Remote Virtual Information Assurance Network

Ronald C Dodge JR[1] , Corey Bertram[2] , Daniel Ragsdale[3]
1,3 Department of Electrical Engineerig and Computer Science,
United States Military Academy, West Point, NY,
ronald.dodge@usma.edu, daniel.ragsdale@usma.edu
2 George Washington Universtiy, Washington DC, qr7@gwu.edu

Abstract. The use of virtualization technologies to increase the capacity and utilization of laboratory resources is widely used in classroom environments using several workstation based virtualization products. These virtual networks are often "air gapped" to prevent the inadvertent release of malware. This implementation however requires users to be in the classroom. A novel extension on this concept is to design the infrastructure to support remote access to the virtual machine(s) using virtual server applications, while maintaining the complete isolation of the virtual networks.

1 Introduction

In 2000, virtualization started to become a popular tool to enable the development of innovative instructional techniques. Many universities have adopted the use of virtual machines to provide students with an environment to understand how systems interact, demonstrate communication protocols, experiment with malware for exploit understanding, and provide more scalable lab environments. These systems are traditionally implemented using workstation based virtualization applications that are accessible only from the workstation they reside on. Students in this environment must come into the lab to access the virtual machines since they are bound to a physical host. The architecture described in this paper frees the student from needing to go into the lab and also reduces the likelihood that no lab resources are available by consolidating the virtualization application onto a server platform. While the discussion here is focused on information assurance, the architecture described supports other requirements like network fundamentals and operating systems.

The paper continues as follows: in section 2 we describe differing virtualization environments and previous work in virtual lab implementations. In section 3 we describe a web services oriented architecture implemented by the authors and conclude in section 4.

2 Background and Previous Work

Starting in the 1960's large main frame systems used virtualization to provide individual computing environments for users (for example, the IBM System 360). As platforms evolved to make personal computers relatively commonplace, the need for virtualization decreased; to the point where it became rarely implemented. As the personal computer became increasingly more capable, system resources began to exceed computing requirements and the opportunity to instantiate multiple virtual machines on a single personal computing platform emerged. [1, 2] Virtualization products include open source projects such as OpenVX and Xen; and commercial products, VMware, VirtualPC, and Parallels.

Virtualization essentially decouples the physical machine hardware from the operating system by inserting a software virtualization layer. The following discussion focuses on the VMware Workstation and VMware Server product implementation. The virtualization application forms a normalized representation of the physical hardware that each "guest" operating system utilizes. This normalization layer sits on top of the host operating system. This virtualization layer presents each guest operating system with its own set of virtual hardware (CPU, RAM, Hard Drive…). The guest operating system (and data) is a file on the host system. This architecture makes it trivial to move virtual machines from one physical system to another. This capability is very important to the deployment described in section 3. The virtual machines' network card can be bridged to the host's network card (enabling the virtual machine to be connected to the physical network) or connected to a virtual switch (enabling communication with other virtual machines).

The past 5 years have seen various employments of virtual infrastructures to facilitate education and training. This is in response to a realization that students must practice the concepts and theories discussed in lecture to fully understand the material. This point has been well argued in many papers [3, 4] and evidenced by the growth in virtual and remote access labs used by education institutions and industry. The general architecture agued for in the referenced literature dictates that an optimal lab provides an "air gapped" system environment to prevent malware or malicious activity (scanning) from interacting with non-lab systems. Additionally, the lab should present the user with a robust suit of operating systems and applications that communicate over an air gapped network that is also separate from other air gapped networks. The advancement offered in this paper is to provide an architecture that satisfies these requirements while permitting access to the lab to remote users. The design of the architecture described in this paper is builds on the experience of leading institutions in the deployment of IA/CS environments, such as the University of Alaska Fairbanks, Brooklyn Polytechnic University, the Universitaet Trier in Germany, and the University of Milan in Italy. The architectures currently adopted by these institutions for use in their IA/CS research and education programs is similar in intent but vary in implementation; each having unique advantages and disadvantages.

The University of Alaska Fairbanks built a virtual infrastructure where VMware workstation is employed in an isolated network where the guest operating system files are stored on a central file server. Students log into a machine in the lab and access their virtual machine files on the file server. The CPU and other non-file system resources on the local machine are used, however all file I/O is accomplished on the file server. The lab has strength in that it is air gapped from any production network and the Internet; preventing any of the security tools or malware from unintentionally interacting with other networks. This architecture however requires the student be in the lab.

An alternative architecture is used at the Brooklyn Polytechnic University. [6] Here, the infrastructure is built on a combination of physical switches and routers and the VMware ESX virtualization platform. Instructors create a collection of hosts (on the ESX server) and switches/routers that are shared by groups of students. The students accessing the lab are presented with a diagram of the network and by clicking on each of the devices receive a console session (over an applet in the browser). The lab uses a VPN concentrator to provide connectivity between the networks over the public network. The strength of the ISIS environment is that students do not need to be in the lab to use the resources. However the VPN concentrator breaks the “air gap” network philosophy, leaving open the opportunity (and challenge to some) to subvert the established controls. The sharing of hosts by a number of simultaneous users is a second drawback.

A third virtual lab is the IT security Tele-lab at the Universitaet Trier, Germany. [7] The Tele-lab implements architecture built using User-Mode Linux. The users interact with the virtual machines directly using a Virtual Network Computing (VNC) client application. This presents a significant drawback as the virtual machines are connected to the physical network; leaving open the possibility of potential for use misuse by exposing the security lab to an external network. Additionally, User-Mode Linux only supports a virtual machine based on the kernel of the host machine. This greatly limits the type of operating systems users can interact with.

A final comparative lab is the Open Source Virtual Lab hosted at University of Milan, Italy. [8] This lab offers remote access to a collection of XEN virtual machines. The open source nature of the lab (and XEN 2.0 limitations) permits only the use of Linux virtual machines. This lab architecture permits users to log into the system through a web interface to enable the system to start the requested virtual machine. Once the virtual machine is started a direct SSH connection between the user and the virtual machine is provided through a terminal shell. Currently, the documented implemented system only supports Linux virtual machines; however XEN 3.0 will support Windows operating systems. The current architecture however still only supports terminal shell interaction. Additionally, while the infrastructure used in the Open Source Virtual Lab differs from the others previously discussed the same disadvantage of user access to local networks and limited operating system interaction.

The solution described in the following section implements a remote accessible architecture that provides the benefits available to the ASSERT and the United States Military Academy’s Information Warfare Analysis and Research (IWAR) lab. [4] The Remote Access Virtual Information Assurance Network provides an architecture where it is not possible to communicate with external networks, while providing a robust suite of Windows and Linux hosts controlled through a VNC session.

3 System Architecture

The objective of the Remote Access Virtual Information Assurance Network (RVIAN) is to provide users with a lab experience – only not in the lab. The laboratories described in the previous section all attempt to provide this environment and recognize that the optimal configuration would provide:

Realistic Heterogeneity: The operating systems used should represent the full spectrum of operating systems that a user should understand from a security perspective.
Configurability: The hosts the user has access to should be as configurable as a system in a physical lab.
Isolation: The network the users control is air gapped from external networks. This is to prevent any accidental exposure of malware and malicious activity to external network.
Scalable: The virtual lab should provide a scalable solution that provides the same if not more capability of a physical lab. For instance, in a physical lab if a student requires two systems to conduct a network sniffing lab, then the designed architecture should not require more than two physical systems.
Cost Effective: The virtual lab should be affordable to implement and not require excessive system administration/maintenance.
In the following sections, we describe the implementation of the RVIAN, and then readdress the five configuration requirements and how they are met.

3.1 Physical/Software Infrastructure

The RVIAN is implemented using a web services architecture, as shown in figure 1

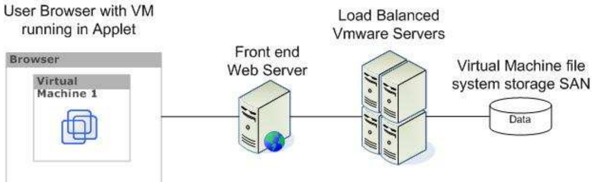
User Browser with VM running in Applet
Browser
Virtual Machine 1
Front end Web Server
Load Balanced Vmware Servers
Virtual Machine file system storage SAN
Data

Figure 1: RVIAN Architecture

The user enters the system through a web browser and authenticates using a userID and password. The front end web server provides information on the RVIAN environment, links to required software, load balancing for the backend virtual machine servers and the authentication logic to allow access to the virtual machines, as shown in Figure 2 The Web server is powered by Apache 2.2 and PHP. Behind the server, a collection of servers running VMware server run virtual machines for each user. The VMware servers use a storage area network to hold the virtual machine file systems.

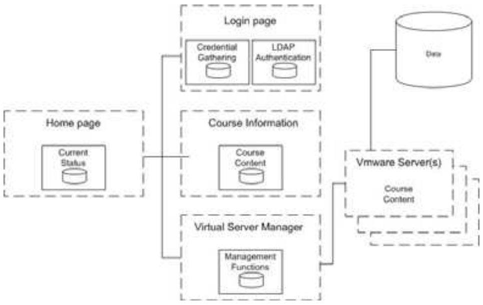

Figure 2: RVIAN Web page layout

Access control is implemented through LDAP credential authentication. Users are initially provided an account on the system. Along with the credentials, users are provided a user directory on the storage area network. The user directory stores the file system for each virtual machine allocated to the user.

Using a menu driven system, the user selects which virtual machine(s) to launch. The selected virtual machines can be based on a predefined selection to be used for a specific lesson module or the user's desire to experiment. The available virtual machines are typically copied over to the user directory on the storage area network by the system manager in advance; however the system can support copying new virtual machines dynamically, assuming the user disk quota is sufficient. The virtual machine manager selects which server to host the virtual machines on and uses port forwarding to return to the user's browser the virtual machine using TightVNC through a Java applet. The Virtual Server Manager load balances the workload using performance statistics from the Virtual Servers to determine which server to host the user's virtual machines.

3.2 Remote Lab Requirements

This architecture supports each requirement previously discussed.

Realistic Heterogeneity: Vmware Server supports a wide range of operating systems in full graphic interaction mode. The operating systems supported include any version of Windows, most distributions of Linux, and Solaris X86. This full interaction capability and robust suite of guest operating systems provides an unmatched flexibility in virtual labs.

Configurability: The VMware server virtual machines can be configured by the instructor to meet a variety of instructional and training needs. Instructors can configure the network infrastructure to consist of many (ranging from 1 to 99) virtual switches that have no external connectivity. Users have full control over the operating system of the virtual machine – the environment is what contains the user.

Isolation: The Virtual Machine Networks (or VMnets) implemented by VMware provide an environment where virtual machines can connect to up to 99 virtual networks (actually each virtual machine limited to a maximum of four network interface cards). These virtual networks can be configured such that there is no connection to a physical network (even though the supporting virtualization software, VMserver, provides a connection through which the user interacts with the virtual machine). This environment provides separates the virtual network from the external networks the users communicate on with the instances of VMware Server. For example, if VMware server were running on three backend servers, the VM manager would determine which server to run the user's virtual machines and also determine what virtual networks they would use. This segregation keeps virtual machines from interacting with both external networks and other user's networks.

Scalable: Depending on how the virtual machines are built, each virtual server we are using (dual 3.2 processor, 8 GB RAM) can support 12~14 virtual machines. In the modules we have developed each user needs approximately four virtual machines. The system scales with the number of virtual servers used. In our lab we have a 10 blade enclosure running VMware server; allowing for approximately 25 simultaneous users.

Cost Effective: Laboratory equipment is costly. A student workstation capable of running a standard suite of virtual machines needed for the modules we implement is roughly equivalent to 50% the cost of a server running VMware server. (VMware server itself is free.) This cost however is outweighed by the flexibility the server based system provides.

4 Conclusion

Literature strongly supports the benefits of experience based learning and the development of laboratories to provide that exposure. The examples discussed in the opening of this paper represent a few of the examples of efforts to provide a laboratory experience; only without the physical lab. The greatest difficulty in providing the capability for remote users interact with and control networks designed for security training and education is the requirement to ensure the networks remain isolated. The benefit of the solution described in this paper rests on the robust virtual networking supported by VMware server. The capability provided allows for students to interact with the virtual machines using TightVNC presented in a Web Browser Java Applet; however the networks the virtual machines use are completely isolated from the external network and the networks of other users. Future work includes the development of a physical hardware management system that allow for the inclusion of physical routing and switching equipment into the environment while still providing separation between the students virtual network and the underlying support network.

References

1. M. Rosenblum, “The Reincarnation of Virtual Machines,” ACM Queue, Vol. 2 No. 5 - July/August 2004.
2. B. Supnik, “Simulators: Virtual Machines of the Past (and Future)”, ACM Queue, Vol. 2 No. 5 - July/August 2004.
3. Cynthia E. Irvine, “Amplifying Security Education in the Laboratory,” Proceedings IFIP TC11 WC 11.8 First World Conference on Information Security Education, pp 139 -146, Kista, Sweden, June 1999.
4. D. J. Ragsdale, S. D. Lathrop, and R. C. Dodge," Enhancing Information Warfare Education Through the Use of Virtual and Isolated Networks," The Journal of Information Warfare, Volume 2, Issue 3, pp. 47-59 August 2003.
5. B. Hay and K. Nance, “Evolution of the ASSERT Computer Security Lab,” Proceedings of the 10th Colloquium for Information Systems Security Education, page 150-156, Adelphi MD, June 5-8, 2006.
6. V. Padman and N. Memon, “Design of a Virtual Laboratory for Information Assurance Education and Research,” Proceedings of the 5th IEEE Information Assurance Workshop, West Point, NY, 17-19 June 2002.
7. J. Hu, D. Cordel, and C. Meinel, “A Virtual Laboratory for IT Security Education,” Proceedings of the Conference on Information Systems in E-Business and EGovernment (EMISA), pp. 60-7, Luxembourg, 6-8 Oct 20041.
8. E. Damiani, F. Frati, and D. Rebeccani, “The Open Source Virtual Lab: a Case Study,” Workshop on Free and open Cource Learning Environments and Tools, Como, Italy, 10 June 2006.