO or IDMS system and assign them different sets of policies, effectively controlling access for individual users and groups to a limited set of tasks or systems.

This eventually ties back to AWS's Identity and Access Management (IAM) roles and policies that provide fine-grained access control. Ensure that adequate password policies are defined that enforce Multi-Factor Authentication (MFA) for all users including root users of the accounts. For programmatic access, AWS STS (Security Token Service) should be used in conjunction with IAM roles in lieu of secret and access keys to obtain temporary limited-privilege credentials.

Core Network Architecture

With the AWS account provisioning out of the way, it's time to turn our attention to the issue of data protection, which demands that the cloud network and perimeter be designed with security and high-availability in mind. Ensure that your VPCs are consistently designed with appropriate CIDR block routing and are deployed in a multi-zone architecture across your preferred region. Multi-cloud or hybrid cloud environments will need Direct Connect circuits or site-to-site VPNs set up with your firewall router or VPN concentrator (e.g., Cisco ASA) at the physical location, along with a Transit Gateway configuration to handle routing. More elaborate infrastructures can be designed by employing solutions from vendors in AWS's ecosystem such as Palo Alto Networks, Fortinet, and OpenVPN Technologies.

The VPCs themselves can be carved up into multiple private and public subnets, with individually configured routing tables and 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 noncontiguous, adding to security by obfuscation, making them difficult to guess, and reducing the likelihood of DDoS attacks 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 ervices like these can be overlaid on top of your architecture. AWS has its own Cloud Front CDN and Advanced Shield DDoS protection services if you so prefer.

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 Terraform or AWS's CloudFormation. This is also where we start monitoring our data flows by ensuring that VPC Flow Logs are enabled and saved to encrypted S3 buckets for auditing purposes.

Working with Cloud Instances

When it comes to plain vanilla compute (or EC2) instances, AWS provides standard machine images or AMIs of most popular Linux and Windows flavors that you can spin up in your environment. Use current operating systems and hardened images in each case. These can be created and customized as per your specific requirements or acquired through one of several specialized vendors in the AWS Marketplace.

For the most part, EC2 and RDS instances do not need to be directly exposed on the Internet and should be spun up in private subnets. Even web servers do not need public IP addresses if they are set up in a farm, fronted by an AWS Application Load Balancer (ALB); the ALB 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. It is critical that all data drives or volumes attached to these instances be encrypted with multi-regional KMS keys that are set to rotate annually. Likewise, all services and endpoints need to be configured for SSL/TLS and HTTPS communications. The instances themselves can be protected via 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 smaller teams may find it easier to share keys between similar groups or clusters of servers. Ensure that users have their own set of SSH keys that can be added to the instances. This would be a good time to consider using a secrets management tool, such as AWS's Secrets Manager or HashiCorp's Vault, that can be integrated into DevOps processes as well as the application.

Beyond these external defenses, systems need to be hardened on the inside as well. If you create a private set of secure AMIs to be used across the environment, note that these need to be maintained and patched on a regular basis—a substantial lift requiring a larger team with the resources to update every couple of weeks. Alternately, you can implement pre-hardened AMIs provided by organizations such as CIS or VMware Bitnami through the AWS Marketplace. The AWS Marketplace is a great resource to find and deploy AMIs and cloud-based solutions from most popular security vendors such as Barracuda, Trend Micro and Cisco. The deployments for all such solutions should be automated through tools such as Ansible or the AWS Systems Manager. When doing so, do not forget to log everything you can from ALB access logs down to systems and applications logs into S3 buckets or other tool of your choice, from where it can be actively monitored.

Working with Containerized Environments

For containerized environments built around ECS/EKS you'll want to implement many of the same security concepts and constructs such as IAM Roles, Security Groups, Secrets Manager, and TLS endpoints.

AWS's Fargate can be used to shift the additional responsibility of the underlying instance and runtime back to AWS.

As a managed PaaS offering however, Fargate provides less control and transparency, and may turn out to be more expensive in the long run despite claims of being a cost-effective serverless solution, so these factors need to be considered based on the environment and workload.

Here are 6 additional considerations for securing Kubernetes environments:

- Create minimal images from scratch.
- Enforce pod security through Policy-As-Code (PAC) or Pod Security Standards (PSS).
- Implement a service mesh such as Istio or Linkerd in the architecture.
- Use Helm to automate deployments.
- Adopt "shift left" principles with code and CICD pipelines.
- Periodically run kube-bench to verify compliance with CIS benchmarks for Kubernetes.

Logging, Monitoring, Alerting

Even experienced IT professionals drop their guard from time to time, and 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 any unauthorized access attempt, data breach, or systems anomaly. Some of the must-have monitoring tools provided by AWS include:

CloudWatch: For monitoring instance availability, systems parameters and service logs.

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.

There are several other time-tested open-source tools that can be used for monitoring services such as Prometheus, Grafana, Nagios, or commercial tools such as Datadog or Dynatrace. Alerting can be set up through an external monitoring facility such as Pingdom, which can test the user experience and collect metrics on any public facing services. We also recommend using a Cloud Management Platform (CMP) such as NetApp's CloudCheckr or VMware's CloudHealth to provide real-time dashboards for cost and compliance reporting.

Along with systems monitoring, it is critical that service and application logs be collected and monitored for signs of forced entry or other malicious activity. You can set up a centralized logging facility with AWS CloudWatch Logs, Elastic Stack or Splunk to analyze your logs; alternately, if you don't want to run a 24x7 shop then consider utilizing a service like CrowdStrike to watch over your instances. Tie in alerting and incident reporting with PagerDuty or Atlassian's OpsGenie to set up on-call rotations and escalations.

Other monitoring and reporting resources from AWS include:

- AWS Audit Manager, to continuously audit your AWS usage and simplify how you assess risk and compliance with regulations and industry standards.
- Amazon GuardDuty, to protect your AWS accounts and workloads with intelligent threat detection and continuous monitoring.
- Amazon Artifact, to automate compliance reporting and provide on-demand access to more than 2,500 security controls and events.

Tools such as these (including the CMP tools mentioned above) allow companies to establish compliance auditing capabilities, that in turn allow security analysts to examine detailed activity logs or reports, and see who had access, IP address entry, what data was accessed, etc.
Finally, 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 CrowdStrike can potentially help.

Backups & Disaster Recovery

Backups is one of the most mundane, yet one of the most important, functions in the safety and security of your environment, and along with Disaster Recovery is a critical component of any regulatory compliance requirement. 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 such as AWS's Elastic DR.

In Short: 7 Ways to Improve Your Security Posture

Creating a secure and compliant AWS environment needs to combine a multi-layer approach with a variety of building blocks and controls available in AWS's ecosystem. Here's a recap:

1. Control access to the AWS console: Use SSO or IDMS, with MFA. Enforce password policies, and different roles & policies for the creation & deletion of resources.
2. 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.
3. 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 trusted users on known hosts. Use encryption everywhere you can – both for data at rest (S3 buckets, EBS and EFS volumes) and data in motion (using TLS endpoints).
4. Standardize your builds: Use automated tools like Terraform combined with configuration management tools like Ansible, Helm, etc. to build and maintain your environment.
5. Adopt a holistic security posture: Use best-of-breed third-party security and management tools available in AWS's ecosystem.
6. Monitor your environment: Use AWS constructs like CloudTrail, Config, Trusted Advisory Reports, and Billing Alarms in addition to other CMP, logging and monitoring tools.
7. Backup your environment: Use snapshots. Store a copy off-region or in a separate account.

Resources:

- AWS: Cloud Security Features
- Akamai: Cloud Optimization Solutions
- Ansible: Automation Solutions for AWS
- Atlassian OpsGenie: Alert Management Services
- Barracuda Networks: Web Application Firewall on AWS
- Cisco: Cisco Products on AWS Marketplace
- CloudFlare: Security Solutions
- CloudStrike: CrowdStrike Falcon for AWS
- 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
- Istio: Service Mesh for Kubernetes
- Linkerd: Service Mesh for Kubernetes
- Nagios: Monitoring Products
- Okta: Identity Platform
- OpenVPN Technologies: VPN Products on AWS Marketplace
- PagerDuty: Alert Management Services
- Palo Alto Networks: Next Gen Firewall on AWS Marketplace
- Pingdom: Monitoring Services
- Splunk: Data Analytics Products on AWS Marketplace
- Trend Micro: Security Products on AWS Marketplace

Tgix is a certified Advanced Consulting Partner with AWS and has worked with many organizations, on their regulatory compliance issues, secure cloud architectures, and support. Contact us for a complimentary evaluation of your current network and security architecture in AWS.

Address: One Bridge Plaza North, Suite 275, Fort Lee, NJ 07024
Web: tgix.com
E-mail: info@tgix.com