Security Best Practices for Amazon Web Services

Version 1.2
Released: January 12, 2015

Securosis, L.L.C. 515 E. Carefree Highway Suite #766 Phoenix, AZ 85085 **T** 602-412-3051
info@securosis.com www.securosis.com

Table of Contents

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog but has been enhanced and professionally edited.

Special thanks to Chris Pepper for editing and content support.

Copyright

Building on a Secure Foundation

Amazon Web Services is one of the most secure public cloud platforms available, with deep datacenter security and many user-accessible security features. Building your own secure services on AWS requires properly using what AWS offers, and adding additional controls to fill the gaps.

Amazon's datacenter security is extensive — better than many organizations achieve within their internal datacenters. Do your homework, but unless you have special requirements you can feel comfortable with AWS's physical, network, server, and services security. AWS data centers currently hold over a dozen security and compliance certifications, including SOC 1/2/3, PCI-DSS, HIPAA, FedRAMP, ISO 27001, and ISO 9001.

Never forget that *you* are still responsible for everything you deploy on top of AWS, and for properly configuring AWS security features. AWS is fundamentally different from a virtual datacenter (private cloud), and understanding these differences is key for effective cloud security. This paper covers the foundational best practices to get you started and help focus your efforts, but these are just the beginning of a comprehensive cloud security strategy.

Defend the Management Plane

One of the biggest risks in cloud computing is an attacker gaining access to the cloud *management plane*: the web interface and APIs that configure and control your cloud. Fail to lock down this access and you might as well just hand over your datacenter to the bad guys.

Fortunately Amazon provides an extensive suite of capabilities to protect the management plane at multiple levels, including both preventative and monitoring controls. Unfortunately the best way to integrate these into existing security operations isn't always clear; it can also be difficult to identify any gaps. Here are our start-to-finish recommendations.

Control access and compartmentalize

1. The most important step is to enable *Multi-factor Authentication (MFA)* for your root account. We recommend using a hardware token for root accounts, which is physically secured in a known location and only accessible for key administrators to access in case of emergency.
2. Configure your *Security Challenge Questions* with random answers not specific to any individual. Write down the answers and store them in a secure but accessible location.
3. Create separate administrator accounts using Amazon's Identity and Access Management (IAM) for super-admins, and also turn on MFA for each of those accounts. These are the admin accounts being used on a day to day basis, saving your root account only for emergencies.
4. Create separate AWS accounts for development, testing, and production, and other cases where you need separation of duties. Then tie the accounts together using Amazon's *consolidated billing*.

Locking down your root account means you always keep control of your AWS management, even when an administrator account gets compromised. Using MFA on all administrator accounts means you won't be compromised even if an attacker manages to steal a password. Additionally, leveraging different AWS accounts for different environments and projects compartmentalizes risks while supporting cross-account access, but only when necessary.

Amazon's IAM policies are incredibly granular, down to individual API calls. They also support basic logic, such as tying a policy to resources with a particular tag. Be aware that these policies can get complicated quickly, so aside from 'super-admin' accounts use these IAM best practices:

First 5 Minutes

For new AWS accounts, do the following within the first 5 minutes:

- Enable root account MFA
- Disable root Access Keys
- Configure the Security Challenge Questions
- Create separate administrator roles

- Use the concept of least privilege and assign different credentials based on job role or function. Even if someone needs full administrative access sometimes, they should have entitlements based on what they do day to day.
- Use *IAM Roles* when connecting instances and other AWS components together. This establishes temporary credentials which AWS rotates automatically.
- Also use roles for *cross account access*. This allows a user or service in one AWS account to access resources in another, without having to create another account, and ties access to those policies.
- Apply object-level restrictions using IAM policies with tags. Tag objects properly and the assigned IAM policies for those tags are automatically enforced.
- Use different accounts and credentials for administrative functions within each AWS region and service.
- Integrate your internal directory service with AWS using *SAML 2.0* for single sign-on, if possible. But be careful; this is most suitable for environments that don't need deep access to AWS resources, as this eliminates the ability to compartmentalize access using different accounts and credentials.
- Never embed Access Keys and Secret Keys in application code. Use IAM Roles, the *Security Token Service*, and other tools to eliminate static credentials. Many attackers are now scanning the Internet for credentials embedded in applications, virtual images, and even posted on code-sharing sites.

These are only a starting point, focused on root and key administrator accounts. Using MFA on these accounts is your best defense against most management plane attacks.

Monitor activity

Amazon provides three tools to monitor management activity within AWS. Enable all of them:

- **CloudTrail** logs all management (API) activity on AWS services, including Amazon's own connections to your assets. Where available it provides complete transparency for both your organization's and Amazon's access.
- **CloudWatch** monitors the performance and utilization of your AWS assets, and ties tightly into billing. Set billing alarms to detect unusually high levels of activity. You can also send system logs to CloudWatch but this isn't recommended as a security control.
- **Config** is a new service that discovers services and configurations within running instances, and tracks changes over time. It is a much cleaner way to track configuration activity than CloudTrail.

CloudTrail and Config don't cover all regions and services, so understand where the gaps are at this point in time. As of this writing (January 2015) Config is still in preview, with minimal coverage, but both services will be extended both from a

capabilities and global footprint over time. These features provide important data feeds, and will provide critical data for existing security data collection and analysis functions, including log management and SIEM.

As a next step, many organizations use a management portal (open source or commercial) to lock down the management console, instead of allowing direct access to AWS. This gives much tighter control over access and monitoring. This allows you to proxy administrator access with *Privileged User Management*, "jump boxes" or similar tools to ensure that only authorized parties have access to the console and activity is monitored.

Implement Built-in AWS Infrastructure Security Features

Once you lock down and establish monitoring for your Amazon Web Services management plane, move on to protecting the virtual infrastructure. Start with these tools that Amazon provides:

Use Security Groups and VPCs for network defense

AWS uses a proprietary *Software Defined Network* that provides more security than physical networks. All new accounts on AWS use *Virtual Private Clouds* for underlying networking, giving you extensive control over network configurations allowing you to run dozens or hundreds of separate virtual networks. *Security Groups* combine features of network and host firewalls to enforce network access control. They apply to groups of instances like a network firewall, but protect instances from each other like a host firewall. Security Groups are the basis of AWS network security:

- By default, instances in the same security group can't talk to each other. This prevents attackers from moving laterally within your cloud environment.
- Separate application components across security groups, with only required ports open between them.
- External administrative access (ssh or RDP) should be restricted to the IP addresses and subnets used by your administrators.
- Minimize the number of public subnets, and use NAT gateways to connect private subnets to the Internet as needed, just like you do in existing enterprise networks.
- Establish *Access Control Lists* to isolate subnets. ACLs are not a substitute for security groups, but a complementary tool.
- Require administrators to connect through a VPN or SSH "jump box" before connecting to instances. This can be implemented using an existing *Privileged User Management* tool.

Defend hosts and data

AWS is a mixture of Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). Amazon bears most responsibility for keeping back-end components secure, but you are still responsible for properly configuring each service and your own instances. AWS IAM policies are your main tool for defense, although Amazon does offer additional features which can help you secure instances and protect data.

- Establish an incident response process to handle compromised instances and other AWS services under attack.
- Use the AWS API or command line to collect all metadata, snapshot storage volumes, quarantine with IAM, and quarantine network connections.

Amazon bears most responsibility for keeping back-end components secure, but you are still responsible for properly configuring each service and your own instances.

- Design applications to use *Autoscaling Groups*. Instead of patching running or compromised servers, you can terminate them and replace older instances with clean up-to-date copies without incurring any downtime.
- AWS supports encryption for several data storage services — including S3, EBS, and RedShift. You can manage the keys yourself with their *Key Management Service* (located in the IAM console).
- Amazon can access keys via the Key Management Service, which may present a security risk. If you need extra security consider using CloudHSM instead to store the keys, although integration with AWS isn't as simple.
- If you use *CloudHSM,* make sure you have at least two redundant instances so you don't lose your keys. Amazon cannot view or recover the keys.

Finish with Additional Security Tools

AWS provides an excellent security foundation, but most deployments require a common set of additional tools:

- Amazon's monitoring tools (CloudTrail, CloudWatch, and Config) offer incomplete coverage, and no correlation or analysis. *Integrate their feeds into existing log management, SIEM, monitoring, and alerting tools* that natively support and correlate AWS logs and feeds, so they can fill gaps by tracking activity AWS currently misses.
- Use a host configuration management tool designed to work in the cloud to automatically configure and update instances.
    - Embed agents into approved AMIs or bootstrap through installation scripts to ensure full coverage.
    - Insert baseline security policies so all instances meet security configuration requirements. This is also a good way to insert other security agents.
- Enhance host security using tools and packages designed to work in highly dynamic cloud deployments:
    - Agents should be lightweight, communicate with the AWS metadata service for important information, and configure themselves on installation.
    - Host Integrity Monitoring can detect unauthorized changes to instances.
    - Logging to collect local audit activity for each instance and setting alerts on policy violations.
    - Host firewalls fill gaps left by security group limitations, such as rule set sizes.
    - Some tools can additionally secure administrator access to hosts without relying solely on ssh keys.
- For web applications use a cloud-based Web Application Firewall.
- Some services also provide DDoS protection. Although AWS can support high levels of traffic, DDoS protection stops traffic before it hits your instances… and your AWS bill.
- Choose security assessments and scanning tools that tie *directly* into AWS APIs and comply with Amazon's scanning requirements.
    - Look for tools that not only scan instances, but can assess the AWS environment.

Where to Go from Here

These fundamentals are just the surface of what is possible with cloud security. Explore advanced techniques like Software Defined Security, DevOps integration, and secure cloud architectures.

Who We Are

About the Author

Rich Mogull, Analyst and CEO

Rich has twenty years of experience in information security, physical security, and risk management. He specializes in data security, application security, emerging security technologies, and security management. Prior to founding Securosis, Rich was a Research Vice President at Gartner on the security team where he also served as research co-chair for the Gartner Security Summit. Prior to his seven years at Gartner, Rich worked as an independent consultant, web application developer, software development manager at the University of Colorado, and systems and network administrator. Rich is the Security Editor of TidBITS, a monthly columnist for Dark Reading, and a frequent contributor to publications ranging from Information Security Magazine to Macworld. He is a frequent industry speaker at events including the RSA Security Conference and DefCon, and has spoken on every continent except Antarctica (where he's happy to speak for free — assuming travel is covered).

About Securosis

Securosis, LLC is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services.

We provide services in four main areas:

- **Publishing and speaking:** Including independent objective white papers, webcasts, and in-person presentations.
- **Strategic consulting for end users:** Including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessments.
- **Strategic advisory for vendors:** Including market and product analysis and strategy, technology guidance, product evaluations, and merger and acquisition assessments.
- **Investor due diligence:** Technical due diligence including product and market evaluations, available in conjunction with deep product assessments with our research partners.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors.

Securosis has partnered with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis.