Hyperdrive: a flexible cloud testbed for research and education

Amirali Sanatinia*, Sanket Deshpande, Apoorv Munshi, Daniel Kohlbrenner,
Michael Yessaillian, Sarada Symonds†, Agnes Chan, Guevara Noubir‡

College of Computer and Information Science
Northeastern University, Boston, USA

*amirali@ccs.neu.edu, †{deshpande.sa,munshi.a,kohlbrenner.d,
yessaillian.m,symonds.s}@husky.neu.edu, ‡{ahchan,noubir}@ccs.neu.edu

***Abstract*—Today, cloud computing is of paramount importance and is widely used. There have been many works illustrating the potential of narrowly focused security attacks against cloud systems, as well as elaborate mitigation techniques. However, the understanding of the implications of these attacks and mitigations on practical cloud systems as a whole, remains limited. One of the reasons is the difficulty and cost of control of the environment, reproducibility of experiments, and flexibility in reconfiguration to different cloud systems. In this work we present "hyperdrive", a highly reconfigurable cloud testbed to support research and education in experimentally assessing the practical impact of attacks and mitigations on realistic cloud systems. The testbed can be used for research activities to evaluate the performance of mitigation techniques spanning crypto-based privacy-preserving cloud computation, and applied system security research focusing on side channel attacks, and stealthy data exfiltration. The proposed testbed can also be used for education within the context of the cloud security course and projects.**

I. Introduction

The goal of setting up a highly reconfigurable cloud testbed is to support research and education in experimentally assessing the practical impact of attacks and mitigations on realistic cloud systems. There have been many papers illustrating the potential of narrowly focused security attacks against cloud systems, as well as elaborate mitigation techniques [1]. However, the understanding of the implications of these attacks and mitigations on practical cloud systems as a whole, remains limited. To illustrate this point, there has been proposals for many crypto-based solutions to computation over encrypted data that focus on a specific operation (e.g., searchable encryption) and do not necessarily combine well with the requirements for other operations (e.g., counting patterns). The testbed can be used for research activities and education in the context of cloud security course. It practically enables the deconstruction and reconstruction of cloud platforms at all layers of the cloud system stack spanning IaaS, PaaS, and SaaS.

Deconstructing and reconstructing the cloud: the flexible reconfiguration of the testbed stack enables the students to develop a precise understanding of the subtleties and intricacies of each component of the cloud stack and the performance and security pitfalls of the various combinations.

Cloud Security Hands-On Exercises: hyperdrive can be used to develop laboratories that cover a wide range of cloud security attacks and defenses. The cloud testbed can be instrumented to enable a fine grain assessment of the attacks potential and performance. Laboratories can include reproducing recent research results. Illustrative examples of the labs include the construction of an access-driven side channel attack that enables a malicious VM to extract fine-grained information from a victim VM running on the same physical computer. In particular the students can experiment with a malicious VM extracting sensitive information from simple traffic/computation patterns to El-Gamal keys (used with the libgcrypt library) [2]. A second example, at the end of the spectrum of cloud security attacks, consists of analyzing the performance of crypto-based privacy-preserving operations such as the most recent Oblivious RAM [3] techniques, Private Information Retrieval, searchable encryption [4], or private stream search [5]. The testbed plays an instrumental role in enabling the secure data analysis research, outlined in the earlier section, by enabling a fine grain evaluation of the performance of the proposed privacy-preserving schemes under different configurations, traffic loads, and a variety of direct and side channel attacks.

II. Architecture

In the following, we provide a description of the choices for the hardware and software key layers of the flexible cloud testbed. The selection of hardware and software is the outcome of the analysis of necessary services required for enabling the research and education components, and the capabilities of the various cloud stacks software. We intended to maximize the usage of Open Source software, except when the goal of an experiment is to analyze a specific commercial cloud system. Figure 1 depicts a reference stack architecture for a cloud tested.

A. Hardware and Virtualization Layers

Hardware Layer: The testbed runs on a combination of bare-metal hardware and virtualized systems. The configuration and administration dashboard runs on the virtualized machines where they control the servers to be imaged. The servers are interconnected through Gigabit ethernet switches for both a private/control network and remote access. The servers support a Tamper Proof Module (TPM) and support Intel Trusted Execution Technology (TXT).

978-1-5090-6356-7/17/$31.00 ©2017 IEEE

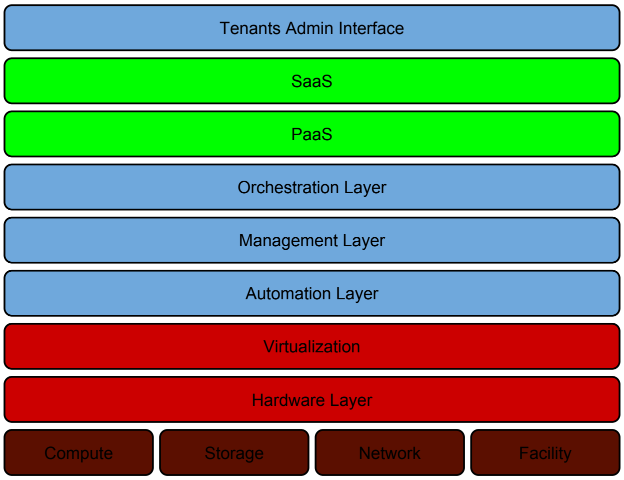


Fig. 1: A reference stack architecture for a cloud tested.

Virtualization Layer: Currently we support Xen [6] and KVM [7], two of the major hypervisors in use for private and public clouds. In the future we are planning to add support for ESXi (VMware) [8] and Hyper-V (Microsoft) [9]. One of the unique characteristics of the testbed is that it enables a fast and automated reconfiguration of the cloud testbed to run a user specified cloudstack suitable for the considered experiment. We use the Preboot eXecution Environment (PXE) [10] to develop a reimaging framework similar to the ORBIT Management Framework [11], and Emulab Control Framework [12].

B. Infrastructure as a Service (IaaS) Fabric Management Layers

An Infrastructure as a Service cloud is typically built as a stack of three layer: the automation, management, and orchestration layers. They span from the finest granularity to the most general of infrastructure processes automation. First, the Automation Layer provides an interface to the lowest hardware/virtualization layers, then the Management Layer is typically responsible for the servers lifecycle management operations (e.g, patches, deployment, monitoring), and finally the Orchestration Layer which enables the end-to-end cloud infrastructure process automation.

For these layers of the testbed, we are using both Open Source and commercial stacks. We leverage the PXE capabilities to reimage the testbed nodes and reconfigure under different IaaS stacks, therefore enabling a versatile education and research testbed. For instance, Apache CloudStack, and OpenStack, are open source software for IaaS. They are designed to deploy and manage large networks of virtual machines, with high availability and scalability in mind. They can be used for building public, private, and hybrid clouds. They include the entire ”stack” of features most organizations want with an IaaS: compute orchestration, Network-as-a-Service, user and account management, a full and open native API, resource accounting, and a good User Interface (UI). They support the hypervisors we are using: KVM, and Xen (also VMware, and Hyper-V for future). Furthermore they allow users to manage their cloud with a Web interface, command line tools, and/or a full-featured RESTful API. CloudStack also provides an API that is compatible with Amazon’s cloud AWS EC2 and S3 for building hybrid clouds with a footprint on Amazon. This would be useful for comparing performance to AWS but also to enable realistic hybrid clouds. We will also consider commercial cloud stacks such as VMware vCloud/vSphere and Eucalyptus for both the purpose of education, and assessment of attacks and performance across cloud systems.

C. Platform as a Service (Paas) and Software as a Service (SaaS) Layers

For the highest layers of the testbed, we can support several systems including Apache’s Stratos, RedHat’s OpenShift, Cloudify, and Cloud Foundry. We can also support Apache’s Hadoop (the open source version of Google’s MapReduce) as it is a key paradigm and platform for developing big data applications and also plays a central role in our private data analysis research activities.

D. Implementation

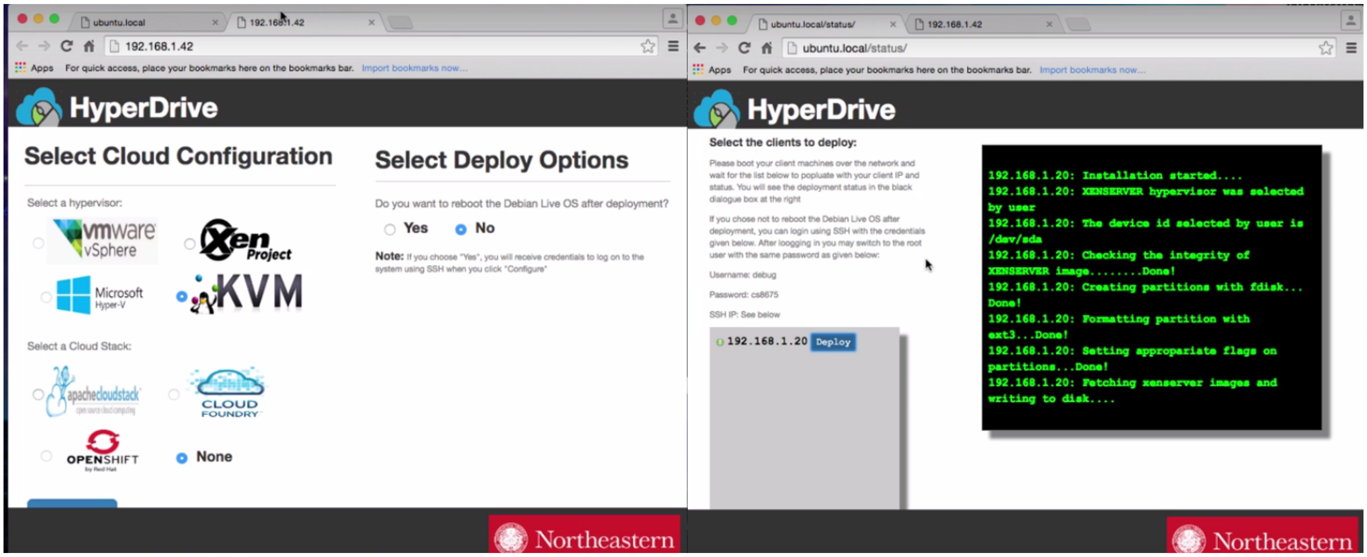
Hyperdrive consists of a management server which is in charge of the deployment of hypervisor and the customized images over the platforms. The management server contains a DHCP server, TFTP Server, HTTP Server, MQTT Server, cloud images and a minimal live Linux OS containing the deployment shell script. The management server hosts a web application for the user to select the cloud configuration and options to deploy on client systems. The shell script and the web application communicate with each other using the MQTT protocol to indicate when client machines are online and also to send diagnostic messages. When the user selects the desired cloud configuration on the web application, he needs to boot the client machines. The client machines boot over the network using Preboot Execution Environment (PXE) and fetch the Debian live OS. The Debian OS boots up and executes the shell script to deploy the cloud and send diagnostic messages to the web application. In the boot process, we use iPXE, which is a full PXE implementation enhanced with additional features such as, boot from a web server via HTTP and control of the boot process with script. Furthermore, to address the security concerns of BIOS and attacks against legacy boot procedure [13], we use the Unified Extensible Firmware Interface (UEFI) secure boot.

We evaluated our system on a Dell Power Edge R710. We deployed different set of configurations using KVM, Xen at the hypervisor level and Apache Cloud stack framework. Figures 2a and 2b depicts snapshots of the Hyperdrive in action for deployment of a cloud system configuration. We made Hyperdrive open source and publicly available at `http://hyperdrive.ccs.neu.edu/` to the benefit of the community. The documentation and a video demo of the system in live deployment is also available.

III. Related Work

Both the industry and academia have been working on tools and frameworks for assisted infrastructure setup and configuration. For example, Emulab Control Framework [12] is a network testbed that allows researchers to develop, debug, and evaluate their systems. It is a group of installation that provides nodes for experiments in the field of networking and

978-1-5090-6356-7/17/$31.00 ©2017 IEEE



Select Cloud Configuration

Select a hypervisor: ○ vmware vSphere ○ Xen Project ○ Microsoft Hyper-V ● KVM

Select a Cloud Stack: ○ apachecloudstack ○ CLOUD FOUNDRY ○ OPENSHIFT ● None

Select Deploy Options

Do you want to reboot the Debian Live OS after deployment?

○ Yes ● No

Note: If you choose "Yes", you will receive credentials to log on to the system using SSH when you click "Configure"

(a) Menu

Select the clients to deploy:

Please boot your client machines over the network and wait for the list below to popluate with your client IP and status. You will see the deployment status in the black dialogue box at the right

If you chose not to reboot the Debian Live OS after deployment, you can login using SSH with the credentials given below. After loogging in you may switch to the root user with the same password as given below:

Username: debug

Password: cs8675

SSH IP: See below

192.168.1.20

```
192.168.1.20: Installation started....
192.168.1.20: XENSERVER hypervisor was selected
by user
192.168.1.20: The device id selected by user is
/dev/sda
192.168.1.20: Checking the integrity of
XENSERVER image........Done!
192.168.1.20: Creating partitions with fdisk...
Done!
192.168.1.20: Formatting partition with
ext3...Done!
192.168.1.20: Setting appropariate flags on
partitions...Done!
192.168.1.20: Fetching xenserver images and
writing to disk....
```

(b) Status

Fig. 2: Snapshots of the Hypervisor in deployment. After the user chooses the configuration [left], the management server start deploying the images over the platforms.

distributed systems. Another example is CloudLab [14], which is based on Emulab and Global Environment for Network Innovations (GENI) to allow users to control the provisioning of cloud stack.

Most of the currently available cloud configuration tools do not provide a platform to image the bare hardware compared to hyperdrive. For example, tools such as Chef and AWS OpsWorks faciliates the provision of cloud platform after it is deployed, while hyperdrive facilitates the deployment of virtualization layers such as Xen and KVM over hardware. Furthermore, hyperdrive uses open source software as much as possible for higher layers (Figure 1

IV. Conclusion

Hyperdrive, provides a flexible testbed to develop a precise understanding of the subtleties and intricacies of each component of the cloud stack and the performance and security pitfalls of the various combinations. Furthermore, it allows the design and development of laboratories that cover a wide range of cloud security attacks and defenses. It is instrumented to enable a fine grain assessment of the attacks potential and performance. Additionally, it enables the reproducing of the recent research results. Illustrative examples include the construction of an access-driven side channel attack that enables a malicious VM to extract fine-grained information from a victim VM running on the same physical computer. For example, it allows experiment with a malicious VM extracting sensitive information from simple traffic/computation patterns to El-Gamal keys (used with the libgcrypt library) [2]. A second example, at the end of the spectrum of cloud security attacks, consists of analyzing the performance of crypto-based privacy-preserving operations such as the most recent Oblivious RAM techniques, Private Information Retrieval [15], searchable encryption [4], private stream search [5], and exploriong the capabilities of Intel SGX [16], [17], [18].

References

[1] Y. Li, J. McCune, J. Newsome, A. Perrig, B. Baker, and W. Drewry, “Minibox: a two-way sandbox for x86 native code,” in *USENIX conference on USENIX Annual Technical Conference (ATC)*, 2014.

[2] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, “Cross-vm side channels and their use to extract private keys,” in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS ’12.

[3] T. Mayberry, E.-O. Blass, , and A. H. Chan, “Pirmap: Efficient private information retrieval for mapreduce,” in *Financial Cryptography and Data Security (FC)*, 2013.

[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: Improved definitions and efficient constructions,” in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS ’06, 2006.

[5] J. Bethencourt, D. Song, and B. Waters, “New techniques for private stream searching,” *ACM Trans. Inf. Syst. Secur.*, vol. 12, no. 3, 2009.

[6] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the art of virtualization,” in *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, ser. SOSP ’03.

[7] K. V. Machine, “Kernel virtual machine (kvm),” http://www.linux-kvm.org/page/Main_Page.

[8] V. ESXi, “Vmware esxi,” http://www.vmware.com/products/vsphere-hypervisor/.

[9] M. Hyper-V, “Microsoft hyper-v overview,” https://technet.microsoft.com/en-us/library/hh831531.

[10] PXE, “Uefi pxe boot performance analysis,” http://firmware.intel.com/sites/default/files/Intel_UEFI_PXE_Boot_Performance_Analysis.pdf.

[11] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, “Omf: a control and management framework for networking testbeds,” *ACM SIGOPS Operating Systems Review*, vol. 43, no. 4, 2010.

[12] Emulab, “Emulab - network emulation testbed,” https://www.emulab.net/.

[13] A. Furtak, Y. Bulygin, O. Bazhaniuk, J. Loucaides, A. Matrosov, and M. Gorobets, “Bios and secure boot attacks uncovered,” http://www.intelsecurity.com/resources/pr-bios-secure-boot-attacks-uncovered.pdf.

[14] CloudLab, “Cloudlab,” https://www.cloudlab.us/.

[15] T. Mayberry, E. oliver Blass, and A. H. Chan, “Efficient private file retrieval by combining oram and pir,” in *Proceedings Network and Distributed System Security Symposium (NDSS)*, 2014.

[16] G. Noubir and A. Sanatinia, “Trusted code execution on untrusted platform using intel sgx,” in *Virus Bulletin*, 2016.

[17] P. Jain, S. J. Desai, M.-W. Shih, T. Kim, S. M. Kim, J.-H. Lee, C. Choi, Y. Shin, B. B. Kang, and D. Han, “Opensgx: An open platform for sgx research,” in *NDSS*, 2016.

[18] V. Costan and S. Devadas, “Intel sgx explained,” Cryptology ePrint Archive, Report 2016/086, 2016, http://eprint.iacr.org/2016/086.