SELF-AWARE DEPLOYMENT ENFORCEMENT OF VIRTUALIZED AND CLOUD-BASED IMAGES

Ethan Hadar[1], Amir Jerbi[1] and Irit Hadar[2]
[1]*CA Technologies, Inc., Herzelia, Israel*
[2]*Department of Information System, University of Haifa, Haifa, Israel*

Keywords: Cloud Virtualization, Infrastructure as a Service, Cloud Security Vulnerabilities, Privileged Access Management.

Abstract: This position paper presents our approach for deployment enforcement of virtual images (VM), in order to prevent an unauthorized usage, potential insider threat, and theft of VMs. In existing systems, regular images in a virtual environment can be mounted and installed in a different location, while our system prevents the intentional and unintentional roaming of these images, triggered by either humans or automation tools. This paper proposes an approach that secures installation location according to policy in virtualized environments, by intercepting the image installation process.

1 INTRODUCTION

In the domain of cloud virtualization (Gurav and Shaikh, 2010), virtual machines (VM) and configured appliances are used to drive the dynamic capacity of the infrastructure, facilitate configuration, and enable the elasticity of roaming systems that lever the abstraction of the physical location. This abstraction enables VMs to be active in one virtual environment, suspended, and then roamed to another virtual environment where they can resume the exact state. When the VM is moved, it is handled as a regular file. These roaming capabilities enable capacity bursting as well as improved performance in terms of energy or physical CPU utilization. Such value propositions created by the VM abstraction of the physical server layers introduce new security vulnerabilities (Blum et al., 2011; Heiser and Nicolett, 2008; Kresimir and Zeljko, 2010) , such as: (1) the ability of non-privileged administrators to access data and the VM when the image is in dormant state; (2) theft of the VM by copying and activating it in a non-approved environment; (3) the ability to disrupt compliance and privacy (Pearson, 2009) regulations unintentionally by activating the image (VM) in a non-monitored environment.

This position paper proposes an approach that precludes these virtual environment security vulnerabilities (Lombardi and Pietro, 2011; Mather et al., 2009) using a self-aware security deployment enforcement system. The enforcement policy is aimed primarily at the prevention of theft or unauthorized roaming of virtualized and cloud-based images. Moreover, this paper suggests a combination of a centralized enforcement approach, and the VM self-aware security capabilities of deployable image.

The proposed system addresses several security vulnerabilities and challenges:

- Prevention of theft of images.
- Governing the compliance of an authorized deployment within the enterprise boundaries.
- Rapid feedback in the case of suspected non-authorized deployment.
- Indication of and alerts of a potential insider threat.
- Optional self-protection and self-destruction of the data and image upon detection of non-authorized usage.
- Increased efficiency of deployment of approved images, which expedites the synchronization of the latest policies based on level of importance. Namely, separation of deployment enforcement from other policies for privileged users that are updated based on the normal security process.
- The ability to certify secured locations for images to be deployed centrally, while, however, enforcing the actual deployment on a standalone, self-aware mechanism.

Hadar E., Jerbi A. and Hadar I..
SELF-AWARE DEPLOYMENT ENFORCEMENT OF VIRTUALIZED AND CLOUD-BASED IMAGES.
DOI: 10.5220/0003912906020605
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 602-605
ISBN: 978-989-8565-05-1

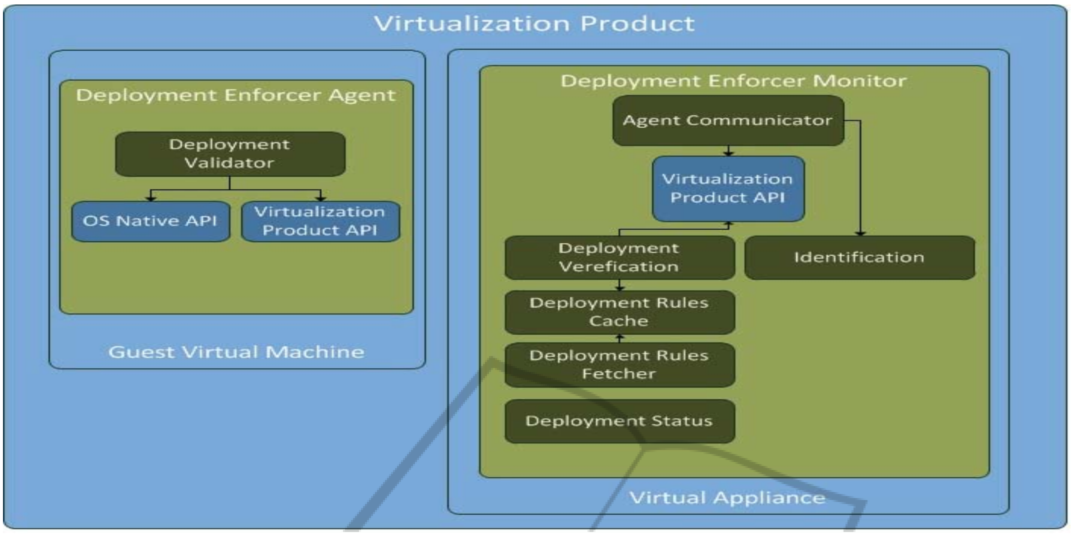

Figure 1: Conceptual architecture.

- Enabling a real-time prevention and visibility into the deployed image.

2 CONCEPTUAL ARCHITECTURE

The proposed system extends existing Privileged Access Management solutions, with additional security enforcement capabilities. These capabilities relate to deployment and installation of VM on top of virtualized, such as VMware, or hosted, such as on Amazon EC2, environments.

This section describes the fundamental components relevant to this position paper. Yet, it extends a basic use case of Privileged Access Management for centralized security control. The basic use case provides a centralized definition of security policies according to compliance, regulation, or any other security need of the enterprise. The centralized system sends security policies to security agents that are deployed on the managed servers (VMs or physical servers). The agents interpret every administrator command, and either permit or deny the execution of that command, with appropriate reporting and auditing.

2.1 System Structure

Figure 1 depicts our incremental conceptual architecture, where components in green represent new components and . Ccomponents in light blue represent virtualization vendor's and native operating system (OS) components. Components in blue represent the native operating system (OS) components. The components are:

Deployment Enforcer Agent - This component is installed on guest virtual machines (VM). The agent validates that the guest VM is working on a verified environment by checking the existence of a trusted *Deployment Enforcer Monitor* on one of the virtual machines running on the *Virtualization Product*.

Deployment Validator - Communicates with the *Deployment Enforcer Monitor* and validates whether it is trusted. Communication with *Deployment Enforcer Monitor* is through using the virtualization product API (provided by the vendor). The *Deployment Validator* uses OS native API to shut down an image if cannot locate the trusted *Deployment Enforcer Monitor.*

Deployment Enforcer Monitor - This component is installed on a dedicated virtual machine (*Virtual Appliance*) and deployed on the *Virtualization Product* (such as Vmware ESXi, Microsoft Hyper-v or other vendors). It communicates with *Deployment Enforcer Agents*, installed on guest VMs running on the same *Virtualization Product* host, and provides proofs that the environment is safe.

The *Deployment Enforcer Monitor* validates that guest VMs running on the *Virtualization Product* are in compliance with enterprise policies. If they are not compliant the *Deployment Enforcer Monitor* prevents a guest VM from running. The *Deployment Enforcer Monitor* inner components are:

- **Agent Communicator** – Responsible for communicating with *Deployment Enforcer*

Agents, installed on guest VM running on the same *Virtualization Product* host where the *Deployment Enforcer Monitor* is. *Agent Communicator* uses the virtualization product API to communicate with *Deployment Enforcer Agents*.

- **Identification** – Responsible for supplying identification information of the *Deployment Enforcer Monitor* to *Deployment Enforcer Agents*.
- **Deployment Verification** - Verifies whether guest VMs deployed on the *Virtualization Product* are compliant with the enterprise security policy.
- **Deployment Rules Cache** – Maintains a local cache of enterprise deployment policies.
- **Deployment Rules Fetcher** - Fetches enterprise deployment policies from the centralized enterprise security management server. The enterprise security management server is an existing component that contains deployment policies as defined by a Privileged Access Management administrator.
- **Deployment Status** - Sends deployment status information of VMs running on the virtualization product to the Access Control Enterprise Management Server.

2.2 Prototypical Scenarios

In order to exemplify the conceptual architecture, several scenarios (use cases) are presented.

2.2.1 Activation Scenario – Theft Prevention

Every time an administrator tries to load a guest VM on the *Virtualization Product*, the *Deployment Validator* (which is a component of *Deployment Enforcer Agent*) sends a request for environment verification from the *Deployment Enforcer Monitor* (installed on a virtual appliance on the same *Virtualization Product*). The *Agent Communicator* component receives this request and passes it to the *Identification* component. The *Identification* component replies back through the *Agent Communicator* component with information which uniquely identifies the *Deployment Enforcement Monitor*. The *Deployment Validator* component receives the information and validates whether it is trusted. When the validation fails, it will prevent the VM from becoming active; when validation is successful, it will allow running the virtual machine.

2.2.2 Activation Scenario – Governance and Compliance

Deployment Rules Fetcher (a component of *Deployment Enforcer Monitor*) receives enterprise deployment rules from the existing Enterprise Management component. Enterprise deployment rules are saved on the *Deployment Rules Cache* component. *Deployment Verification* component validates that the deployed VMs are compliant with the enterprise deployment rules. In the case of deviation, the *Deployment Status* component reports to the Enterprise Management security server with the deviation information.

2.3 Prototypical Implementation

Figure 2 describes the implementation of our solution in a VMware environment that has several virtualization products, such as VMware ESXi.

The ESXi virtualization product can run many VMs on it, such as Windows and Linux VMs. The system administrator installs *Deployment Enforcer Agents* on these VM. The administrator deploys *Virtual Appliance VM* with the *Deployment Enforcer Monitor* on the ESXi virtualization product system.

2.3.1 Use Case 1: Theft Prevention

When a VM starts the *Deployment Enforcer Agent* that is installed on it, it verifies the existence of a trusted *Deployment Enforcer Monitor* on the ESXi *Virtualization Product*. Different methods can be used for verification, for example, validating that the *Deployment Enforcer Monitor* passes a certificate (token) issued by a trusted source. If the *Deployment Enforcer Agent* fails to validate the *Deployment Enforcer Monitor*, it will not allow the virtual machine to be started. This act prevents theft of virtual machines.

2.3.2 Use Case 2: Governing the Compliance of Deployment within the Enterprise Boundaries

The *Deployment Enforcer Monitor* receives the enterprise's deployment rules from the Enterprise Management security component. The *Deployment Enforcer Monitor* validates that every image deployed on the local VMware ESXi server is compliant with the deployment rules it fetched from Access Control Enterprise Management. An example of such validation is that *Deployment Enforcer Monitor* can check that all VMs are compliant with the enterprise deployment policy that

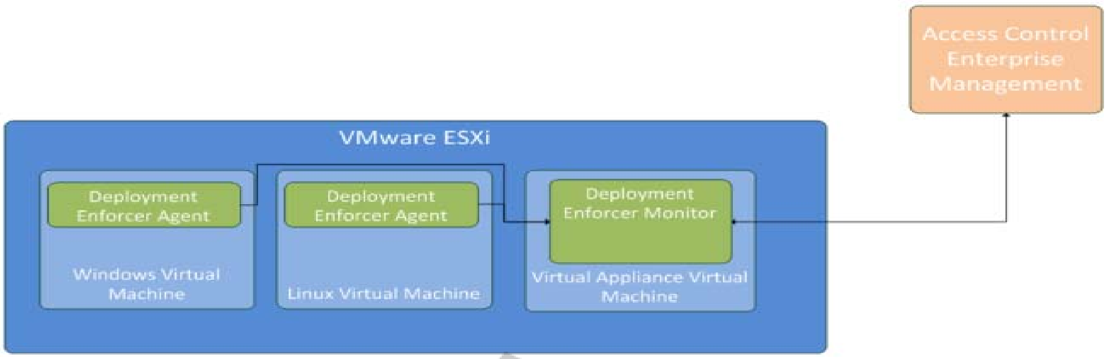
Access Control Enterprise Management
VMware ESXi
Deployment Enforcer Agent
Windows Virtual Machine
Deployment Enforcer Agent
Linux Virtual Machine
Deployment Enforcer Monitor
Virtual Appliance Virtual Machine

Figure 2: Deployment enforcer implementation for VMware.

requires that on same ESXi there should not be a mixture of Windows and Linux VMs. *Deployment Enforcer Monitor* can notify Access Control Enterprise Management about deviation from this requirement.

3 DISCUSSION AND CONCLUSION

This position paper addresses security vulnerability issues in virtualized cloud environments, where the scale-up, bursting scenario of virtual images (VM) is common. In such environments, an IT management service should provide an agile response: the deployment of a VM by the centralized managing tool, supporting rapid security responses. In these virtual environments, the VM can be stored in a managed repository, or extracted and installed in different unauthorized location, exposing it to intentional theft of data.

The presented approach offers a way to separate the deployment security needs from other security needs in order to increase efficiency of deployed images that are controlled in terms of location only. Furthermore, when a VM is mounted in an external or un-authorized virtualization environment, the system prevents the installment of the protected image, thus, preventing theft or insider threat.

As a result, the presented system prevents the intentional and unintentional deployment and consequent activation of a virtual image. Our approach secures the enterprise’s recommended pre-built VMs that cannot be accessed except with the approval and verification of the organization policy. Consequently, our solution addresses a new form of security vulnerability in the virtualization domain according to compliance and regulations needs.

REFERENCES

Blum D., Schacter P., Maiwald E., Krikken R., Henry T., Boer M., Chuvakin A., 2011. 2012 Planning Guide: Security and Risk Management, G00224667, *Burton IT1 Research*, 1 November 2011.

Gurav U., Shaikh R., 2010. Virtualization: a key feature of cloud computing. *ICWET '10 Proceedings of the International Conference and Workshop on Emerging Trends in Technology*. Mumbai, India — February 26 - 27, 2010.

Heiser H, Nicolett M, 2008. Assessing the Security Risks of Cloud Computing. *Gartner Research Report* G00157782, 3 June 2008.

Kresimir P., Zeljko H., 2010. Cloud computing security issues and challenges, *MIPRO, 2010 Proceedings of the 33rd International Convention*, Opatija, Croatia, 24-28 May 2010.

Lombardi F., Pietro R. D., 2010. Secure virtualization for cloud computing. *Journal of Network and Computer Applications*, Volume 34, Issue 4, pp. 1113-1122.

Mather T., Kumaraswamy S., Latif S., 2009. Cloud security and privacy: an enterprise perspective on risks and compliance. Book, publisher *Sebastopol, CA: O'Reilly*.

Pearson, S., 2009. Taking account of privacy when designing cloud computing services. *ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD '09.* Vancouver, BC, 23-23 May 2009