

ATOMIC CORPORATE INDUSTRIES

The New IT Environment

Security Challenges in Cloud Computing

2/5/2016

Michael Shinn

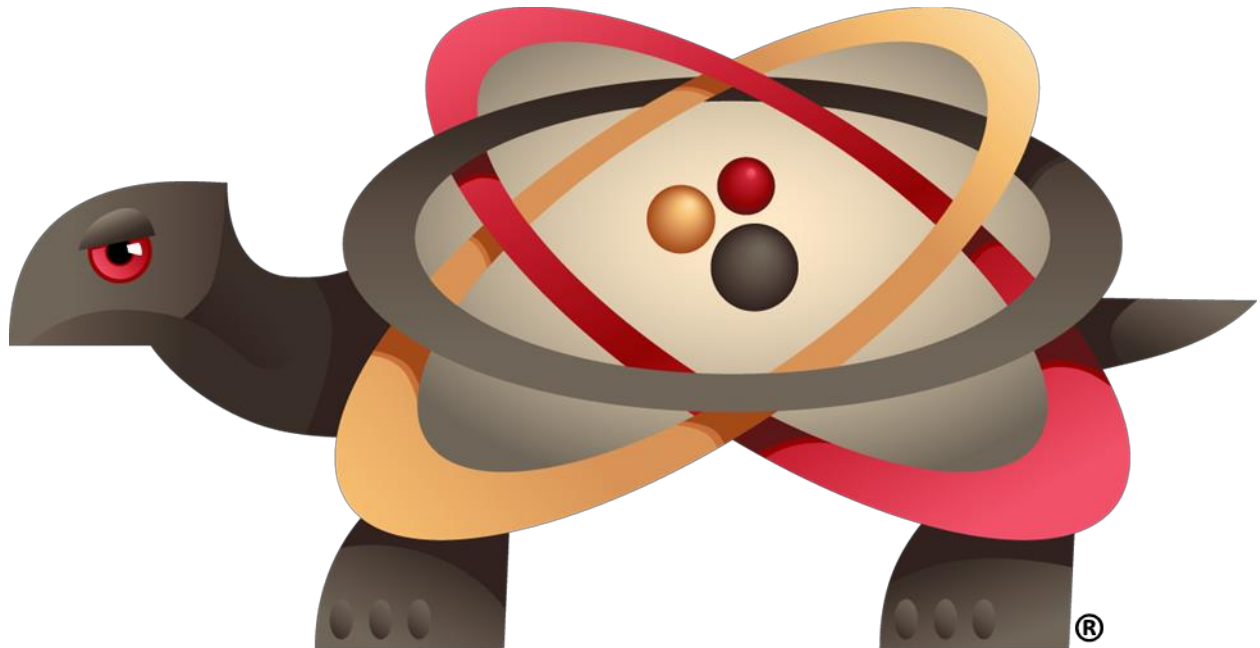


Table of Contents

ABSTRACT.....	2
INTRODUCTION.....	3
THE NEW SECURITY CHALLENGE.....	4
THE ADVERSARY IS SIMPLY BETTER.....	5
CLOUD TECHNOLOGIES.....	6
VIRTUALIZATION TECHNOLOGIES	8
SOFTWARE MOBILITY TECHNOLOGIES.....	9
CONCLUSION.....	11

Abstract

Many organizations are turning to cloud computing, system virtualization, container-based application virtualization, network virtualization and Infrastructure-as-a-Service (IaaS) to deliver their computing and application requirements. These platforms require the convergence of cross-functional cloud teams—developers, technology operations, and security.

Whether an organization is already using or currently evaluating cloud, container or virtualization technologies, these teams need agile methods to meet security requirements that work with these new technologies. They need to address new threats and work within these dynamic and fundamentally different IT architectures of application development and deployment.

Coupling these non-trivial IT shifts with more advanced adversaries that continue to bypass traditional, perimeter-based security devices and inherited security controls requires a more resilient security approach.

Introduction

This white paper provides technology and security professionals with real-world advice on addressing security in the cloud, virtual and containerized environments. It includes guidelines on how security applies to these technologies; information on the shared security responsibility across providers, security and clients; and strategies to maximize automation of security and compliance operations using a single approach across physical, virtual, and cloud servers.

The New Security Challenge

There are four key challenges driving significant change for the security practitioner:

- 1) The adversary is simply better; they are getting past the perimeter, defeating protection solutions and bypassing inherited security controls for the assets those approaches are meant to protect
- 2) Cloud technologies remove the traditional control points at the perimeters, and adjacent systems and cloud providers expect end users to "bring their own security"
- 3) Virtualization technologies, micro-segmentation and Software Defined Networks are creating dynamic pathways, eliminating traditional perimeters and thereby creating new pathways into the enterprise
- 4) Software mobility technologies, such as Docker, make it difficult to control the quality and security of software entering and moving throughout the enterprise

The Adversary is Simply Better

2015 was yet another year where cyber adversaries demonstrated improved capabilities to defeat cyber defensive techniques, including major compromises of government agencies. The Government Accountability Office recently noted that the number of information-security incidents affecting systems supporting the federal government grew 1,121% since 2006 -- 5,503 incidents in 2006 to 67,168 in fiscal year 2014. Similarly, the number of information-security incidents involving just PII reported by federal agencies has more than doubled in recent years, from 10,481 in 2009 to 27,624 in 2014.

These realities, combined with an increase in state-based actors, including weaker adversaries that are also attracted to cyber warfare, has created a climate where we can only expect the adversary to get better and the pace of attacks to continue to grow. Weaker adversaries also present a new and troubling shift in attacker dynamics. We are now seeing bolder attacks from these weaker adversaries, targeting areas that stronger adversaries have traditionally avoided due to a desire to prevent escalation. The lack of respect based upon mutual destruction in cyberspace is especially apparent when the threat actors' own military and political apparatus does not rely on networks and cyber capabilities. This becomes even more pronounced when vulnerabilities are asymmetric, and one side is heavily reliant on cyberspace for military, economic, and political activity whereas the other side barely uses them. In these cases, the weaker adversary is more likely to carry out cyber warfare as the prospect of losing its own capabilities in retaliation is minimal. The future is not good for the cyber defender.

Cloud Technologies

According to Gartner, the naive belief that cloud providers are totally responsible for their customers' security discourages organizations from ensuring cloud services are used appropriately and securely, and through 2020 Gartner states that 95% of cloud security failures will be the customer's fault.

Organizations that haven't taken a strategic approach towards security within cloud computing can easily find themselves in a far less secure position than they had with traditional computing, resulting in unnecessary compliance incidents, compromises, and data losses.

While the parts of the stack under the responsibility of Communications Service Providers (CSPs) are generally secure, the characteristics of the parts of the cloud stack under the control of clients will not prevent naive users from implementing poor security practices, which can easily result in security or compliance failures.

The safe use of all forms of cloud technologies requires new organizational policies, skills and activities. No technologies — on-premises or in the cloud — can ever be considered 100% secure or reliable — especially when users and IT staff are provided a new capability with no guidance on its use or management. Maximizing cloud benefits means carefully governing your organizational practices for IaaS, PaaS, and SaaS to avoid introducing security, compliance or regulatory exposures on top of what would otherwise be robust computing platforms.

Data within cloud infrastructure is often more accessible to unauthorized people than that same data would be within traditional data centers. Even if sensitive data is not actually exposed through confidentiality failure, it may be desirable to further ensure that through the use of security control technologies. Reliable identity — not just of people, but increasingly of things and processes — is a crucial building block for the secure use of distributed components accessed remotely.

Shared environments, especially within the cloud, represent a challenging network security environment. IaaS vendors only offer relatively non-granular firewall mechanisms that stop at Layer 3, and their customers have no ability to place a physical firewall within their cloud provider's environment. This means that cloud users that want more granular or application-aware network controls must use software-defined mechanisms.

Therefore, for cloud-based systems, customers must collapse security controls into their cloud servers and containers to compensate for non-granular firewall mechanisms that stop at Layer 3. This action suddenly requires systems to implement their own layer 7 firewalls (for example, WAFs) on the servers and within the containers themselves. Additional technologies must also be implemented on endpoints in order to compensate for the weaker capabilities of non-granular level firewalls, as this increases both the attack surface for the servers and containers, and thereby increases the likelihood of compromise.

To that end, servers and containers will also need to have their own self-defense and self-healing mechanisms to both prevent intrusions and inevitably enable automatic removal or mitigation of the effects of a compromise. When combined with software mobility technologies, such as Docker, the need for these technologies becomes more pronounced and is discussed further below.

Virtualization Technologies

Placing any workload in a virtualized environment, or taking advantage of applications through virtualization requires modified or new security controls and a renewed understanding of the updated risks introduced by the IT architecture.

VMs use a layer of abstraction (the hypervisor) that mediates requests between the multiple VMs being hosted and the hardware being shared underneath. The virtualization layer enforces separation between network, computer and storage resources of multiple VM tenants. However, vulnerabilities in the hypervisor can potentially expose a guest VM to attacks from another guest VM by allowing execution of code in the hypervisor host system.

Significant vulnerabilities have been publicly disclosed in major virtualization technologies within recent months. Notably, CVE-2015-7835, which affected Xen service providers such as Amazon AWS, laid unpublished for over seven years. Combined with the incredible complexity of these technologies, the probability of additional yet undiscovered vulnerabilities is increasingly higher. Therefore, systems that host virtual machines must implement both kernel-level and system-level protections to both prevent and respond to virtual machine guest escapes.

An added complication due to virtualization technologies is what is sometimes referred to as “east/west” traffic. That is data moving within shared infrastructure between one or many organizations, as opposed to through its external access points to their final destination. This “east/west” traffic is outside the view of access points, firewalls and even network choke points because these flows are either within a single system, as with entirely virtual machines, or are simply impossible to see because choke points can not be introduced into the environment without defeating the purpose of software defined networks.

Software Mobility Technologies

In contrast to virtualization, containers share a common OS, so it is the OS that is providing isolation, not the hypervisor. Without the implementation of additional security controls, network access is available between all hosted containers sharing the same OS; thus, a successful attack on the OS kernel layer exposes all containers. Therefore, it is vital that containers that host the OS be particularly well protected from compromise and have the ability to recover from compromises with a high level of assurance.

A dynamic and freewheeling DevOps environment often takes the approach of "build first/fix later." Containers have become a popular and growing mechanism to enable this new freewheeling DevOps approach to software development. Containers, such as Docker, enable OS kernel and some library sharing while maintaining separate application spaces when necessary, with the promise of providing predictable application compatibility for developers. Multiple applications can share an OS, even multiple versions of the same application, and the container model enables a package for easy and consistent provisioning of applications to OSs.

Containers are most frequently consumed in conjunction with Docker; Docker makes containers user-friendly to developers, and most use of containers will be to help developers be more agile, particularly when they are building frequently updated applications that need consistency and predictability through the full application life cycle. Note that Docker does not create a standard level of container security, which means that not only is the pace of change in these environments substantially faster than with any other technology, but unlike VMs, containers inherit vulnerabilities in the host platform as well as introduce vulnerabilities that do not exist with VMs.

For example, containers, even when run “de-privileged,” do not provide protection from attacks on the underlying shared kernel vulnerabilities or vulnerabilities that result in privilege escalation. Using an analogy, VMs are apartments in a shared building, and containers are roommates in the same apartment.

Combining the lack of a “Dockerized” level of container security with the reality of very rapid deployment and application changes, the impracticality of expecting operations managers to ensure Docker and machine images are up to date and free of vulnerabilities creates an environment rife with software and configuration vulnerabilities. The addition of Public repositories for Docker applications increases the risk. Practitioners must assume that there will be vulnerable Docker containers (intentional or otherwise) and software in their environments and build security controls into the host platforms to address these vulnerabilities and potential compromises.

Therefore, any use of containers will require strong separation protection at the kernel level, additional controls on the host OS to prevent containers from affecting containers on the same system, and, most importantly, application immunization against weaknesses as described below. This protection requires an end-point solution on the host OS that supports containers and provides per-container visibility and policy enforcement. Containers and VMs with higher assurance levels will also need agentless anti-malware scanning, agentless file integrity monitoring, and agentless in-line network intrusion prevention on the Host OS.

Conclusion

Immunize the System

As described above, containers bring unique and challenging vulnerabilities into the environment, and tools that attempt to detect vulnerabilities in libraries and applications in containers rely on either the vulnerability to be known and public, which means the vulnerabilities are discovered after the fact, or static and dynamic analysis. Static analysis is only possible if the source code for all libraries and applications are available before the application is deployed, and a manual code review is performed as a result of the static code analysis. The former, access to all source code in the container, is essentially impossible. Commercial and closed source software is rarely made available with the source code, and confirming that a binary object, such as a library, maps to a specific source repository is also difficult and sometimes impossible. Vendors will sometimes “back port” changes to code to older versions of libraries, making the actual source code in the library or application very different from the version of the application or library. Dynamic analysis is also prone to false negatives due to the exponential nature of all branching options within an application. And finally, if cyber security has taught us anything, it’s that everything essentially has vulnerabilities in it; most of them are simply not known to the user community yet.

The solution to this vexing problem is addressing the specific weaknesses that these vulnerabilities take advantage of and removing the weaknesses from the system, thereby making these vulnerabilities irrelevant. For example, if a system can be made immune to buffer overflow attacks, then it does not matter if an application or a library has a buffer overflow vulnerability. This proactive approach to security is scalable, there are a finite number of weaknesses, it addresses zero-day threats fundamentally, and most importantly for container based software development, it neither slows down the process of development, nor does it place impossible demands on developers. This is the cyber security equivalent of making the system immune to whole types of disease, making it unnecessary for the system to not be exposed to these diseases.

How to solve the Problem

New forms of software-defined security mechanisms, such as Atomic Secured Linux and Atomic Secured Docker, are becoming practical mechanisms for the maintenance of system and network security within highly dynamic virtual and containerized environments that deliver all of the capabilities described above. These solutions start by immunizing the system from classes of weaknesses, which helps to eliminate vulnerabilities in applications, virtualization, and container-based technologies by default. They also address the need to deliver agentless anti-malware scanning, agentless file-integrity monitoring, and agentless in-line network intrusion prevention on the Host OS to cope with both scalability issues and the lack of granular layer 7 firewall capabilities within cloud environments. These technologies also work unobtrusively with the freewheeling nature of DevOps and the promises of innovation that container-based technologies deliver.

Additionally, Atomic Secured Linux is designed to integrate with both Continuous Integration (CI) environments, where the application developer can vet changes to the application at the time the change is submitted, rather than weeks or even months later. Atomic Secured Linux in a CI environment allows for automated testing to occur with not only the security controls active as part of the utility testing, but also to allow Atomic Secured Linux to provide feedback to the Developer in near real time if the application is creating a vulnerable condition. This is a unique capability as this provides actionable information during the development process for more meta-type vulnerability identification to the developer(s) that are best positioned to understand and remediate the issue. This level of security automation supports the development process by reducing the number of tasks that could otherwise overburden those developer(s) and lead to compounding mistakes.

For questions regarding this white paper, please contact Cliff Richardson at Atomicorp

By phone at 703-299-6667 or email cliff@atomicorp.com

For additional information visit the company website: www.atomicorp.com