acmqueue

Securing Elasticity in the Cloud

Elastic computing has great potential, but many security challenges remain.

Dustin Owens, BT Americas

As somewhat of a technology-hype curmudgeon, I was until very recently in the camp that believed cloud computing was not much more than the latest marketing-driven hysteria for an idea that has been around for years. Outsourced IT infrastructure services, aka IaaS (Infrastructure as a Service), has been around since at least the 1980s, delivered by the telecommunication companies and major IT outsourcers. Hosted applications, aka PaaS (Platform as a Service) and SaaS (Software as a Service), were in vogue in the 1990s in the form of ASPs (application service providers).

Looking at cloud computing through this perspective had me predicting how many more months it would be before the industry came up with another "exciting" technology with which to generate mass confusion and buzz. But… I have recently been enlightened as to the true potential of cloud computing and have become very excited about it, to say the least. This concept, which has generated the most industry hype in years—and which has executives clamoring for availability because of promises of substantial IT cost savings and innovation possibilities—has finally won me over.

So, what did I discover about cloud computing that has made a convert out of someone who was so adamantly convinced that it was nothing more than the latest industry topic du jour? First let me explain that it was no small feat. It took a lot of work to sort through the amazing amount of confusion concerning the definition of cloud computing, let alone find a nugget of real potential. Definitions abound, and with my curmudgeon hat still solidly in place I was beginning to see a lot of hair-splitting and "me too" definitions that just seemed to exacerbate the problem. I finally settled on the definition provided by NIST (National Institute of Standards and Technology) because of the simplicity the framework provides (see sidebar). Still, it wasn't until a good friend who had already discovered the real potential hidden in all this madness provided me with some real-world use cases for *elasticity* that the light began shining very brightly.

Elasticity, in my very humble opinion, is the true golden nugget of cloud computing and what makes the entire concept extraordinarily evolutionary, if not revolutionary. NIST's definition of elasticity is as follows: "Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time." When elasticity is combined with on-demand self-service capabilities it could truly become a game-changing force for IT.

Advanced outsourced IT infrastructure and software services, once available only to organizations with large budgets available to develop, build, and support ongoing use of these resources, can now be provided to small to medium organizations. In addition, these resources can be added, changed, or removed much more rapidly, potentially allowing for exponential advances in operational efficiency. The sorts of changes to major IT services environments that previously (and for the most part currently) took months if not years to plan and execute might be done in a matter of

minutes or hours if elasticity lives up to its promise. In other words, elasticity could bring to the IT infrastructure what Henry Ford brought to the automotive industry with assembly lines and mass production: affordability and substantial improvements on time to market.

Enlightening as this realization has been, it has also become clear that several monumental security challenges (not to mention many monumental nonsecurity-related challenges, not least of which are full functionality availability and how well an organization's environment is prepared to operate in a distributed model) now come into play and will need to be addressed in order for the elasticity element of cloud computing to reach its full potential. Most of the dialogue I am engaged in with customers today and that I see in publicized form, however, is simplistically centered on security challenges with IT outsourcing in general. These are challenges that have existed for some time in the predecessor models mentioned earlier: who within an outsourcer is able to access a customer's data, perimeter security considerations when outsourcing, DOS/DDOS (denial of service/distributed denial of service), resource starvation, and compliance challenges with where data is stored or backed up. These are all challenges that I have been providing counsel on for many years and are nothing new or insurmountable. Don't misunderstand me. These challenges are indeed very real and still need to be addressed, but I strongly believe most should be fairly well known by now and can be readily met through existing procedural or technological solutions.

The challenges I am more concerned about are those introduced by adding elasticity and on-demand self-service to form the full extent of cloud computing—those elements that in my opinion make a particular service something more than just an outsourced service with a prettier marketing face.

ELASTICITY SECURITY CHALLENGES

Enabling elasticity in the cloud strongly implies the use of virtualization. Though virtualization and the inherent security challenges it brings are certainly not new, *how* it is likely to be used by cloud-computing providers to achieve elastic IT environments on a grand scale poses some interesting security challenges worth exploring in more detail. In addition, as virtualization technology continues to evolve and gain popularity, new vulnerabilities continue to be discovered; witness the recently announced vulnerability (http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-3733) whereby one is able to traverse from one VM (virtual machine) client environment to other client environments being managed by the same hypervisor.

These new vulnerabilities could have significantly greater impacts in the cloud-computing arena than within an organization's corporate environment, especially if not dealt with expeditiously. Case in point: imagine that many customers are being managed by a single hypervisor within a cloud provider. The vulnerability noted above might allow a customer to access the virtual instances of other customers' applications if not addressed. Consider the impact if your bank or particularly sensitive federal government or national defense information happen to be managed in this sort of environment, and the cloud provider does not immediately deal with, or even know about, a vulnerability of this nature.

With this bit of background, it is clear that providing adequate administrative separation between virtual customer environments will be a significant security challenge with elasticity. Cloud providers will need to be prepared to account for and show how their particular services are able to control vulnerabilities such as the earlier example and keep similar vulnerabilities yet

to be discovered from having devastating impacts on their customers. Perhaps more importantly, critical infrastructure could be subject to insurmountable risk and/or loss of sensitive information if providers lack the necessary controls. As services offered from the cloud continue to mature and expand, the threat posed is not limited to unauthorized information access but may include any cloud-provided computing systems (i.e., virtual servers, virtual desktops, etc.). We hope that the U.S. government recognizes and addresses this challenge as federal agencies move rapidly toward adoption of cloud-based services, because the potential consequences are particularly unsettling.

Addressing this challenge may be no small feat. For one, in order for cloud providers to minimize their management costs and obtain profitability, they are expected to have to use shared administrative management systems (i.e., hypervisors) across multiple virtual customer environments. I can envision certain service models where this theory may not hold true: for example, if each customer were given sole hypervisor (or hypervisor-like) management access that connected only to that customer's virtual environment, such as within a virtual private cloud offering. Use of a separate management system for every customer in every service model is probably not realistic simply because of cost.

In researching several cloud providers' capabilities in this regard, I could not clearly see how their solutions could effectively address the entirety of the traversal vulnerability example when multiple customers are using the same hypervisor, at least at the time of writing this article. Although some provide details of built-in software functionality within their hypervisors meant to curtail one customer from gaining access to another's environment, I suspect that these capabilities would not fully address the vulnerability in question. They are certainly worthy of further detailed review.

Another interesting challenge with elasticity in the cloud will be in the ability to provide fine-grained access and predefined security controls across the entirety of a virtual customer environment. The service models to which this might apply most directly are those that provide IaaS and PaaS functionality such as dynamic multilevel security services or multitier application environments. To understand the challenge better, it is probably useful to provide some context for how these types of services are built and administered in today's corporate infrastructure, such as with a multitier application. One typical scenario is where the application development group needs to work closely with the network and hopefully IT security groups to establish proper communication paths among the various tiers, including limiting which network protocols are allowed to interface with each of the tiers. This would be done to ensure proper routing of information and to limit the attack surface available to hackers or malware once the system is put into production.

In addition, when dealing with certain types of data, such as financial or credit card data, certain regulations and industry standards require separation of duties to aid in protection from certain scenarios—for example, an application developer inserting code into software that would allow skimming of financial data and not having an audit trail available as the developer elected not to enable one for obvious reasons. Although various cloud providers do provide some detail on how their solutions address this concern, proper implementation by the user organization, as well as due diligence review of actual capabilities within a desired delivery model, will be critical to ensuring this challenge can be adequately addressed.

Fast forward to the cloud scenario in which a developer now has access to a self-service portal where in a few mouse clicks he or she can build out a new multitier virtual application environment. Without fine-grained access controls available through the self-service portal it will be extremely

difficult to enforce separation of duties to keep this developer from accessing sensitive data he or she shouldn't have access to, or promoting new code to production without having gone through proper security review or change management. In this scenario, the application could be extremely vulnerable to attack or even inadvertently cause a production application to cease operating properly. The ability to implement and enforce access controls to a granular level, defining who has the authority to perform which actions within these environments, will be absolutely necessary.

Having the ability to predefine security control templates may also aid in this sort of scenario. This means that the organization's IT security group is able to define a set of controls that must be applied to a given application depending on the type of data it will be processing or how the application will be used. For example, as the developer builds out the new virtual environment that processes credit-card information, the self-service portal might identify the type of data to be processed and apply predefined security controls to the database, application, and Web front end, as well as predefined firewall rule sets limiting network access to the various tiers. It is unlikely that this capability exists today, anywhere, and we are probably years away from ubiquitous availability.

Another security challenge that develops out of this scenario and in the same vein is how to enforce proper configuration and change management in this more dynamic and elastic model. Even where a portal is capable of granular-access controls that control *which* actions a given user is able to perform, it also needs to enforce *when* and *under what circumstances* a user is allowed to perform certain actions. Without this ability, untested code or system changes could result in business-impacting (or even devastating) results. Even something as "slight" as rolling a new system into production without ensuring that proper server and application patches have been applied could result in significant damage to an organization. Therefore, a mechanism within self-service portals for enforcing an organization's change policies becomes a worthy and necessary capability.

These are but a few of the challenges that come to mind within a truly elastic PaaS and/or IaaS service model, not even delving into separate challenges with SaaS. Other challenges include the ability to provide audit trails across these environments for regulatory compliance and digital forensic purposes; enforcement and awareness of differing levels of zones among development, test, and production environments to protect the integrity of services deployed in the higher-level environments; as well as controlling who is authorized to expand or contract a service within one of these environments. This last concern could pose particular financial issues in the elastic "pay by the drink" service model if, for example, users are able to add services at will and an organization gets a bill at the end of the month for excessive service additions.

Changing tack slightly, however, it is worth mentioning the challenges in providing adequate levels of security services within nonsecurity-related environments. One of these challenges is with traditionally nonsecurity-minded providers needing to supply service options for common security capabilities such as intrusion detection, firewalls, content filtering, and vulnerability testing. In predecessor service models, such as an ASP, these services could be offered through partnerships with security vendors and manually designed and provisioned into the outsourced environment. In the new model, however, how providers are able to provide tighter integration with these services in order not to lose full elasticity may be interesting. It may require creating optional service hooks from a provider's self-service portal to security service products or perhaps developing interesting but complex multiservice cloud models provided by multiple specialty service providers. Either way, this challenge is probably worthy of a discussion in and of itself because of the perceived

number of additional issues it brings to mind. Note that some vendors do offer these capabilities today, particularly within virtual private cloud models, but of the vendors researched, none is fully addressing them for every model it offers.

Encryption for data-at-rest may be an interesting challenge as well. For example, given the previous environment traversal example, file-based encryption within a virtual environment would be essentially worthless in offering protection from remote access. If one can readily gain access to another's environment, this would also provide access to any front-end encryption mechanism used for file-based encryption within the virtual environment. Disk-based encryption becomes particularly challenging because of the nature of virtual storage and potential lack of user organizational control over where data may be physically stored (which disk does one encrypt for a given customer and other constraints in sharing of physical disks among multiple customers). It will certainly be necessary to explore a prospective provider's capabilities for encrypting data-at-rest and how well it addresses the shared concerns, especially for organizations with regulatory requirements dictating the use of file- and/or disk-based encryption.

It should be apparent by now that cloud computing is fraught with a number of security challenges. While some of the concepts and scenarios discussed here are focused on more advanced service models, the intent is to bring a bit more awareness to what the industry will be faced with in moving toward these new models that offer greater levels of "true" cloud computing. Depending on the type of service model being discussed and various use cases, exploring all of the challenges is all but impossible, especially in a single discussion. In addition, some of the security challenges discussed appear to be recognized by certain cloud providers but are primarily being addressed through the use of private cloud models (Amazon and OpSource are two such vendors offering answers within a virtual private cloud offering), suggesting perhaps higher costs versus a public cloud offering and/or limited availability in addressing within other cloud-delivery models.

The promise of what elastic cloud-computing could do for the IT world, however, is extremely invigorating and certainly worth pursuing. It can only be hoped that organizations already taking this path or seriously considering doing so will take the time to fully appreciate the security challenges facing them and whether or not adoption at this point fits into their risk tolerance. Certainly, keeping these and other security challenges in mind while assessing how a prospective cloud provider can address these concerns (and at what cost and with what deployment constraints) should be a critical business objective. Q

LOVE IT, HATE IT? LET US KNOW

feedback@queue.acm.org

DUSTIN OWENS (dustin.owens@bt.com) is a senior principal consultant with BT Americas' Business Innovation Group. He provides consulting services centered on operational risk and security management for multinational customers, specializing in applying these concepts to various areas of strategic business innovation. He has more than 14 years of practical experience in addressing information security within distributed computing environments. He is a graduate of the University of Nebraska at Kearney.

© 2010 ACM 1542-7730/10/0500 $10.00

The NIST Definition of Cloud Computing

Peter Mell and Tim Grance, National Institute of Standards and Technology,
Information Technology Laboratory

Version 15, October 7, 2009

Cloud computing is still an evolving paradigm. Its definitions, use cases, underlying technologies, issues, risks, and benefits will be refined in a spirited debate by the public and private sectors. These definitions, attributes, and characteristics will evolve and change over time. The cloud-computing industry represents a large ecosystem of many models, vendors, and market niches. The following definition attempts to encompass all of the various cloud approaches.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service-provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.

ESSENTIAL CHARACTERISTICS

On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.

Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multitenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

Rapid elasticity. Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

SERVICE MODELS

Cloud SaaS (Software as a Service). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a Web browser (e.g., Web-based e-mail). The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Cloud PaaS (Platform as a Service). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application-hosting environment configurations.

Cloud IaaS (Infrastructure as a Service). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

DEPLOYMENT MODELS

Private cloud. The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on or off premise.

Community cloud. The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on or off premise.

Public cloud. The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.

Hybrid cloud. The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

Note: Cloud software takes full advantage of the cloud paradigm by being service oriented with a focus on statelessness, low coupling, modularity, and semantic interoperability.