ADLES v2.0: Managing Rapid Reconfiguration of Complex Virtual Machine Environments

Jason Allen
alle3774@vandals.uidaho.edu

Dr. Daniel Conte de Leon, Dr. Michael Haney
{dcontedeleon, mhaney}@uidaho.edu

University of Idaho
875 Perimeter Dr.
Moscow, ID 83844

Abstract – Cybersecurity education environments and other computer laboratories, such as those that support a DevOps effort, rely on virtualization of many computer systems to meet their complexity and scalability requirements. Often, these environments are set up once in a stable configuration and left relatively unchanged during operation. Given the need to rapidly reconfigure and redeploy environments to suit varying scenarios necessary for cybersecurity research and education, time required for redeployment must be kept to a minimum (e.g. the time allowed between class changeovers on a given day). To support this effort, we have developed ADLES: the Automated Deployment of Lab Environment Systems. Initial release of this tool provided for the management of VMware vSphere in a single hypervisor environment. The current improvements discussed in this paper include management of VirtualBox systems and deployment onto multiple workstations across a network. This improved iteration of ADLES offers greater flexibility to meet the needs of managing complex private cloud environments for cybersecurity research and education, as well as potentially other users with similar needs and requirements for rapid reconfiguration.

Keywords

ADLES, VM, virtual machines, deployment, configuration management, networking, VirtualBox, YAML

1. INTRODUCTION

Virtualization technology and the benefits it brings for flexible computing environments are well established in today's modern technology infrastructure. There are many benefits of managing virtual environments for on-demand computing either in the ever-expanding cloud platforms that provide infrastructure as a service or for security by isolating applications from the host OS. The virtualization of computers has improved drastically and become a staple in the commercial arena. Thanks to the maturity of the technology, educational institutions have incorporated it for providing dynamic lab spaces and, in particular, teaching cybersecurity in a safe controlled and monitored environment.

One of the current challenges of using virtual machines (VMs) for education is managing the deployment and configuration of the various individual VMs and their internetworked configurations. Reconfiguration and redeployment is also a concern since computer lab spaces are generally used for several different classes each with their own requirements. Being able to quickly reconfigure for different classes during the day or adding new VMs as the need arises would provide the ability to support multiple classes in a given semester and keep the computer hardware available for multiple purposes, including research and out-of-class lab exercises.

In many cases, a single VM is sufficient for a lab exercise or for the entirety of a semester class. Often, cybersecurity exercises are *de facto* limited to a single VM for a single student to work on a given exercise. However, more complex cybersecurity labs and projects often require multiple VMs configured to interact with one another, such as for simulating network traffic that is sent, received, and "sniffed" in between (requiring a minimum of 3 VMs and a virtual network switch), or for having multiple systems in various complex attack-defend scenarios, such as the emulation of an entire enterprise scale network. Indeed, deployment of enterprise-scale simulated environments in which different security architectures, policies, and tools are to be tried, examined, and understood requires far more

complicated configurations. The true benefits of virtualization for such requirements are gained in the ability to rapidly reconfigure the simulation on demand.

Generally, VMs along with their networking topology are not grouped together for easy replication onto other host computers or for rapid reconfiguration and modification. Performing restoration and network configuration is a tedious process that takes away from time spent in learning activities, and often requires levels of personnel support not easily come by for a given classroom. This is a present concern for our dedicated cybersecurity classroom and research laboratory environment.

2. BACKGROUND

2.1 RADICL

RADICL is the "Reconfigurable Attack-Defend Instructional Computing Laboratory" utilized by the cybersecurity research and education program at our university [1, 2]. It has been designed and built for the purpose of teaching cybersecurity in a safe and segmented computer lab environment, with the promise of being highly reconfigurable on demand. With such a computer lab environment, it is possible to manage multiple virtualized systems and configurations such as Kali Linux [3], out-of-date Windows XP systems [4], DVWA [5], or Metasploitable [6, 7], that can be made available to students for cybersecurity testing, attack, and defense experience.

There have been several iterations of the RADICL design, and given the benefits of virtualization technology, each iteration has offered some level of flexible configurability. However, the management of various configurations and sets of virtual machines has proven to be challenging for faculty, support staff, and student administrators over the years. Providing more efficient and better controlled deployment and reconfiguration options would improve RADICL's overall effectiveness for providing virtual environments that support hands-on cybersecurity education and research. This paper presents

the most recent developments in our ability to meet the demands of our stakeholders for putting the “reconfigurable” into RADICL.

2.2 ADLES

The ADLES project name stands for Automated Deployment of Lab Environment Systems. The original work and implementation is published and available for download [8, 9]. The objective of the ADLES project was to aid in the deployment and sharing of VM hands-on cybersecurity exercises. The virtualized technologies that enable hands-on exercises are time consuming to set up manually for each instance. ADLES was created to alleviate that need by automating the process. Additionally, the hands-on lab exercises that are built with ADLES can be shared and deployed in a consistent manner to multiple systems or environments.

ADLES is a specification language and associated deployment system created to address these issues by enabling 1) the formal specification of hands-on computing exercises, 2) the automated deployment of specified exercise virtual machines. ADLES is written in Python and utilizes YAML (YAML Ain't Markup Language) [10] configuration files for defining the setup of the VM hands-on exercise systems. The VMs, networking configuration, and groups are specified inside the YAML file that ADLES interprets to build the environment on demand.

ADLES was built to manage the RADICL virtual environment on a central blade chassis system that all students would access via console. It works by managing the VMware vSphere environment specifically. However, the RADICL lab also incorporates the use of desktop computers that are designed to host VMs for students, as well as other hardware configurations such as researcher desktops or student-owned laptops. Not all of these systems fit neatly within the vSphere hypervisor’s management system. From this arises the need to enhance ADLES capabilities and manage a wider variety of hardware and VM possibilities.

2.3 Virtualization Platforms

Initial efforts to incorporate technologies to solve the challenge of rapid redeployment included examining solutions such as Ansible [11], Puppet and Chef [12], as well as systems designed to specifically support VM management, such as those offered by VMware. System container technologies, such as the popular Docker platform [13] and Vagrant [14], show promise for many similar lab environments, such as those used in DevOps efforts [15]. However, these types of systems do not meet the needs of teaching security management in complex enterprise-scale IT environments found in government agencies and commercial organizations around the world. While they work quite well for lab exercises that focus on a single application or task, they do not provide the rich high-fidelity simulation of networks consisting of multiple file share systems, email, web, and DNS services, various subnet zones with internal firewalls, etc., that must be secured with a layered approach of segmentation, minimization, patching, and logging.

One promising system for managing on-demand system configuration is PXE, or the Preboot eXecution Environment (pronounced pixie) [16]. PXE Boot was first looked at as a solution to managing computers in the lab to keep them standardized and easily rebuildable. Additionally, it could be used to run a VM on the bare metal without the need for a host OS with a hypervisor. There are some technical hurdles to building a PXE boot image and executing it correctly. The primary concern is PXE is implemented at the BIOS level which relies on Trivial File Transfer Protocol (TFTP) to receive the image to boot. TFTP is not a securable protocol and could be exploited to modify or load a malicious PXE image. Having such a security concern is not ideal for an environment designed for teaching cybersecurity. It is particularly challenging to monitor, log, and audit its use.

There is a more modern alternative to PXE called iPXE [17, 18] which offers encryption methods for acquiring the boot image. Unfortunately, most BIOS and network cards do not support iPXE. A bootable storage medium is

generally used to launch an iPXE instance which can then load the iPXE boot image from the server. The necessary hardware costs and additional management overhead required for setting up iPXE along with the original issue of making a large library of bootable images made it not a desirable option.

VMware offers systems quite capable of running and managing VMs and is the common choice in business environments. VMware comes in several different types, the most commonly deployed being VMware Player and VMware Workstation. VMware Workstation worked well because it allowed for more control of configuring the virtual network settings. In VMware Player, the free version, virtual networks are still present but the software lacks the internal/separate LAN (local area network) option to isolate the network traffic to just the specified VMs in the way we require. Because students may run these VMs locally on their personal computer without required VMware Workstation licenses, VMware as a platform is not an optimal choice.

As a result of our efforts to work with different VM hypervisors, both VirtualBox and KVM have been evaluated and determined to meet our requirements overall. They are each open source and free to use making it easier to acquire and share them with students' personal systems, as well as across a mixed environment of desktops and blade servers. The extension pack for VirtualBox is proprietary but allows for personal and educational use [19]. Included with VirtualBox is VBoxManage which is similar to VMware's vmrun application. This provides a command-line application for creating and managing VirtualBox VMs. A command-line interface or an API is needed for automating the VM configuration and deployment process at a larger scale.

3. ADLES IMPROVEMENTS IN VERSION 2.0

The current version of ADLES works with VMware vSphere locally on the hypervisor system. To make ADLES more accessible and incorporate the

software choices, support for VirtualBox has been added. Furthermore, ADLES can deploy VirtualBox VMs to the local computer and remote computer on the LAN with SSH.

The new ADLES version which incorporates VirtualBox management still uses the same YAML configuration style and structure as the existing vSphere version and is thus backwards compatible. As a result, the YAML parameters described in the original ADLES publication [8] as part of 1) the infrastructure specification, 2) the package specification, and 3) the exercise specification, continue to be valid. ADLES now also utilizes the VBoxManage application, which is included with the default VirtualBox installation, to interface with VirtualBox. Either VirtualBox or VMware vSphere can be specified in the configuration.

Since new features have been added, several new parameters can be set in the ADLES configuration files. The first new parameters in the 'infrastructure.yaml' file are found in a new "SSH" section. This section defines fields to determine what target host computer to use for remote deployment and how to connect to it. Under the SSH section, sections with different target node names can be added, each with values for hostname, connection port, and user login ID. The values must point to a computer with those parameters and have a non-password protected SSH key already registered on the target SSH computer. The SSH private key file must match the name that is given to the SSH host in the 'infrastructure.yaml' file. This is used to associate the correct SSH key to the correct SSH host in the configuration. If the SSH section is not present, ADLES will deploy to the local machine on which it is running.

```
ssh:
    pc1-ssh-name:
        hostname: 192.168.0.42
        port: 22
        username: alice
        basefolder: ~/.ADLES/
```

The ‘infrastructure.yaml’ configuration file also now provides a “Group” section. This provides control over a more sophisticated model of deploying groups of VMs that are targeted to individual student workstations, rather than on a single hypervisor environment that all students connect to. This capability allows for duplication of complex VM environments to multiple students simultaneously. An example is a scenario for 3 or more VMs that include a “victim,” an offensive “attacker,” and a defensive “security monitor” system for each student to work with. These VMs can be configured for Windows 7, Kali, and Security Onion respectively, as an example, and then grouped together for deployment as a class exercise for each student’s workstation in the lab.

```
...
group:
    description: "group description"
    services:
      name-of-a-service:
        service: vm-alias-name
        networks: ["network-name"]
        instances: 1
        ssh: pc1-ssh-name
```

With these three new capabilities, the ability to use VirtualBox locally and on remote hosts in addition to the existing VMware vSphere control, and the ability to deploy groups of systems in parallel, the functionality and effectiveness of ADLES as a lab management system is greatly increased. With version 2.0 of ADLES published and available to the community, a wider variety of cybersecurity lab environments similar to RADICL can be more efficiently managed and reconfigured on demand.

4. FUTURE WORK

The current work effort on the development and maintenance of ADLES includes adding additional features and support to make it better suited for a wider range of environments. The primary planned feature is to add support

for the KVM hypervisor. KVM is planned to be added because VirtualBox does not meet all the scenario requirements for RADICL. VirtualBox is a good option for an easy end-user interface that anyone can quickly set up and work with; it is classified as a "type 2" hypervisor in that it operates within a host OS (e.g. a student's workstation or laptop). However, there are times where an end user interface is not needed and the VMs need to be run on a centralized server for greater control of security policies or manipulation by the instructor. In that scenario, a type 1 hypervisor (one that runs on "bare metal" and provides greater functionality and control of guest VMs) would be better suited to run and manage a larger set of VMs for different exercise scenarios. This is the current support structure for ADLES managing VMware vSphere environment. However, due in part to licensing costs and issues with multiple segmented virtual environments, it is desirable to migrate the RADICL lab environment to KVM [20] and the oVirt management platform [21]. It is likewise a goal of these projects to make the system available to as wide an audience as possible in the cybersecurity education community.

Secondly, the remote SSH deployment feature added to ADLES is currently bottlenecked by the network interface and replication of systems. When multiple remote SSH computers all need the same VM image, the host computer running ADLES must send it individually to all those computers. This results in the same VM image data being sent multiple times. To speed up the process and utilize the bandwidth of the target remote SSH computer, a peer-to-peer file syncing application will be incorporated. Additional optimization is currently planned through the management of virtual machine snapshots or incremental additions and changes, rather than copying the entire guest OS filesystem each time a configuration change is implemented. While this proves to be time consuming, the actual effort on the part of the instructor or exercise designer is still dramatically reduced, and pushing out guest VMs can be scheduled and fully automated.

5. CONCLUSION

The deployment and configuration of VMs for a hands-on cybersecurity exercise or other reconfigurable computer lab scenarios can be time consuming and error prone. This has, in our experience, led to detraction from the time spent on learning by doing cybersecurity exercises. The RADICL lab used for scenario-based cybersecurity research and education is an example environment that can greatly benefit from the automation provided by ADLES. With VM lab exercises defined in an ADLES YAML file which is easy to understand and manage, the setup can be shared and easily deployed in other locations. This allows cybersecurity instructors to focus their time and effort on teaching rather than on the system management necessary for set up and reconfiguration of scenarios and exercise environments.

REFERENCES

[1] Caltagirone, S., et al. *RADICL: A Reconfigurable Attack-Defend Instructional Computing Laboratory*. in *Security and Management*. 2005.
[2] Caltagirone, S., et al. *Design and implementation of a multi-use attack-defend computer security lab*. in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*. 2006. IEEE.
[3] *Kali Linux*. 2019; Available from: https://www.kali.org/.
[4] Microsoft. *Support for Windows XP Ended*. 2014; Available from: https://www.microsoft.com/en-us/WindowsForBusiness/end-of-xp-support.
[5] Dewhurst, R., *Damn Vulnerable Web Application (DVWA)*. 2012.
[6] Rapid7. 2019; Available from: https://github.com/rapid7/metasploitable3.
[7] Sinha, S., *Setting Up a Penetration Testing and Network Security Lab*, in *Beginning Ethical Hacking with Kali Linux*. 2018, Springer. p. 19-40.
[8] de Leon, D.C., et al., *ADLES: Specifying, deploying, and sharing hands-on cyber-exercises.* Computers & Security, 2018. **74**: p. 12-40.
[9] Goes, C. *ADLES*. 2018; Available from: https://github.com/GhostofGoes/ADLES.
[10] Ben-Kiki, O., C. Evans, and B. Ingerson, *Yaml ain't markup language (yaml™) version 1.1.* yaml. org, Tech. Rep, 2005: p. 23.
[11] Hochstein, L. and R. Moser, *Ansible: Up and Running: Automating Configuration Management and Deployment the Easy Way*. 2017: " O'Reilly Media, Inc.".
[12] Önnberg, F., *Software Configuration Management: A comparison of Chef, CFEngine and Puppet*. 2012.
[13] Merkel, D., *Docker: lightweight linux containers for consistent development and deployment.* Linux Journal, 2014. **2014**(239): p. 2.
[14] Matthias, K. and S.P. Kane, *Docker: Up & Running: Shipping Reliable Containers in Production*. 2015: " O'Reilly Media, Inc.".
[15] Ebert, C., et al., *DevOps.* Ieee Software, 2016. **33**(3): p. 94-100.
[16] Tao, L., *Implementation of Computer Lab Maintenance Using DHCP and GHOST Based on PXE [J].* Experiment Science & Technology, 2006. **2**.
[17] iPXE. *iPXE - Open Source Boot Firmware*. 2018; Available from: http://ipxe.org/download.
[18] Torres, M.D., *An open source approach to serve a large number of computer users using block-level streaming*. 2016, The University of Texas Rio Grande Valley.
[19] VirtualBox. *Oracle VM VirtualBox Extension Pack Personal Use and Evaluation License (PUEL)*. July, 2017; Available from: https://virtualbox.org.
[20] Chirammal, H.D., P. Mukhedkar, and A. Vettathu, *Mastering KVM Virtualization*. 2016: Packt Publishing Ltd.
[21] Lesovsky, A., *Getting Started with OVirt 3.3*. 2013: Packt Publishing Ltd.