Advanced Security at Scale in the Public Cloud

Introduction

Security is the critical enabler of application and service delivery in cloud environments. For organizations, contemplating the migration of essential activities to the cloud and the ability to align security with workloads at scale is a key business consideration. The auto-scaling process helps spin up and down cloud security appliances without user intervention to ensure the highest availability of resources and consistent application of security profiles across premises.

Auto-scaling integration helps ensure cloud security is automatically applied/removed as VMs spin up/down to ensure consistent security profiles across premises and high availability of cloud workload lifecycles.

Scale Security in Public Cloud

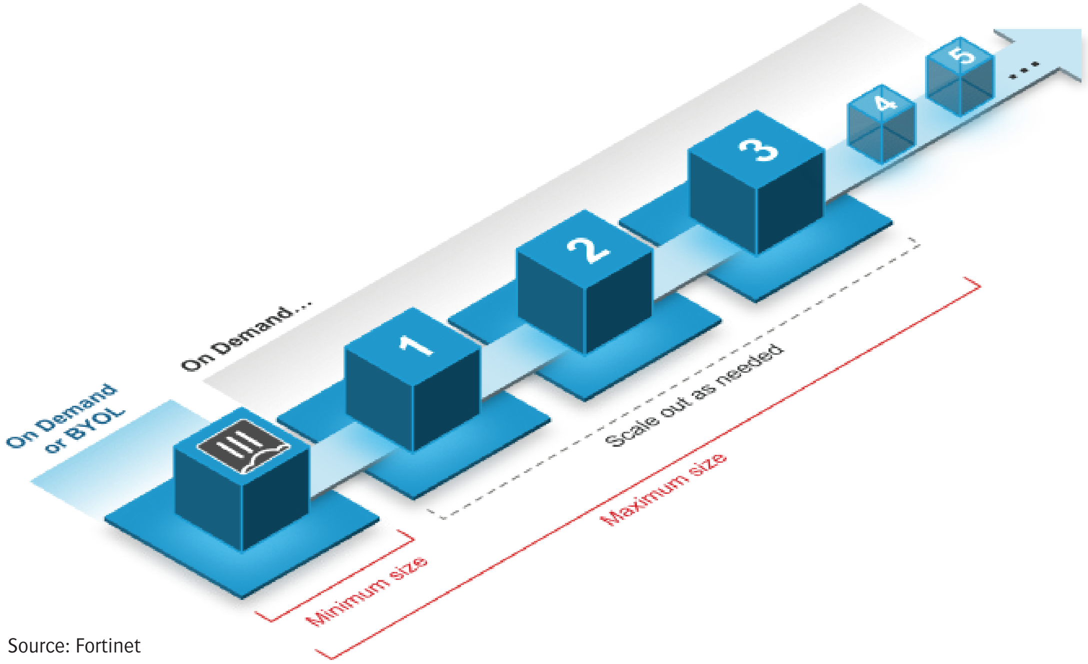
Whether the driving reason is flexibility, dynamic deployment, strictly one of cost, or other, the move to hybrid and public cloud services is well underway. Security, for both the elements and services living in the cloud as well as data inbound and outbound has remained a primary and lingering limiter for these migrations. What good is rapid scale and mobility if it doesn't meet or exceed the current standards now expected in physical deployments?

The leading public cloud providers are moving to secure the infrastructure and assure customers that a shared virtual machine is as safe as it would be in their own datacenter. However, the shared responsibility challenge of securing cross-premise traffic to and from those new networks and between those instances has remained.

The standard cloud security concerns are:

- Securing communications between VM instances
- Securing inbound communications from internet
- Securing cross communications among Enterprise Public Cloud subscriptions
- Securing communications over WAN

DIAGRAM A



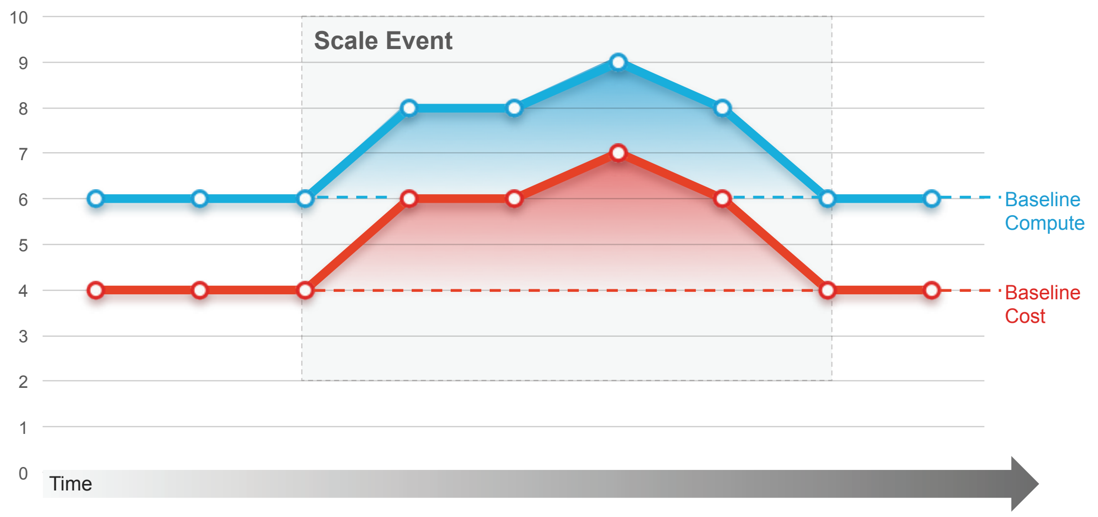
Source: Fortinet

There are two essential challenges for securing cloud infrastructure: scale and elasticity. Organizations are becoming quickly addicted to the idea that additional compute resources are just a few mouse clicks away. The early industry leaders in the space, namely Amazon Web Services (AWS) and Microsoft Azure, have made instantiation and provisioning of storage and compute nearly seamless and by an exponential factor less painful than traditional methods of deployment. As such, rapid scale is easily achieved, but this new frontier comes coupled with a sacrifice of hardware offloading. This singular concept, that has accelerated network growth in the past two decades has more recently become mature and a necessary component in the network security space. Furthermore, the rapid growth and contraction of services is one of the biggest draws for this new configuration. With security devices deployed in the cloud that lack the custom ASICs that have allowed them to keep up with the pace of network expansion in the traditional network, can they be made to scale? In short, yes.

Fortinet has been a pioneer in virtualized network security solutions and has continued that pace with product implementations in the top cloud providers. In parallel, dynamic horizontal scale had been a top priority. Using standard service-monitoring tools in the providers' environment as a platform, service or even simple execution, Fortinet can trigger a scaling event to add and subsequently remove security assets. Examples of this event ranges in the initial form as metrics such as concurrent session count, throughput, CPU or memory usage, but has no limit in its implementation. This event triggers a new device instance with a cloned configuration and adds it to the load-balancing cluster so that outbound connections can be then spread to multiple instances.

See Diagram A to illustrate our example steady state with our illustration of fixed cost in Diagram B.

DIAGRAM B



Source: Fortinet

Now that we have scale up, allowing for a consistent baseline cost and only bursting up during busy times such as a major sporting match, software launch, development push, or even storms and natural disasters that might render other service points offline, what about the return of those compute resources to our baseline? Using the same monitoring and timing mechanisms, or different, if we choose, we can turn off the dynamic instantiations, just leaving on the base instance(s). See Diagram B for an example of the return to a fixed operational cost once the scale event has subsided. As a bonus, with these providers, the oldest machine is shut down first, so the latest images are running on the providers' latest hardware.

Summary

Fortinet can enable your cloud security to grow with your business at scale never-before imagined with security elasticity leveraging Auto Scaling. This capability enables dynamic application security which scales in conjunction with your cloud compute resources automatically according to conditions you define.

Source: Fortinet

Research from Gartner:

Best Practices for Securing Workloads in Amazon Web Services

Amazon Web Services is the most widely adopted cloud infrastructure-as-a-service provider. Here, we provide security professionals with best practices for the secure deployment of workloads in AWS, many of which apply to securing workloads in any IaaS provider.

Key Challenges

- You can't protect what you can't see. Enterprise IT may not be aware of cloud workloads, especially those created by users directly, making protection impossible.
- The shared-responsibility model for cloud services means that enterprises will need to adapt their traditional security models to secure cloud-based workloads.
- Network security remains important for securing workloads, but Amazon Web Services (AWS) owns the physical network and won't allow placement of physical network security appliances in its data centers.
- A greater emphasis on guest instance workload-based security is needed, but most legacy host security solutions weren't designed to support the protection of cloud-based workloads.

Recommendations

- Get visibility of enterprise AWS usage, and monitor AWS workloads by activating AWS CloudTrail and Amazon CloudWatch; bring these logs to log management or security information and event management (SIEM) systems.
- Start with the foundation — proper identity and access management (IAM) administration, especially for administrative and API-based access within AWS.
- Change your mindset to have more emphasis on workload security — rather than creating secure networks of instances, create networks of secure instances — and adopt a "no patch" philosophy.
- Pressure incumbent host and network security solution providers to support deployment of their software in cloud environments, with explicit support for AWS and with full API enablement.

Strategic Planning Assumption

Through 2020, 80% of cloud breaches will be due to customer misconfiguration, mismanaged credentials or insider theft, not cloud provider vulnerabilities.

Introduction

AWS is the most widely deployed public cloud infrastructure as a service (IaaS) solution in the world and is a leader in Gartner's Magic Quadrant for IaaS. AWS offers a large number of built-in security capabilities (see Note 1), and questions on the proper practices for securing workloads in AWS are increasing.[1]

AWS is a not a "consumer grade" IaaS cloud. It is a market leader, with a portfolio of security capabilities and security ecosystem partners unmatched by other IaaS providers. However, simply moving existing workloads to AWS without rethinking security tools, processes and system management will result in workloads that are less secure than they were when located within enterprise data centers. Conversely, a properly managed and secured workload in AWS will be at least as — and, in most cases, more — secure than in an enterprise data center.

To securely deploy workloads in AWS, changes in enterprise security will be needed. Here, we provide the best practices for securely deploying enterprise workloads in AWS.[2] In addition, most of these best practices will apply to securing workloads in other IaaS providers.

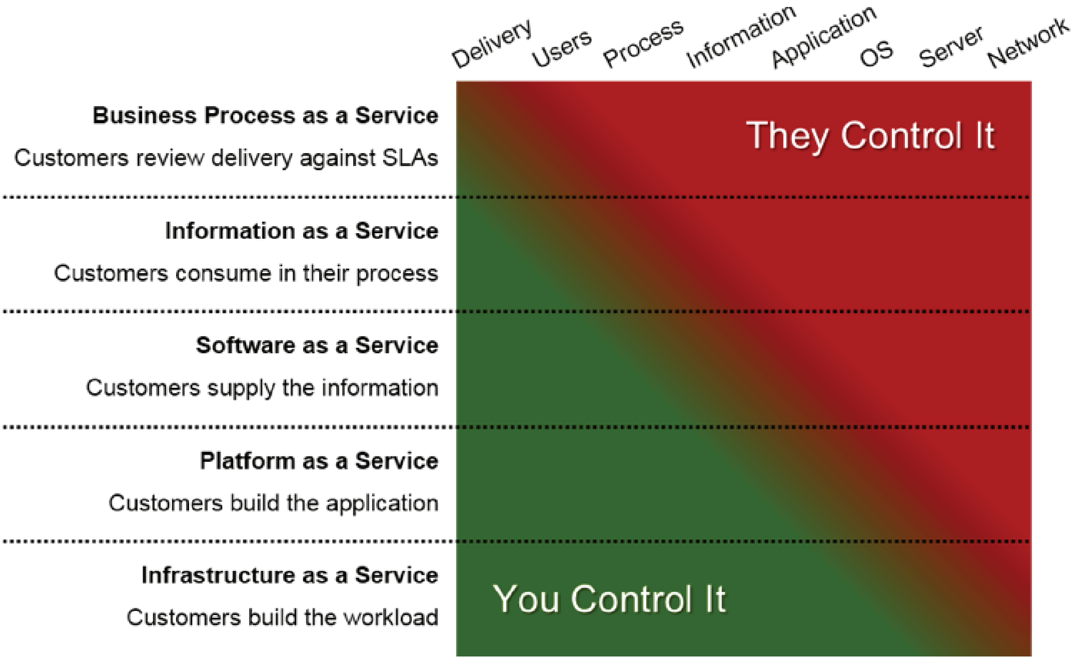
Analysis

Understand Your Responsibilities in the Shared-Responsibility Model

Within any cloud-based service, there is a shared responsibility for security between the cloud provider and the customer. For IaaS, including AWS, the cloud provider uses and secures its own data centers, network, system hardware and typically the virtualization layer, while the enterprise is responsible for the security of the OS and application code of the image that runs on the underlying infrastructure (see the bottom layer of Figure 1). Because the enterprise builds the workload, there will be an increased emphasis on host configuration and security with cloud IaaS-based workloads.

Even within IaaS providers, there are shades of gray. For example, with a higher-touch managed cloud IaaS, the cloud provider might take responsibility for the patching of images. Also, even though the IaaS provider owns the physical network, you are likely responsible for the correct logical configuration of network address translation and virtual LANs. AWS publishes its own guidance for its shared-responsibility model in "Amazon Web Services — Overview of Security Processes." Finally, the IaaS provider almost always maintains the virtualization/hypervisor layer, which represents risk if it has a vulnerability that is attacked.[3]

FIGURE 1 Typical Cloud Shared-Responsibility Model


Delivery
Users
Process
Information
Application
OS
Server
Network
Business Process as a Service
Customers review delivery against SLAs
Information as a Service
Customers consume in their process
Software as a Service
Customers supply the information
Platform as a Service
Customers build the application
Infrastructure as a Service
Customers build the workload
They Control It
You Control It

Source: Gartner (April 2015)

Get Visibility of Cloud-Based Workloads, and Activate CloudTrail

You can't secure what you can't see. To gain visibility into which departments might be using AWS directly without IT knowledge, you can analyze enterprise firewall logs and proxy logs in a tool such as Splunk to identify enterprise access to/from AWS. For visibility into running workloads and administrative access, AWS provides APIs to enumerate virtual machines (VMs)/services at any given point in time. Amazon CloudWatch can be used to gather workload OS and application logs. In addition, AWS CloudTrail (see Note 2) is an Amazon-unique capability that logs all administrative and API-based access and operations on workloads in AWS.

Specific best practices:

- Get visibility of AWS usage taking place outside the knowledge of IT using firewall and proxy logs or using one of the emerging cloud application discovery tools.[4]
- Seek an umbrella AWS enterprise agreement that applies to all customer accounts, and seek consolidated billing for visibility of spending across all accounts.
- Activate Amazon CloudWatch to log all OS and application logs, and activate AWS CloudTrail to log all API actions. This will provide visibility into all user actions at AWS and enables you to centralize all actionable data in a log management or SIEM system.
- Make visibility of cloud workloads a continuous process. End users may continue to try and procure directly, and workloads are transient, especially in bursty cloud-native applications. Use AWS APIs to gain visibility into enterprise workloads as they come and go.

Make Solid IAM Practices Your Foundation

At the heart of AWS's security architecture is AWS's IAM service. It's the conceptual equivalent of an enterprise directory — a repository of users, groups, roles, server roles and entitlements, providing granular access control on objects and actions within AWS. Since the enterprise is responsible for defining users and access in IaaS (see Figure 1), proper management is essential for maintaining identity-based workload isolation. Most of AWS's services are integrated into AWS's IAM service so that role-based access control (RBAC) can be configured to assign specific people and systems to specific roles. This enables organizations to set up and configure separation of duties as their compliance needs require.

Specific best practices:

- Require the use of AWS's IAM services on all AWS services.
- Make the person responsible for cloud architecture set guidance around IAM usage and roles with these considerations:
    - Don't force business unit users (such as developers and admins) into a single account structure, unless it makes sense.
    - Federate IAM where it makes sense, but delegate administration into the separate account structures.
        - For Mode 1-type workloads (see Note 3), link AWS's IAM to enterprise directories[5] for consistency in identity and group memberships.
        - For Mode 2-type workloads (see Note 3), rightsize the IAM strategy, and manage independently if enterprise IAM processes are too rigid.

  - Use AWS accounts to scope risk in a way that follows the concerns of the enterprise. For example, if production stability is a key issue, isolate nonproduction and production AWS accounts.

  - Use AWS account boundaries to control risk scopes with regard to audit, governance and risk requirements, if they exist (for example, PCI).

- Use groups for users. AWS supports hierarchical RBAC. Use AWS's RBAC to ease administration and management of administrative accounts.

- Use roles to group permissions for servers of similar configurations together logically (for example, "Web servers") and to ease administration of workload-specific policies.

  Apply the principle of least privilege for administrative access. Don't use "all powerful" administrative accounts as this represents unacceptable risk in AWS.[6]

  - Require multifactor authentication for administrative access to protect against credential theft.

  - Require strong authentication for sensitive operations, such as deleting a VM.

  - Use context-based policies, and restrict administrative activities based on location or time of day.

  - Extend enterprise system administrator privilege management tools and processes to cloud services.[7]

- Use the AWS Security Token Service (STS) to place restrictions on application-level access (such as time durations). As of March 2015, STS is available locally in all AWS regions worldwide.

- Create an AWS credential rotation policy to regularly rotate access tokens and secret keys to reduce the "blast radius" of a compromised credential.

- Avoid the use of Secure Shell (SSH) to perform administrative maintenance on images.

  - Watch out for embedded SSH keys in images where SSH is used.

  - Monitor for the creation of unauthorized SSH keys, and if located, ensure they are part of your credential life cycle.

- Issue an API key for every application and restrict its privileges, and architect to perform all administrative activities via APIs.

- Watch out for embedded credentials and API keys in AWS machine images (AMIs) (for example, when writing out AMIs, use roles), in scripts and especially if publishing a community image.

- Don't overlook the critical issue of privileged account deprovisioning (see Note 4).

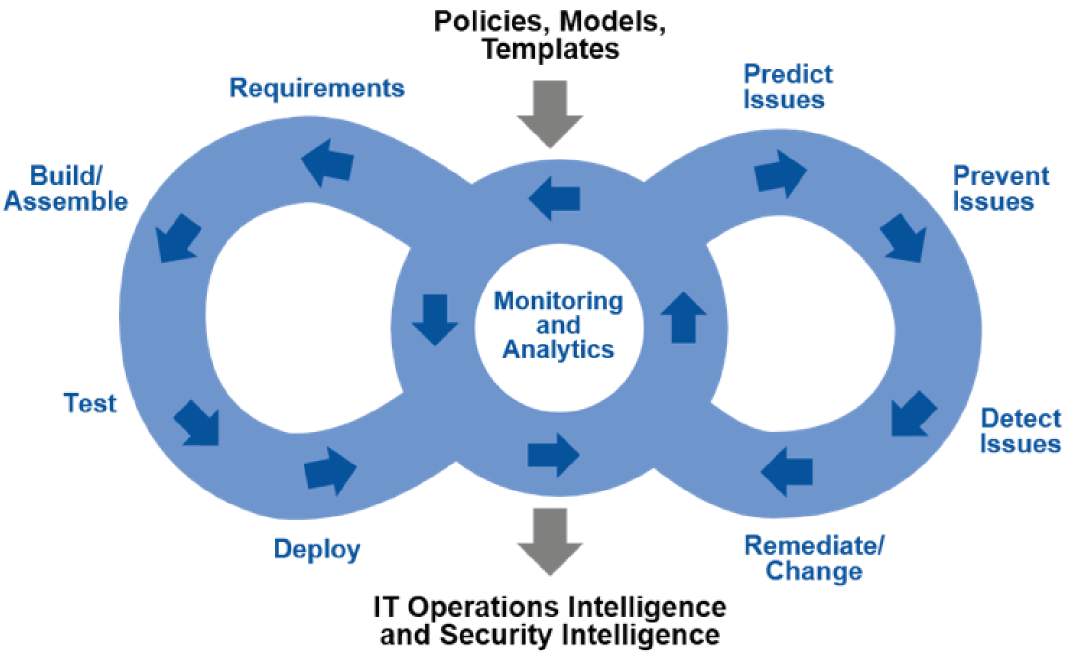
Bake Security Into the AWS Workload From Development

The heart of Elastic Compute Cloud (EC2) is the server workload. A server security approach rooted in strong configuration, patching and whitelisting (default deny) is much more effective than relying on a signature-based anti-malware approach (see Figure 2 and "How to Devise a Server Protection Strategy"). However, workloads in AWS are different from workloads in data centers. With cloud-native application designs (Mode 2; see Note 3), workload instances spin up quickly without human intervention and just as quickly disappear. The construct of these workload images and their policies (for example, network connectivity) is critical to get right. Security constructs, such as correct configuration, patched images and network connectivity, should be baked into the deployment process and image file automatically, without relying on configuration after deployment.

Over time, the security and operations mindset will switch to setting policies programmatically using APIs to configure security policy enforcement points based on policy. This will be achieved typically using scripts or automation tools, such as Chef, Puppet and others where the security policies are embedded and configured within scripts/blueprints/recipes that drive the configuration without requiring human intervention. This "need for speed" and shift in organizational thinking are captured within a movement referred to as secure "DevOps" — a cultural/mindset change that tears down the traditional walls within IT (development on the left side of Figure 2, and operations and security on the right side) to deliver faster IT-enabled capabilities for the business.

Whether you call this secure DevOps or simply the need for speed and IT agility, the result is the same. IT needs to be able to provision/deprovision workloads more quickly, and to do this, people, tools and processes must change. Security needs to be an integral part of this transformation — thus, the term "DevSecOps," which brings security into the center of this shift.

FIGURE 2 DevSecOps



Source: Gartner (April 2015)

Specific best practices:

- Harden AMIs. Use AWS Trusted Advisor to help.
- Harden the deployment scripts that layer things on top of the AMI (such as security) after deployment.
- Use AWS's predefined OS templates rather than creating your own from scratch, especially when using Linux and open-source software stacks.
- Architect to set security policy using scripts and APIs, not security consoles.
- Perform regular vulnerability and configuration scanning. Some vendors provide this capability directly to AWS-based workloads.

Adopt a "No Patch" Strategy Where Possible

In the shared-responsibility model of AWS and other IaaS providers (see Figure 1), the enterprise customer is responsible to ensure its workload images are patched. An emerging trend (especially in DevOps-type environments is to adopt a "no patch" strategy. Despite the name, this doesn't mean that enterprises don't keep their workloads patched. What this philosophy embraces is that "IT won't patch live systems." Specifically, the *templates* from which the workloads are derived are kept patched at all times. When a live system needs to be patched, it is replaced/regenerated from its underlying templates (in a DevOps environment, typically via automated scripts and toolchains).

Specific best practices:

- Use automation to build workload images without requiring human intervention.
- Use AWS's OS templates within these workloads to ensure guest images are up to date when deployed.
- Don't patch live systems when patches are needed:
  - Rebuild the workload using the most up-to-date AWS OS templates (which are kept patched by Amazon).
  - Patch the templates, not live systems, if you built your own OS templates. Regenerate the live systems using the underlying templates.
- Scan for vulnerabilities continuously. AWS may see this as an attack on its systems, so these need to be scheduled with AWS, or use a provider such as Qualys that integrates directly with AWS.
- Consider proactively implementing a systematic workload reprovisioning strategy, since any workload could be compromised at any point (see Note 5).

Encrypt All Network Traffic — Treat AWS as an Extension of Your Own Data Center

As enterprises start out with AWS, another best practice is the use of Amazon Virtual Private Cloud (VPC) combined with a VPN and network address translation (NAT) services to extend the enterprise's network address space to AWS workloads. There is nothing special about the Amazon VPC — it's essentially an encrypted VPN tunnel combined with NAT so that the AWS workloads look like an extension of the enterprise's own data center. Until December 2013, VPC was a separately charged item — it is now included with AWS EC2 as a standard capability.

There are several alternatives for the VPN tunnel. The most common are IPsec tunnels, and AWS supports termination of a number of major VPN provider's implementations (such as Cisco's and Juniper Network's). Many organizations go through a carrier-neutral colocation facility, such as Equinix, to cross-connect into AWS. Yet another alternative is the use of a Direct Connect,[8] wherein an enterprise can use dedicated Multiprotocol Label Switching (MPLS) circuits to directly connect to AWS data centers without requiring the traffic to flow on the Internet.

Specific best practices:

- Adopt Amazon VPC, and combine it with NAT to extend enterprise address space to your EC2 workloads.
  - Consider optional hardware-based IPsec at additional cost for better performance.
  - Consider connecting via a carrier-neutral collation facility or Direct Connect dedicated circuits for more predictable performance.

- Note that VPC provides isolated address space, but *not* dedicated compute, which is available at additional cost (but rarely required).

- Use VPC peering for redundancy (this is the default).

- Ensure all programmatic, API-based access to AWS is also encrypted using Transport Layer Security (TLS) or via SSH.

Use AWS Security Groups by Default, and Leverage a Third-Party Firewall for More Advanced Functionality

Solid security starts with isolation and segmentation. With all AWS services, AWS delivers its own basic, native stateful, Layer 3 firewalling *capability*. Note that the firewalling capability has not been certified by ICSA Labs. The firewall is configured in a "default deny" posture so that nothing can communicate in or out until it is explicitly permitted to do so. For ease in management, servers can be assigned to security groups. Security groups then have firewall policies that apply to all members of the group automatically when created.

For basic firewalling and segmentation functionality, AWS's firewalling capabilities via security groups are sufficient. Additional defense in depth may be configured using network access control lists between subnets.[9] However, if any complexity of policy or additional network-based security functionality (such as intrusion prevention or detection services, malware sandboxing, email, or secure Web gateway functionality) is needed, there is a broad selection of third-party security vendors available to deliver these capabilities (see Note 6).

We see two primary use cases:

- If the enterprise is protecting Type 1 workloads, then use a next-generation firewall in AWS, ideally the same as you use on-premises in the enterprise data center. Most enterprises want a single brand of firewall, as multiple consoles make all firewall operations more difficult and increase the chance of misconfiguration.

- If the enterprise is protecting Type 2 workloads, use security groups plus VPCs for isolation and segmentation with the logs brought back to enterprise log management systems. If more advanced capabilities are needed, use your on-premises vendor if possible — but only if its network security platform is fully automatable via APIs with a licensing model that makes sense.

Specific best practices:

- Use network isolation via AWS security groups as the first line of security protection by applying least privilege for network connectivity. Open only the absolute minimum level of connectivity necessary for workloads to function.
  - Use servers assigned to AWS security groups to assign policy.

- Audit and re-evaluate network connectivity regularly to ensure that a strong posture is maintained and expected isolation does not inadvertently degrade over time (see Note 7).

- Use an AWS-ready version of your current network security products if intrusion prevention system (IPS) or more capable firewalling is needed (see Note 6).
  - Consider the performance impacts of running network security products, including firewalls, in virtual platforms.
  - Evaluate the vendor's high-availability and fail-closed capability and the complexity of inserting the policy enforcement point in line.
  - Ensure that all necessary licensing and capacity are in place (since many network security products have restrictive licensing), especially if using highly elastic applications, to avoid the firewalls becoming a chokepoint or prohibitively expensive.

- Ensure the firewalling logs of any firewalling solution — AWS's or third parties' — are monitored.

- Don't assume you can get a network tap. Most IaaS cloud providers, including AWS, won't enable it.

- Evaluate and mature your existing network security architecture, security monitoring and change management processes so that a consistent set of processes (not necessarily technologies) are developed that span network security for on-premises and cloud-based workloads.

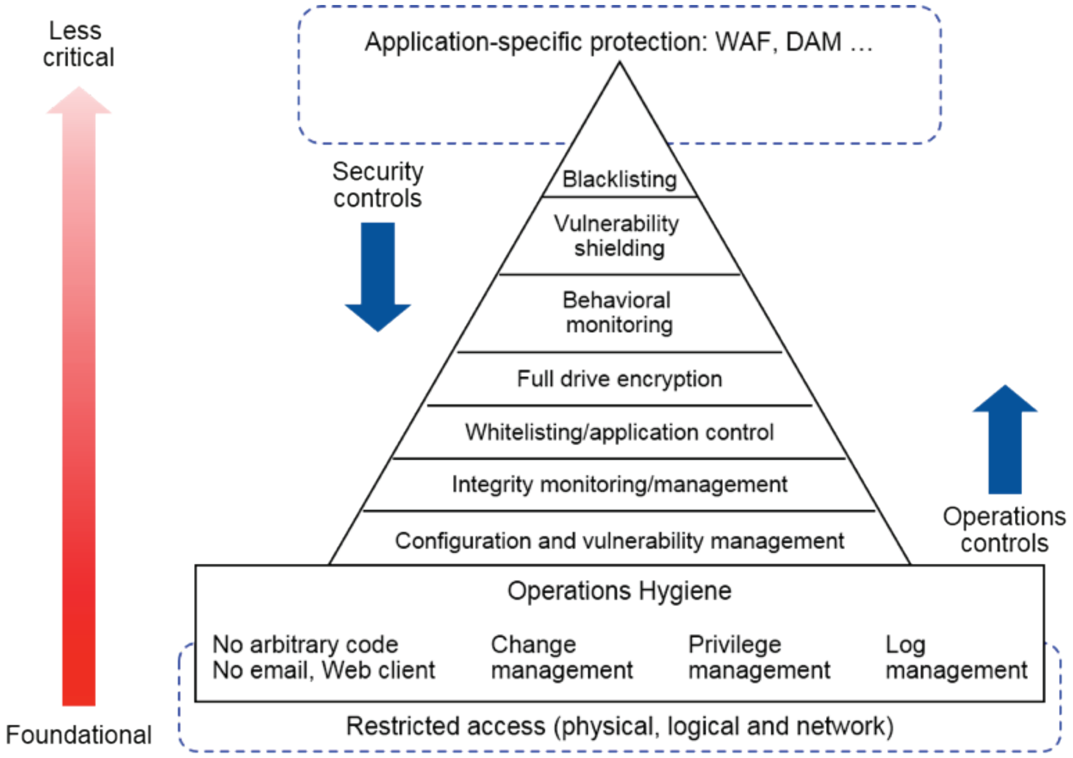
Adopt a Workload-Centric Security Strategy

More than with any other layer of cloud-based computing services, with IaaS, organizations have flexibility of security controls within their guest workload images (see Figure 3), such as a local firewall and host-based IPS in the form of agent-based software running in each guest instance. Network security is still important — and as we stated in the previous section, isolation is a critical first step — but proper guest workload security is more essential in AWS. Further, workload-based firewalls can fill gaps left by the use of AWS security groups for firewalling capability.

Although most server OSs provide basic whitelisting, leading application control solutions handle trusted change through digital signatures and trusted software distribution agents. As shown in Figure 3, there is minimal value of blacklisting — antivirus and anti-malware scanning — on a well-managed and secured server workload. Use application control as your primary control strategy.

Don't assume you can simply run on-premises agents in cloud-based guest instance workloads. The reality is they often won't work, or if they work, they perform poorly. Most vendors' agents assume they are running on a local LAN segment with contiguous address space and direct connectivity to the management console. In addition, the sheer scale of some cloud workloads or the fact that workloads may come and go in a short period of time will overload legacy management consoles. Ideally, agent-based software would be designed for the cloud from the beginning (see Note 8).

FIGURE 3 Cloud Workload Security Hierarchy



DAM = database activity monitoring; WAF = Web application firewall
Source: Gartner (April 2015)

Specific best practices:

- Replace signature-based anti-malware scanning with a whitelisting-based approach for all server workloads, except those that handle and host arbitrary files (such as a file server, FTP server and so forth).

- Use a model where the signature file can be updated on boot from a peer agent, a security controller or the Internet without requiring large file uploads from the enterprise management console on each reboot/rebuild if anti-malware scanning is needed (and in most cases, it is not).

- When evaluating host-based agents for AWS:

    - Ensure the agent can be managed using Internet-friendly and firewall-friendly protocols — typically RESTful- or JSON-based APIs carried over HTTP.

    - Ensure that, at a minimum, the vendor can support the ability of the agents to be managed from an on-premises server from a console located back in its data center.

        - Ideally, the vendor should support its management console running as a VM in AWS and be capable of supporting multitenancy for larger organizations that require it.

        - Ideally, the vendor should provide a multitenant "console in the cloud" or "console as a service" option so the enterprise doesn't need a separate management VM.

    - Ensure the agent and console architecture can scale elasticity up and down to support workloads that also scale elastically, potentially to tens of thousands of workloads.

    - Ensure the agent can bootstrap from a template and obtain its correct configuration centrally either via a logical tag in the agent or by supporting AWS's tagging infrastructure.

Encrypt All the Data the VM Stores Locally by Default

A straightforward way of protecting AWS workloads is to encrypt all the data the VM creates (in AWS, this is stored in Elastic Block Store [EBS]). Everything the VM creates after it is booted is stored encrypted. This protects the disk storage of the VM from outside snooping and theft, as well as providing assurance of the destruction of the EBS data when the key is destroyed. A well-designed architecture will allow the enterprise to store the key on its own premises or in AWS's hardware-based CloudHSM, thus providing protection from the IaaS administrator. Multiple vendors and solutions are appearing on the market for Windows- and Linux-based workloads in AWS using a variety of techniques, including AWS, which now provides this capability at no additional cost (see Note 9).

Specific best practices:

- When evaluating encryption solutions for AWS:

    - Ensure the vendor has certified its solution for AWS and is available as an AMI.

    - Require that support for customer-controlled and customer-supplied keys is kept on-premises, in AWS CloudHSM or alternatively in AWS Key Management Service.

    - Look for certification on other cloud providers if multicloud support is needed.

- Understand the limitations of encrypting VM data at rest. There are no "silver bullets" in information security. In-memory attacks, OS- and application-level attacks, credential theft, and VM snapshotting risks aren't addressed with EBS encryption.

    - Keep the workload on-premises or use a cloud provider with bare metal provisioning if the data is so sensitive that memory snapshotting is an unacceptable risk.

- Don't forget about protecting other types of cloud storage. For example, data written outside the VM won't be protected without additional configuration or application changes (for example, data written out by the application to other AWS services, such as Simple Storage Service [S3] or Relational Database Service [RDS]).

Don´t Overlook the Support Protection Infrastructure for the Application and Associated Network Infrastructure Services

Any application security strategy in AWS must extend to include the protection and availability of the associated set of application layer services that support the application. For example, if the application is a Web application, it may need to be protected by a Web application firewall (see Note 10).

Specific best practices:

- Consider security-as-a-service options outside AWS for network security services other than firewalling.
- Consider potential performance implications, as well as availability concerns, if a virtual appliance option running in AWS is chosen and if a single point of failure is created.
- Don't overlook application-supporting network infrastructure services, such as DNS, DHCP and NTP.
- Include application layer identity services, such as Active Directory (AD) and LDAP, in your protection strategy.
- Consider network and application layer distributed denial-of-service protection.

Pressure Your Incumbent Security Vendors to Support AWS

Not all third-party security vendors offer a software version of their capabilities in AWS, and just running security software in AWS is not always as simple as running legacy software in a VM. Ideally, the vendor has built its solution for the cloud from the ground up, so there are no issues with performance and scalability when running in AWS. For deployments that will span traditional enterprise data centers and AWS, hybrid deployment models should be supported. Here, the management console should treat on-premises and cloud-based workloads the same.

Specific best practices:

- When evaluating third-party solutions for AWS:
  - Require that security vendors certify their offering for AWS and make their solutions available as preconfigured AMI images.
  - Ensure all security functionality is fully API-enabled (for example, RESTful), supporting automating through scripts/recipes/blueprint systems such as Chef and Puppet. The vendor's security console should not be assumed to be the primary way that you will be provision security services — it will likely be API-based. Ideally, the vendor's console is built entirely on its own APIs.
  - Require that the vendor's solution natively understand and support AWS's logical tagging, groups and identity of VMs for the application of policy.
  - Require that the vendor integrate with AWS's native APIs so that the vendor's console can quickly identify new cloud workloads that don't have proper security protection.
  - Require Internet-addressable management. Don't assume VPC/VPN and address space continuity.
  - Request that the vendor provide support and visibility across cloud providers, not just AWS, if pursuing a multicloud strategy.
  - Request integration with and understanding of AWS security groups and VPC logical configurations.
  - Request that the vendor provide a multitenant "console in the cloud" option in addition to an AMI image.
  - Demand licensing models that make sense for cloud: "Per VM" doesn't always work, especially with bursty cloud workloads; "per VM hour" based on image size makes more sense.
  - Start voting with your wallet to force security vendors to support cloud environments natively with security software, without requiring physical appliances and without a compromise in functionality.

Evidence

[1] This is based on hundreds of client inquiries on AWS security during 2013 and 2014.

[2] AWS also provides security best practices (see "Tips for Securing Your EC2 Instance," Amazon Web Services).

[3] AWS's hypervisor is based on Xen. A Xen vulnerability in 2014 exposed workloads (see "The Xen Vulnerability That Rebooted the Public Cloud," eWeek). Another occurred in 2015 (see "Amazon Reboots Some Cloud Servers for a Xen Vulnerability ... Again," Network World). In general, AWS was less affected than some IaaS providers, but the incident demonstrated that these hypervisor-layer vulnerabilities do exist and can have serious consequences for cloud workloads.

[4] Gartner's "Hype Cycle for Cloud Security, 2014" includes a technology profile for cloud application discovery solutions. Vendors such as SkyHigh Networks, CipherCloud, Netskope, Elastica, Skyfence and Bitglass offer this capability, as do OpenDNS, McAfee and others.

[5] See AWS Directory Service.

[6] The unacceptable risk of all-powerful AWS administrators was demonstrated in the 2014 attack on Code Spaces (see "Hacker Puts Hosting Service Code Spaces Out of Business," Threatpost). Administrative credential theft resulted in the deletion of all Code Spaces' AWS machine images and resulted in the company going out of business. Separation of duties and multifactor authentication would have mitigated this theft. The root cause of the attack was poor IAM hygiene.

[7] See Next Generation Protection for AWS. For more detail on shared account password management tools, see "Market Guide for Privileged Account Management."

[8] See AWS Direct Connect.

[9] See "Security in Your VPC," Amazon Web Services.

[10] AWS security capabilities most requested by customers that AWS doesn't yet deliver are as follows:

- CloudTrail needs to audit *all* activities, including those of AWS administrators.
- CloudTrail needs to digitally sign logs for high assurance against tampering.
- EBS encryption is great, but it needs to integrate with CloudHSM and encrypt the boot volume (not just data volumes).
- On any encryption alternative, AWS needs to provide cryptographic agility — specifically, the ability to change out algorithms and key strengths if a weakness in a given algorithm is discovered.
- AWS's network configuration capability needs easier ability to insert security gateway services in traffic flows; for example, routing traffic to an in-line network-based intrusion prevention appliance.
- AWS needs the ability to restrict S3 nodes for access only within an enterprise's VPC from a network connectivity and isolation perspective.
- AWS provides no standardized way to achieve a network tap.
- AWS provides no direct SDN support (yet) exposed to customers (although AWS uses its own implementation behind the scenes extensively).
- It needs an OpenStack integration roadmap for seamless hybrid orchestration.
- It needs local full-drive (DAS) encryption for Hadoop worker nodes in Amazon's Elastic MapReduce offering.
- It needs a program for sharing attack data on AWS infrastructure for organizations that want this visibility. AWS's own security infrastructure doesn't have a way of sharing this visibility in a multitenant fashion across its customers.
- It needs a dedicated storage option. However, as a compensating control, an enterprise can encrypt the data being stored to achieve cryptographic isolation.
- It needs a bare-metal provisioning option (this is an infrequent requirement, but it is important to some enterprises requiring the equivalent of air-gapped isolation with no hypervisor in the middle).
- Unauthorized VM snapshotting by AWS administrators remains an area of customer concern.

Note 1
AWS Sets a High Bar for Security

- Secure access via Secure Sockets Layer (SSL)
- Built-in firewalling
- AWS IAM
- Multifactor authentication
- Private subnets
- Dedicated compute hardware option
- Encrypted data storage, including EBS
- Dedicated connection option
- Isolated GovCloud
- CloudHSM — dedicated, hardware-based cryptographic key storage option
- Amazon Key Management Service — a cloud-based, scalable key management infrastructure
- AWS Directory — an AD-compatible directory service
- AWS Trusted Advisor

Note 2
CloudWatch and CloudTrail

CloudWatch and CloudTrail may be activated at no additional cost, except for storage of the logs — typically stored in S3. These logs should be gathered and linked into enterprise log management systems and SIEM systems. However, the size of the logs may overwhelm existing tools not designed for the volume of data created.

Note 3
Bimodal IT

A full discussion of bimodal IT is outside the scope of this research. Mode 1 workloads are more traditional, and reliability is key. Mode 2 workloads are less predictable and are designed for flexibility and speed. The implications on cloud infrastructure are discussed in "Best Practices for Planning a Cloud Infrastructure-as-a-Service Strategy — Bimodal IT, Not Hybrid Infrastructure."

Note 4
Account Deprovisioning

In an enterprise data center, when administrators leave, they likely no longer have physical access to servers, which restricts their ability to log in, even if their credentials remained intact. In the cloud, the workloads are likely available from anywhere, making account revocation (and removal of the other factors in multifactor authentication, such as returning a timed token) critical. In all cases, privileged account management deprovisioning is a critical process that must be extended to the cloud.

Note 5
Systematic Workload Reprovisioning

More advanced application architectures will take the concept of "no patch" further — with systematic workload reprovisioning (SWR) (see "Systematic Workload Reprovisioning as a Strategy to Counter Advanced Persistent Threats: Concepts"). With SWR, all server workloads are considered suspect (even if they ostensibly show no signs of infection and are systematically reprovisioned from known good images and templates) as a way to proactively displace malicious software that may have unknowingly gained a foothold in the system.

Note 6
Example Third-Party Software-Based Firewalls in AWS

Based on the need for advanced network security capabilities beyond what AWS security groups provide, many vendors have delivered solutions certified for use in AWS. Examples include:

- Alert Logic
- Barracuda Networks
- Brocade
- Check Point
- Cohesive Networks (VNS3)
- Fortinet
- Imperva
- Sophos
- McAfee
- Trend Micro

Note 7
Audit and Misconfiguration Risk

The common architectural weak point in most enterprise security architectures moving to AWS is the difficulty in matching the degree of network defense in depth as would be possible in internal-to-enterprise deployments. A good analogy is that most AWS deployments will be as a single firewall with multiple interfaces rather than multiple firewalls in depth. The best practice in network security is that no single compromise of an element should compromise the whole application stream. AWS VPC can provide virtual depth; however, the misconfiguration risk and resultant impact are higher than when there are physical firewalls providing separation and depth.

Note 8
Example Agent-Based Solutions That Natively Support AWS

- CloudPassage
- Dome9
- Illumio

Note 9
Example Full-Volume Encryption Solutions for AWS

- AWS (free, but doesn't encrypt the boot volume). As of May 2014, AWS provides this capability for EBS (the default storage of EC2) at no additional cost. Although this has not yet been integrated into CloudHSM, it is a common request of customers.[10] As of March 2015, Amazon's EBS encryption has been integrated into its new Key Management Service.
- HP (OEM Porticor's solution)
- Porticor
- SafeNet
- Trend Micro
- Vormetric

Note 10
Example WAF Solutions for AWS

- Alert Logic
- Barracuda
- DenyAll
- F5
- Fortinet
- Imperva

Source: Garter Research Note G00275221,
Neil MacDonald, Greg Young
15 April 2015

About Fortinet

Fortinet (NASDAQ: FTNT) is a worldwide provider of network security appliances and a market leader in unified threat management, next generation firewall and high performance datacenter firewall. Our products and subscription services provide broad, integrated and high-performance protection against dynamic security threats while simplifying the IT security infrastructure. Our customers include enterprises, service providers and government entities worldwide, including the majority of the 2012 Fortune Global 100. Fortinet is headquartered in Sunnyvale, Calif., with offices around the world.

A key differentiator, Fortinet's custom-built FortiASIC content and network processors enable our flagship FortiGate systems to detect and eliminate even complex, blended threats in real time without degrading network performance, while an extensive set of complementary management, analysis, database and endpoint protection solutions increases deployment flexibility, assists in compliance with industry and government regulations, and reduces the operational costs of security management.

US Headquarters
1090 Kifer Road
Sunnyvale, CA 94086
USA
Tel: +1-408-235-7700
Fax: +1-408-235-7737

Advanced Security at Scale in the Public Cloud is published by Fortinet. Editorial content supplied by Fortinet is independent of Gartner analysis. All Gartner research is used with Gartner's permission, and was originally published as part of Gartner's syndicated research service available to all entitled Gartner clients. © 2016 Gartner, Inc. and/or its affiliates. All rights reserved. The use of Gartner research in this publication does not indicate Gartner's endorsement of Fortinet's products and/or strategies. Reproduction or distribution of this publication in any form without Gartner's prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. The opinions expressed herein are subject to change without notice. Although Gartner research may include a discussion of related legal issues, Gartner does not provide legal advice or services and its research should not be construed or used as such. Gartner is a public company, and its shareholders may include firms and funds that have financial interests in entities covered in Gartner research. Gartner's Board of Directors may include senior managers of these firms or funds. Gartner research is produced independently by its research organization without input or influence from these firms, funds or their managers. For further information on the independence and integrity of Gartner research, see "Guiding Principles on Independence and Objectivity" on its website.