What CISOs Need to Know About Cloud Computing

Version 1.0
Released: January 6, 2014

Reviewed and Approved

Securosis, L.L.C.

http://securosis.com

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.

Reviewed and Approved by the Cloud Security Alliance

This content of this *independently created* paper has been reviewed and approved by the Cloud Security Alliance. It does not imply endorsement of any specific vendors or products. Securosis would like to thank the CSA for their support in reviewing the content.

For more information visit http://cloudsecurityalliance.org.

Licensed by CloudPassage

CloudPassage is the leading cloud infrastructure security company. Based on a next-generation security-as-a-service architecture, CloudPassage Halo was purpose built to automate security and compliance in any private, public, or hybrid cloud, or data center. Hundreds of businesses today, including some of the biggest cloud companies, rely on CloudPassage to deliver critical protection, visibility and control over these agile environments and achieve order of magnitude improvements in operational efficiency and business agility.

For more information please visit http://cloudpassage.com.

Copyright

Table of Contents

EXECUTIVE SUMMARY

What CISOs Need to Know About Cloud Computing

Cloud is Different, but Not the Way You Think

There is no question cloud computing is fundamentally changing how we deliver and consume technology resources, but the main impacts to security are not outsourcing or sharing infrastructure with others. Cloud computing doesn't necessarily reduce security risks, it shifts them.

Cloud computing is a radically different technology model – not just the latest flavor of outsourcing. It uses a combination of abstraction and automation to achieve previously impossible levels of efficiency and elasticity. But in the end cloud computing still relies on traditional infrastructure as its foundation.

Most security professionals focus on the risks of *multitenancy* in cloud computing, but the key risks actually result from **abstraction** and **automation**.

- Between business benefits and current adoption rates, *we expect cloud computing to become the dominant technology model over the next ten to fifteen years*. As we make this transition it is the technology that creates clouds, rather than the increased use of shared infrastructure, that really matters for security.
- *Multitenancy is more an emergent property* of cloud computing than a defining characteristic, despite being the trait many security professionals become distracted by.
- It is ***Abstraction*** *and* ***Automation*** that impact security more than multitenancy or outsourcing in cloud computing.
- **Abstraction** separates resources from the underlying infrastructure.
  - Your entire (cloud) infrastructure is now managed over the network using web interfaces and APIs. This *management plane* provides remote, complete, control over your infrastructure.
  - Security may lose visibility since we can't rely on physical network routing or asset management. We don't even necessarily know which exact hard drives hold which data.
  - Everything is virtual and portable. Entire servers can migrate to new physical systems with a few API calls or a click on a web page.

  - Compliance may be more complex due to less knowledge of where things are located, and auditors who don't understand cloud computing technologies.

- **Automation** uses *orchestration* technologies to manage provisioning and configuration of resources based on policies.

  - Security compliance is easier to automate, since everything runs through the cloud controller. *This reduces certain security risks*.

  - The environment is highly dynamic, with servers appearing and disappearing on-demand. Manual security controls or non-continuous assessments can't keep up.

  - You *gain greater governance and visibility* of your infrastructure, since the orchestration layer knows where everything is, at all times, and how it is configured.

Here are some examples of how cloud is different:

- **Autoscaling:** Monitoring tools in the cloud automatically launch new virtual servers based on templates to meet demand, then delete them when load drops. No human IT admin needed.

- **Immutable Servers:** Instead of patching servers, some organizations use the same techniques to launch new, up to date servers and delete the old ones. Even on live applications with users connected, thanks to new application architectures.

- **Snapshots:** A snapshot is a near-instant backup of all the data on a cloud storage volume, without taking the system down or affecting performance. These snapshots are incredibly portable and, in public clouds, can be made public.

- **Management Credentials:** The entire infrastructure deployed on the cloud is managed, even down to the network and server level, using API calls and web interfaces. Configured incorrectly, someone can own your datacenter by hacking an admin's personal laptop.

- **Software Defined Security:** Security can use these same features and APIs to automate security controls, and tightly integrate with the infrastructure. New servers automatically deploy with secure configurations, and you can instantly identify all digital assets.

Adapting Security for Cloud

Cloud computing poses new risks, while both increasing and decreasing existing risks. The trick is to leverage the security advantages, freeing up resources to cover the gaps. Start with Five general principles, all of which we see used today:

- *You cannot rely on boxes and wires*. Networks are virtual, so you can't put physical security devices inline, and virtual security tools behave differently.

- *Security should be as agile and elastic as the cloud itself.* Your security tools need to account for the highly dynamic nature of the cloud, where servers might pop up automatically and run for only an hour before disappearing forever.

- *Rely more on policy-based automation.* Wherever possible design your security to use the same automation as the cloud itself.
- *Understand and adjust for the characteristics of the cloud.* Take advantage of the native automation and orchestration of the cloud to embed security. By, for example, inserting security agents into virtual servers that automatically connect and self configure.
- *Integrate with DevOps*. Not all organizations are using DevOps, but DevOps principles are pervasive in cloud computing. Security teams can integrate with this approach and leverage it themselves for security benefits.

Real-World Examples

Here are a few examples of cloud security used in production environments today:

- Applications stacks are *hypersegregated* as the cloud platform places a virtual firewall around every single server, making it nearly impossible for an attacker to spread internally.
- Cloud-aware security agents are automatically embedded in every virtual machine as they launch. They then automatically configure themselves based on policies and the environment.
- Administrator access to the cloud management plane runs through security proxies to monitor all infrastructure changes.
- *Software Defined Security* programs constantly monitor the entire cloud for policy violations. They can automatically fix configurations and quarantine systems, or identify system owners and recommend changes.

Introduction

One of a CISO's most difficult challenges is sorting the valuable wheat from the overhyped chaff, and then figuring out what it all means in terms of risk to your organization. There is no shortage of technology or threat trends, and CISOs need to determine both which matter and how they impact security.

The rise of cloud computing is a true transformation, which is fundamentally changing core security practices. Far more than a mere outsourcing model, cloud computing alters the very fabric of our infrastructure, technology consumption, and delivery models. In the long run, the cloud and mobile computing are likely to mark a larger shift than the Internet itself.

This paper details the critical differences between cloud computing and traditional infrastructure for security professionals, and suggests where to focus security efforts. We will show that the cloud doesn't necessarily increase risks – instead it shifts them, providing new opportunities for substantial security improvement.

Different, but Not the Way You Think

Cloud computing is a radically different technology model – not just the latest flavor of outsourcing. It uses a combination of abstraction and automation to achieve previously impossible levels of efficiency and elasticity. But in the end cloud computing still relies on traditional infrastructure as its foundation. It doesn't eliminate physical servers, networks, or storage, but enables organizations to use them in different ways – with substantial benefits.

Sometimes this means building your own cloud in your own datacenter; other times it means renting infrastructure, platforms, and applications from public providers over the Internet. Most organizations will use a combination of both. Public cloud services eliminate most capital expenses, shifting them to on-demand operational costs instead. Private clouds allow more efficient use of capital, may reduce operational costs, and make technology more responsive to internal needs.

Between business benefits and current adoption rates, we expect cloud computing to become the dominant technology model over the next ten to fifteen years. As we make this transition it is the technology that creates clouds, rather than the increased use of shared infrastructure, that really matters for security. Multitenancy is more an emergent property of cloud computing than a defining characteristic, despite being the trait many security professionals become distracted by.

Security Is Evolving for the Cloud

Cloud computing isn't really more or less secure than traditional infrastructure – it is different. Some risks are greater, some are new, some are reduced, and some are eliminated. Our primary goal for this paper is to provide an overview of where these changes occur, what you need to do about them, and when.

Cloud security focuses on the different risks associated with *abstraction* and *automation*. Multitenancy tends to be more a compliance issue than a security problem, and we will cover both aspects. Infrastructure and applications are opened up to network-based management via Internet APIs. Everything from core network routing to creating and destroying entire application stacks is now possible using command lines and web interfaces. The early security focus has been on managing risks introduced by highly dynamic virtualized environments such as autoscaled servers, and broad network access, including a major focus on compartmentalizing cloud management.

Focus is gradually shifting to hardening cloud infrastructure, platforms, and applications, and then adapting approaches to use the cloud to improve security. For example, the need for data encryption increases as you migrate more sensitive information into the cloud. But the complexities of internal network compartmentalization and server patching are dramatically reduced as you leverage cloud infrastructure.

We expect to eventually see more security teams hook into the cloud fabric itself – bridging gaps between security tools and infrastructure and applications with Software Defined Security. The same APIs and programming techniques that power cloud computing can provide highly-integrated dynamic and responsive security controls – this is already happening today.

This paper describes the key differences, with suggestions for where security professionals should focus. Hopefully, by the end, you will see the cloud and cloud security in a new light, and agree that the cloud isn't just the latest flavor of shared services.

How the Cloud Is Different for Security

In the early days of cloud computing, even some very well-respected security professionals claimed it was little more than a different kind of outsourcing, or equivalent to multitenancy on a mainframe. But the differences run far deeper, and demand different cloud security controls. We know how to manage the risks of outsourcing or multi-user environments – cloud computing security builds on this foundation and adds new twists.

These differences boil down to *abstraction* and *automation*, which separate cloud computing from basic virtualization and other well-understood technologies.

Abstraction

In the cloud context, abstraction is the extensive use of multiple virtualization technologies to separate compute, network, storage, information, and application resources from their underlying physical infrastructure. In cloud computing we convert physical infrastructure into *resource pools* that are sliced, diced, provisioned, deprovisioned, and configured on demand with *automation* (discussed next).

It really is a bit like the matrix. Individual servers run little more than a hypervisor with connectivity software to link them into the cloud, and the rest is managed by the cloud controller. Virtual networks overlay the physical network, with dynamically configured routing at all levels. Storage hardware is similarly pooled, virtualized, and then managed by cloud control layers. The entire physical infrastructure, less some dedicated management components, becomes a collection of resource pools. Servers, applications, and everything else runs on top of the virtualized environment.

Abstraction impacts security significantly in four ways:

- Resource pools are managed using standard, web-based (REST) Application Programming Interfaces (APIs). The infrastructure is managed with network-enabled software at a fundamental level.
- Security can lose visibility into the infrastructure. On the network we can't rely on physical routing for traffic inspection or management. We don't necessarily know which hard drives hold which data.
- Everything is virtualized and portable. Entire servers can migrate to new physical systems with a few API calls or a click on a web page.
- We gain greater pervasive visibility into the infrastructure configuration itself. If the cloud controller doesn't know about a server it cannot function. We can map the complete environment with the same API calls.

We have focused on *Infrastructure as a Service*, but the same issues apply to *Platform and Software as a Service*, except they often offer even less visibility.

Automation

Virtualization has existed for a long time. The real power cloud computing adds is ***automation***. With basic virtualization and virtual data centers, we still rely on administrators to manually provision and manage virtual machines, networks, and storage. Cloud computing turns these tasks over to the cloud controller to coordinate all these pieces and more, using *orchestration*.

Users request resources via web page or API call, such as a new server who needs 1tb of storage on a particular subnet, and the cloud management software determines how best to provision it from the resource pool; then it handles installation and configuration, and coordinates all the necessary networking/storage/compute/etc. resources to pull everything together into a functional and accessible server. No human administrator required.

Or the cloud management system can monitor demand on a cluster and add and remove fully load-balanced and configured systems based on rules, such as a configurable maximum threshold for average system utilization. Need more resources? Add virtual servers. Systems underutilized? Drop them back into the resource pool. Need to perform complex analysis on petabytes of storage? Just rent the storage space and processing power until the job completes. In public clouds this keeps costs down as you expand and contract precisely based on immediate requirements. In private clouds it frees resources for other projects and requirements, but you still need a shared resource pool to accommodate total demand. But you are no longer stuck with underutilized physical boxes in one corner of the data center, and inadequate capacity in another.

Our primary focus is on Infrastructure and Platform as a Service (IaaS and PaaS), but the principles in this paper also apply to Software as a Service (SaaS), you just have less control

This is true for all three main cloud delivery methods: *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) and *Software as a Service* (SaaS). You can expand and contract database storage, software application server capacity, the number of supported users, and storage as needed without additional capital investment.

In the real world it isn't always so clean. Heavy use of public cloud may exceed the costs of owning your own infrastructure. Managing your own private cloud is no small task and ripe with pitfalls. And abstraction does reduce performance at certain levels, at least for now. But with the right planning, and as the technology continues to evolve, the business advantages are undeniable.

But even more transformative is the capability for applications to manage their own infrastructure because everything is now programmable. The lines between development and operations blur, offering incredible agility and resilience, which is one of the concepts underpinning the DevOps movement (explained below). But of course done improperly it can be disastrous.

Cloud, DevOps, and Security in Practice: Examples

Here are a few examples to highlight the impact of abstraction and automation on operations and security:

- **Autoscaling:** As mentioned above, many IaaS providers support autoscaling. A monitoring tool watches server load and other variables. When the average load of virtual machines exceeds a configurable threshold, new instances are launched from the same base image with advanced initialization scripts. These scripts automatically configure all aspects of the server, pulling metadata from the cloud or a configuration management system. Advanced tools can configure entire application stacks. But these servers may exist only for a short period – perhaps never coinciding with a vulnerability assessment window. Or images might launch in the wrong zone,

with the wrong network security rules. The images and initialization scripts might not be up to date for the latest security vulnerabilities, creating cracks in your defenses.

- **Immutable Servers:** Autoscaling can spontaneously and automatically orchestrate the addition and subtraction of servers and other resources. The same concepts can, in some cases, eliminate the need to patch. Instead of patching a running server you might use the same scripting and configuration management techniques, behind a virtual load balancer, to launch a new up-to-date version of a server, and then destroy the old unpatched virtual machine.
- **Snapshots:** Cloud data typically relies on virtual storage, and even running servers use what are essentially virtual hard drives with RAID. A snapshot is a near-instant backup of all the data on a storage volume, without taking the system down or affecting performance. These snapshots are incredibly portable and, in public clouds, can be published with an API call. Or you could write a program to snapshot all your servers at once (if your cloud has the capacity). This is great for forensics, but also enables an attacker to copy your entire data center and make it public in about 10 lines of code.
- **Management Credentials:** The entire infrastructure deployed on the cloud is managed, even down to the network and server level, using API calls and perhaps web interfaces. Administrator tools typically keep these credentials in memory as environment variables or in the registry, making them accessible even without administrative control over the cloud admin's workstation. Additionally, few clouds offer audit logging of administration commands. Many organizations fail to compartmentalize the rights of cloud administrators, leaving their entire infrastructure open to a single compromised system.
- **Software Defined Security:** With roughly 20 lines of code you can connect to your cloud over one API, your configuration management tool with another, and your security tool with a third. You can instantly assess the configuration and security of every server in your environment, without scanning, in real time. This is nearly impossible with traditional security tools.

Snapshots highlight some of the risks of abstraction. Autoscaling illustrates some risks of automation. Management credential exploits demonstrate the risks of both. But Software Defined Security and immutable servers offer countervailing advantages. Now that we have highlighted the core differences, we can dig into specifics.

And all that without once mentioning multitenancy or outsourcing.

The NIST Model

The NIST model of cloud computing is the best framework for understanding the cloud. It consists of five Essential Characteristics, three Service Models (IaaS, PaaS, and SaaS) and four Delivery Models (public, private, hybrid and community). Our characteristic of abstraction generally maps to resource pooling and broad network access, while automation maps to on-demand self service, measured service, and rapid elasticity. We aren't proposing a different model, just overlaying the NIST model to better describe things in terms of security.

Adapting Security for Cloud Computing

If you didn't already, you should now have a decent understanding of how cloud computing differs from traditional infrastructure, so we can switch gears to evolving security to address shifting risks.

These examples are far from comprehensive, but offer a good start with a sampler of how to think differently about cloud security.

General Principles

As we keep emphasizing, taking advantage of the cloud poses new risks, while both increasing and decreasing existing risks. The trick is to leverage the security advantages, freeing up resources to cover the gaps. There are a few general principles for approaching the problem to put you in the proper state of mind:

- *You cannot rely on boxes and wires*. Quite a bit of classical security relies on knowing the physical locations of systems and the network cables connecting them. Network traffic in cloud computing is virtualized, which completely breaks this model. Network routing and security are instead defined by software rules. There are some advantages in the new model, beyond the scope of this paper but which we will detail in future research.
- *Security should be as agile and elastic as the cloud itself.* Your security tools need to account for the highly dynamic nature of the cloud, where servers might pop up automatically and run for only an hour before disappearing forever.
- *Rely more on policy-based automation.* Wherever possible design your security to use the same automation as the cloud itself. For example there are techniques to automate (virtual) firewall rules based on tags associated with a server, rather than applying them manually.
- *Understand and adjust for the characteristics of the cloud.* Most virtual network adapters in cloud platforms disable network sniffing, so that risk drops off the list. Security groups are essentially virtual firewalls on individual instance, so you get full internal firewalls and compartmentalization by default. Security tools can be embedded in images or installation scripts to ensure they are always installed, and cloud-aware tools can auto-configure. SAML can be used to provide absolute device and user authentication control to external SaaS applications. All these and more are enabled by the cloud once you understand its characteristics.
- *Integrate with DevOps*. Not all organizations are using DevOps, but DevOps principles are pervasive in cloud computing. Security teams can integrate with this approach and leverage it themselves for security benefits, such as automating security configuration policy enforcement.

These principles will get you thinking in cloud terms, but let's look at specifics.

Defining DevOps

DevOps is an IT model that blurs the lines between development and IT operations. Developers play a stronger role in managing their own infrastructure through heavy use of programming and automation. Since cloud enables management of infrastructure using APIs, it is a major enabler of DevOps. While it is incredibly agile and powerful, lacking proper governance and policies it can also be disastrous since it condenses many of the usual application development and operations check points.

Control the Management Plane

The management plane comprises the administrative interfaces – both web and API – used to manage your cloud systems. It exists in all types of cloud computing service models: IaaS, PaaS, and SaaS. Someone who compromises a cloud administrator's credentials has the equivalent of unmonitored physical access to your entire data center, with enough spare hard drives, forklifts, and trucks to copy the entire thing and drive away. Or blow the whole thing up.

We cannot overstate the importance of hardening the management plane. It literally provides absolute control over your cloud deployment – often including all disaster recovery.

We offer five recommendations for securing the management plane:

- If you manage a private cloud, ensure you *harden the web and API servers*, keeping all components up to date and protecting them with the highest levels of web application security. This is no different than protecting any other critical web server.
- *Leverage the Identity and Access Management features* offered by the management plane. Some providers offer very fine-grained controls. Most also integrate with your existing IAM using federated identity. Give preference to your platform/provider's controls and…
- *Compartmentalize with IAM*. No administrator should have full rights to all aspects of the cloud. Many providers and platforms support granular controls, including roles and groups, which you can leverage to restrict the damage potential of a compromised developer or workstation. For example you might have a separate administrator for assigning IAM rights, only allow administrators to manage certain segments of your cloud, and further restrict them from terminating instances.
- *Add auditing, logging, and alerting where possible*. This is one of the more difficult problems in cloud security because few cloud providers audit administrator activity – such as who launched or stopped a server using the API. For now you will likely need a third-party tool or to select providers who offer adequate auditing.
- *Consider using security or cloud management proxies*. These tools and services proxy the connection between a cloud administrator and the public or private cloud management plane. They can apply additional security rules and fill logging and auditing gaps.

Automate Host (Instance) Security

An instance is a virtual machine, based on a stored template called an image. When you request a server from the cloud, you specify the image to base it on, which includes the operating system and might bring a complete single-server

application stack. The cloud then configures it using scripts which can embed administrator credentials, provide IP addresses, attach and configure storage, etc.

Instances can exist for years or minutes, are configured dynamically, and can be launched nearly anywhere in your infrastructure – whether public or private. You cannot rely on manually assessing and adjusting their security. This is very different than building a server in a test environment, performing a vulnerability scan, and then physically installing it behind a particular firewall and IPS. More security needs to be embedded in the host itself, as well as the images instances are based on.

Fortunately, these techniques improve your ability to enforce secure configurations.

- *Embed security in images and at launch*. If you use your own images you can embed security settings and agents in the base images, and set them to activate and self-configure (after connecting to their management server) when an instance launches. Alternatively, many clouds and images support passing initialization scripts to new instances, which process them during launch using the same framework the cloud itself uses to configure essential settings. You can embed security settings and install security software in these scripts.
- *Integrate with configuration management*. Most serious cloud administrators rely on a new breed of configuration management tools to dynamically manage their systems with automation. Security can leverage these to enforce base security configurations, even down to specific application settings, which apply to all managed systems. Set properly, these configuration management tools handle both initial configuration and maintaining state, overwriting local changes as necessary in response to policy pulls and pushes.
- *Dynamically configure security agents*. When security agents are embedded into images or installed automatically on launch, default settings rarely meet a system's particular security requirements. Fortunately cloud platforms offer rich metadata on instances and their environment – including system configuration, network configuration, applications installed, etc. Cloud-aware security agents connect to the management server on launch, and then self-configure with policies leveraging its information. They can also update settings in near-real-time based on new policies or changes in the cloud.
- *Security agents should be lightweight, designed for the cloud, and cloud agnostic*. You should not need 8 different security agents for 12 different cloud providers, and agents shouldn't materially increase system resource requirements or struggle to communicate in a dynamic and virtual network.
- *Host security tools should support REST APIs*. This enables you to integrate your security into the cloud fabric itself as needed. Agents don't necessarily need to communicate with the management server over a REST API, but the management server should expose key functions via (secure) API. This enables you to, for example, write scripts to extract host security information, compare it against network security information, and make adjustments or report as necessary. Or integrate security alerts and status from the host tool into your SIEM without additional connectors.

REST APIs are the dominant format for APIs in web-friendly applications. They run over HTTP and are much easier to integrate and manage compared to older SOAP APIs.

Intelligently Encrypt

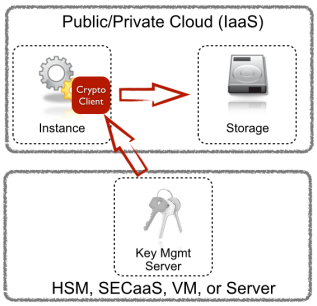
There are three reasons to encrypt data in the cloud, in order of importance:

1. Compliance.

2. To protect data in backups, snapshots, and other portable copies or extracts.
3. To protect data from cloud administrators.

How you encrypt varies greatly depending on where the data resides and which particular risks most concern you. For example many cloud providers encrypt object file storage or SaaS by default, but they manage the keys. This is often acceptable for compliance but doesn't protect against a management plane breach.

We wrote a paper on infrastructure encryption for the cloud, from which we have extracted requirements which apply across different encryption scenarios:



Encrypt with External Key Management

- If you are encrypting for security (as opposed to targeting a compliance checkbox) you need to *manage your own keys*. If the vendor manages your keys, your data may be exposed by a management plane compromise.
- *Separate key management from cloud administration.* Sure, we are all into DevOps and flattening management, but this is one situation where security should manage outside the cloud management plane.
- *Use key managers that are as agile and elastic as the cloud.* Like host security agents, cloud key managers need to operate in environments where servers appear and disappear automatically and networks are virtual.
- *Minimize SaaS encryption*. The only way to encrypt data going to a SaaS provider is with a proxy, and encryption breaks the processing of data at the cloud provider. This reduces the utility of the service, so minimize which fields you need to encrypt. Or, better yet, trust your provider, but only if they provide acceptable SLAs and solid documentation of adequate security practices.
- *Use secure cryptography agents and libraries* when embedding encryption in hosts or IaaS and PaaS applications. The defaults for most crypto libraries used by developers are not secure. Either understand how to make them secure or use libraries designed from the ground up for security.

Federate and Automate Identity Management

Managing users and access in the cloud introduces two major headaches:

- Controlling access to external services without having to manage a separate set of users for each.
- Managing access to potentially thousands or tens of thousands of ephemeral virtual machines, some of which may only exist for a few hours.

For the first case, and often for the second, federated identity is the way to go.

- For external cloud services, especially SaaS, *rely on SAML-based federated identity* linked to your existing directory server. If you deal with many services this can become messy to manage and program yourself, so consider one of the identity management proxies or services designed specifically to tackle this problem.
- For access to your actual virtual servers, consider managing users with a *dynamic privilege management agent* designed for the cloud. Normally you embed SSH keys (or known Windows admin passwords) as part of instance initialization – the cloud controller handles this for you. This is highly problematic for privileged users at

scale, and even straight directory server integration is often quite difficult. Specialized agents designed for cloud computing dynamically update users, privileges, and credentials at cloud scale and speed.

Adapt Network Security

Networks are completely virtualized in cloud computing, although different platforms use different architectures and implementation mechanisms to complicate things further. Despite the diversity, there are consistent traits to focus on. The key issues come down to loss of visibility using classical techniques, and adapting to the dynamic nature of cloud computing.

All public cloud providers disable networking sniffing, and that is an option on all private cloud platforms. A bad guy cannot hack a box to sniff the entire network, but you also cannot implement IDS and other network security like in traditional infrastructure. Even when you can place a physical box on the network hosting the cloud, you miss traffic between instances on the same physical server, and scheduled scanning either misses entirely or is too slow to be useful for detecting highly dynamic network changes or short-lived server instances. You can sometimes use a virtual appliance instead, but even tools that can run in virtual environments are likely to crack in the cloud due to performance and functional limitations, unless they were designed with cloud demands in mind.

While you can embed more host network security in the images your virtual machines are based on, the standard tools typically don't work because they don't know exactly where on the network they will pop up, nor what addresses they need to talk to. On the other hand, all cloud platforms include basic network security. Set your defaults properly and every single server effectively comes with its own firewall.

We recommend several steps:

- *Design a good baseline of Security Groups* (the basic firewalls that secure the networking of each instance), and use tags or other mechanisms to automatically apply them based on server characteristics. A Security Group is essentially a firewall around each instance, offering a level of compartmentalization that is extremely difficult to manage in a traditional network.
- *Use a host firewall or host firewall management tool* designed for your cloud platform or provider. These connect to the cloud itself to fetch metadata, and configure themselves more dynamically than standard host firewalls.
- Consider *pushing more network security, including IDS and logging, into your instances*.
- P*refer virtual network security appliances that support cloud APIs* and are designed for the cloud platform or provider. For example, instead of forcing you to route all your virtual traffic through it as if you were on a physical network, the tool could distribute its own workload across instances – perhaps even integrating with hypervisors.
- *Take advantage of cloud APIs*. It is very easy to pull every Security Group rule and then locate every instance. Combined with some other basic tools, you could then automate finding errors and omissions. Many cloud deployments do this today as a matter of course.
- Whatever tools you use, they must be able to handle the high rate of churn as servers appear and disappear in the cloud. *Security policies must follow the virtual machines.*

Leverage Cloud Characteristics

This section is a bit more advanced, but you can reap significant security advantages by leveraging the nature of the cloud. Instead of patching, just launch properly configured new servers and swap them in using a cloud load balancer.

Find every single system in your cloud deployment, with extensive metadata, using a simple API call. Deploy applications to a Platform as a Service (PaaS) and stop worrying about misconfigured servers. Here are some real world examples and recommendations to get you started, but these barely scratch the surface:

- *Use immutable servers* instead of patching, where practical. Few admins patch servers without trepidation, and patching and not patching are both frequent sources of downtime. Eliminate the worry by running your applications behind cloud load balancers, and instead of patching simply launch new servers with the updates in place. Then slowly (or quickly) switch traffic to the new 'patched' servers. If something doesn't work your old servers are still there and you can shift traffic back. It's like having a spare data center lying around.

- *Leverage stateless security* to manage your environment in real time. Normally we rely on knowledge from scanners and assessments to understand our assets and environments, which can be out of date and difficult to keep complete. The cloud controller knows where everything is, how it is configured (to a degree), and even who owns or created it, so we have a constant stream of comprehensive real-time data. A server cannot exist in the cloud without the controller knowing about it. Your entire network architecture is just an API call away – no scanners needed. Track and manage your security state in real time.

- *Automate more security.* Embed security configurations and agents into images, or inject them into instances when they launch. Every virtual machine, when launched, can automatically configure host-based security – especially if it can communicate with a management server designed for the cloud. For example a host could register itself with a configuration management server, then secure running services by default depending on its intended role. You can even automatically adjust Security Group firewall rules based on the software services running on the host, who owns it, and where it sits in your application stack.

- *Standardize security with Platform as a Service*. Hate patching database servers or configuring them properly? Struggle with developers and admins who open up too many services on application servers? Use Platform as a Service instead, and improve your ability to standardize security.

- *Build a security abstraction layer.* Nothing prevents your security team from using the same cloud APIs and management tools as administrators and developers. Configured and used properly, they provide security oversight and control without interfering with development or operations. For example you might restrict management of cloud IAM to the security team, enabling them to assume management of a server in case of a security incident. The security team could control key network Security Groups and security in production, while still allowing developers to manage network security directly in more isolated development environments. Embed a host security agent into every image (or instance, using launch scripts) and security gains a hook into every running virtual machine.

- *Move to Software Defined Security.* This concept is an extension of basic automation. Security can write its own programs using cloud APIs and the APIs of security and operations tools, in order to create powerful and agile security controls. For example a small program could find every instance in your environment that isn't linked into your configuration management tool, rechecking every few minutes. The tool would identify who launched the server, its operating system, where it was on the network, and the surrounding network security. You could, at a keystroke, take control of the server, notify the owner, and isolate it on the network until you know what it is for; then integrate it into configuration management and enforce security policies. All with perhaps 100 lines of code.

These should get you thinking – they start to show how the cloud can nearly eradicate certain security problems and enable you to shift resources.

Real World Examples

Cloud computing covers such a wide range of different technologies that there is no shortage of examples to draw from. Here are a few generic examples from real-world deployments. These get more technical because we want to highlight practical, tactical techniques to prove we aren't just making it all up:

Embedding and Validating a Security Agent Automatically

In a traditional environment we embed security agents by building them into standard images or requiring server administrators to install and register them. Both options are highly prone to error and omission, and hard to validate because you often need to rely on manual scanning. Both issues become much easier to manage in cloud computing.

To embed the agent:

- The first option is to build the agent into images. Instead of using generic operating system images build your own, then restrict users to launching only approved images. In a private cloud you can enforce this with absolute control of what users run. In public clouds enforcement is a bit tougher, but you can quickly catch exceptions using our validation process.
- The second option, and our favorite, is to inject the agent when instances launch. Some operating systems support initialization scripts which are passed to the launching instance by the cloud controller. Depending on your cloud platform, you can inject these scrips automatically when autoscaling, via a management portal, or manually at other times. These scripts install and configure software in the instance before it is accessible on the network.
- Either way you need an agent which understands how to work on cloud infrastructure and is capable of self-registering to the management server. The agent pulls system information and cloud metadata, then connects with its management server, which pushes configuration policies back to the agent so it can auto-configure. This process is entirely automated when the agent runs.
- Configuration may be keyed to detected services running on the instance, metadata tags applied to the instance (in the cloud management plane), or other characteristics such as network location.
- Our Software Defined Security paper provides a detailed technical example of agent injection and self-configuration.

The process is simple. Build the agent into images or inject it into launching instances, then have it connect to a management server to configure itself. Agent capabilities vary widely. Some replicate standard endpoint protection but others handle system configuration, administrative user management, log collection, network security, host hardening, and more.

Validating that all your instances are protected can be quite easy, especially if your security management tool supports your cloud APIs:

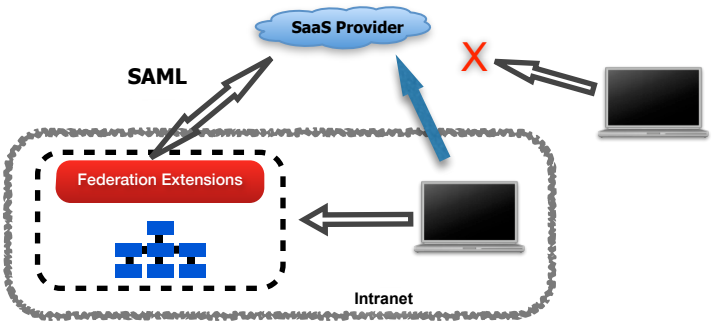
- Obtain a list of all running instances from the cloud controller. This is often as easy as a simple API call.
- Obtain a list of all instances running the security agent. This should be an API call to your security management platform, but might require pulling a report if that isn't supported.
- Compare the lists. You cannot hide in the cloud, so you know every single instance. Compare active instances against managed instances and find the exceptions.

This paper shows how.

Controlling SaaS with SAML

Pretty much everyone uses some form of Software as a Service, but controlling access and managing users can be a headache. Unless you link up using federated identity you need to manage user accounts on the SaaS platform manually. Adding, configuring, and removing users on yet another system, which is always Internet accessible, is daunting. Federated identity solves this problem.

- Enable federated identity extensions on your directory server. This is an option for Active Directory and most LDAP servers.
- Contact your cloud provider to obtain their SAML configuration and management requirements. SAML (Security Assertion Markup Language) is a semi-standardized way for a relying party to allow access and activities based on approval from an identity provider.
- Configure SAML yourself or use a third-party tool compatible with your cloud provider(s) which does this for you. If you use several SaaS providers a tool will save considerable effort.
- With SAML users don't have their own usernames and passwords with the cloud provider. The only way to log in is to first authenticate to your directory server, which then provides (invisible to the user) a token allowing access to the cloud provider. Users need to be in the office or come through the VPN rather than connecting directly to the cloud, because the cloud service does not have their username or password.
- If you want to enable remote users without VPN you can set up a cloud proxy and issue a special URL to use instead of the SaaS provider's standard address. This address redirects to your proxy, which then handles connecting back to your directory server for authentication and authorization. This is something you typically buy rather than build.



Users authenticate on the corporate intranet, and are given a SAML token to log into the cloud provider. They *cannot log in* unless first authenticating to the directory server.

Why do this? Rather than creating users on the SaaS platform, you can instead use existing user accounts in your directory server and authorize access using standard roles and groups, just like you do for internal servers. This way you

also get to track logins, disable accounts from a single source (your directory server), and otherwise maintain control. It also means people can't steal a user's password and then access your data in Salesforce.com from anywhere on the Internet.

Compartmentalizing Cloud Management with IAM

One of the worst new risks in cloud computing is Internet-accessible management of your entire infrastructure. Most cloud administrators use cloud APIs and command line interfaces to manage the infrastructure (or PaaS, and sometimes even SaaS). This means access credentials are accessed through environment variables or even the registry. If they use a web interface they are exposed to browser-based attacks. Either way, without capability compartmentalization an attacker could take complete control over the infrastructure by merely hacking one laptop. With a few API calls or a script they could copy or destroy everything in minutes.

All cloud platforms support internal identity and access management to varying degrees – this is something to look for during your selection process. You can use this to limit security risks – not just to break out development and operations teams. The following scenario isn't supported on all platforms yet, but it gives you an idea of possible options:

- Create a security team group and assign it IAM rights, and remove these rights from all other groups. "IAM rights" means the security team manages new users, changes user and group rights, and prevents privilege escalation. They can even revoke administrative access to running instances by modifying the associated rights.
- Use separate cloud development and production groups and accounts. Even if you use DevOps, require users to switch accounts for different tasks.
- The development group can have complete control over a development environment, which is segregated from the operations environment. Restrict them to building and launching in cloud segments that are isolated from the Internet and only route back to your organization. Developers can have free access to create, destroy, and otherwise manage development instances.
- In your production environment break out administrative tasks. Restrict all snapshotting and termination of instances to separate roles. This prevents attackers from copying data or destroying servers unless they manage to get into one of those accounts.
- Security Group changes should be restricted to the security team (or another designated group). Cloud administrators can move instances into and out of different Security Groups if necessary (although ideally you would also restrict this), but only a small team should set the rules for production.
- You will still need super-admin accounts, but these can be highly restricted and used as infrequently as possible.
- In general use different groups, with different credentials, for different parts of your infrastructure. For example in production you could break out management by application stack.
- If you need auditing of API calls, and your cloud platform doesn't support it, require administrators to connect through a proxy server that logs activity.

This way an attacker needs to break into multiple accounts to cause the worst damage. Note that what we just described isn't necessarily easy to manage at scale – this is an area where you might spend resources freed up from other security tasks such as patching.

Hypersegregate with Security Groups

Our last example is also one of the simplest and most powerful.

As mentioned earlier, a *Security Group* is essentially a basic stateless firewall implemented by the cloud platform. They are like a small cheap firewall in front of each server. When starting with Security Groups many users think of them like subnet firewalls, but that isn't quite how they work. A subnet firewall in front of a group protects access to the systems in the group from the outside (and perhaps vice-versa). A Security Group is more like a firewall policy applied on a per-system level. Instances in a Security Group cannot communicate with other instances in the same group unless you create an explicit rule to allow that.

Every single instance is, by default, firewalled off from every other one. This enables an incredible level of compartmentalization we like to call *hypersegregation* – we're analysts and like to make up our own words.

For example, within an application stack you will likely have multiple instances of your web servers, application servers, and database servers. Each of those should be in a Security Group that allows it only to talk to the layers immediately above and below in the stack. Instances in a Security Group shouldn't be allowed to talk to each other so cracking a server only allows very limited communications, over approved ports and protocols, to the servers directly above and below.

Security Groups also should not allow any public Internet access (except from the web server group). Administrative access should be restricted to known addresses from either a jump server or your internal IP range.

Better yet, instead of always leaving every server open to administrative access, keep it closed when not actively in use. Then adjust the individual server's Security Group to make the change, and remove administrative access again. You do this through the cloud management plane so an attacker would need to crack the management plane *and* obtain server credentials to access the server.

This setup is nearly impossible to create with traditional infrastructure. We cannot afford all those physical firewalls, and creating that many switch-based rules is a non-starter at scale. We could do it using host firewall rules, but managing those across multiple platforms in a dynamic environment is insanely complex.

This is a case where the cloud offers substantially *better security by default.*

Where to Go from Here

This paper can only offer a high-level overview to highlight how cloud computing is different for security, and to offer ideas for how to adjust your security controls to leverage its advantages while taking the different risks into account. The devil is in the details and we always worry that these overviews oversimplify.

But every single thing described above is being used, today, in the real world. These aren't cases of "maybe it will work", but examples of what leading cloud users are implementing on a daily basis. Our examples are generally far more basic than what we have seen in practice.

The problem is that most security professionals don't have the time or resources to become cloud security experts. Their days are filled with the ongoing minutiae of stopping attacks, meeting compliance requirements, and fighting fires. It becomes easy to dismiss cloud computing as yet another fad or trend we can manage as we always have, especially in light of vendors' deluge of announcements that their products work *just the same* in the cloud.

But cloud computing is ***not*** business as usual. It is an entirely new technology and operations model that fundamentally disrupts existing practices. One with a staggering rate of change, where entirely new platforms and capabilities emerge constantly. Two years ago big data was accessible only to those with top-line resources and massive data centers. Now anyone can rent petabyte-scale data warehouses for a few hours of analysis. Using a web browser.

Adoption of the cloud will only accelerate, and it is vital that security professionals come up to speed on the technologies and adjust to meet new demands. The opportunities to improve security over existing practices are powerful and practical.

We will continue to cover this in depth in future research, digging into the specifics of *how* to handle cloud security and what it means to existing practices. We hope you find it useful.

Who We Are

About the Analyst

Rich Mogull, Analyst and CEO

Rich Mogull, Analyst & CEO Rich has twenty years experience in information security, physical security, and risk management. He specializes in cloud security, data security, emerging security technologies, and security management. Rich is the primary developer of the Cloud Security Alliance CCSK training program. 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, Black Hat, 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, L.L.C. 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 project accelerator workshops, product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessments.
- Strategic consulting for vendors: Including market and product analysis and strategy, technology guidance, product evaluations, and merger and acquisition assessments.
- Investor consulting: 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.