VTE: THE VIRTUAL TRAINING ENVIRONMENT
Advanced Virtual Lab Authoring and Delivery

C. A. Cois, J. Beckwith, J. Wrubel and J. Herrman

Software Engineering Institute, Carnegie Mellon University, 4500 Fifth Ave, Pittsburgh, PA, U.S.A.

Keywords: Virtual labs, Distance learning, Content authoring, Asynchronous Labs, Automated delivery, VMware.

Abstract: The Virtual Training Environment (VTE, https://www.vte.cert.org) is an online education and training system developed by the Immersive Learning Technologies group at the Carnegie Mellon University Software Engineering Institute. VTE offers asynchronous, remote training in cyber-security, information technology, digital forensics, and other technical areas to over 30,000 users, primarily within the Department of Defense (DoD) and various branches of the United States Federal Civilian Agencies. A key aspect of the comprehensive training suite offered by VTE is an on-demand virtual lab system, featuring automated delivery of complex virtual lab environments to users through a standard web browser, with no additional client-side software installation required. VTE is able to deliver training environments 24 hours a day to users in remote locations, operating within various levels of network security. In addition to a virtual lab system that is uniformly accessible to student users, VTE provides a virtual lab authoring system with the same ease of use, delivering powerful virtual lab development tools through an accessible interface to domain experts and educators, eliminating the need for expertise in proprietary virtualization technologies. This paper will present the design and function of the VTE virtual lab system, including both student and lab author user experience descriptions.

1 INTRODUCTION

Training students in applied fields such as cyber security, computer forensics, or information technology presents a unique set of challenges. Lectures and reading materials have limited efficacy, as students will not be comfortable performing learned techniques until they have experienced hands-on training on real computer systems. However, many techniques and software packages, particularly those related to cyber security, malicious code, and computer forensics, can pose real threats to production systems when used by students in training exercises. Such training is only safely performed in an isolated network environment, secured from production networks. While many have attempted to build isolated physical computer networks for education and training labs (Robila, 2004; Yang, 2004), this solution is both time-consuming and costly, as IT professionals must construct lab environments from purchased hardware. This approach is also highly inefficient, allowing only one training exercise to take place at a time, and requiring manual reconfiguration of the

network or systems hardware to deliver a new training exercise. Virtual labs have become increasingly popular in recent years as an alternative to the imperfect physical training lab, providing realistic hands-on training in an inexpensive, highly adaptable system. (Bulbrook, 2006; Greenberg, 2004; D. Hu, 2008; J. Hu, 2004; Nabhen, 2006; Vollrath, 2004)

Virtual labs consist of virtual machines configured with the desired operating systems and software, deployed in a specific virtual network configuration. Such an environment provides students with hands-on training in real software systems in a safe and secure network environment. This leaves students free to interact in any way with an immersive environment, isolated from the actions of other students, without risk to production networks or fear of damaging systems. When training students in cyber-security, network vulnerability, malicious software, or hacking techniques, this isolation becomes a vital component of any training system. While isolated physical networks can be built to serve the same purpose, they require an exceptional amount of time and

expense to build and maintain. Additionally, changes in configuration of a physical training network can take hours of time from IT experts, while virtual labs of any configuration can be created and deployed on the fly by automated virtual lab management software.

The Virtual Training Environment (VTE, https://www.vte.cert.org) provides a robust virtual lab system currently supporting over 30,000 users around the world. By delivering a virtual desktop interface to computers in a virtual lab environment through a standard web browser, VTE is able to give users access to realistic, immersive training at any time, from any location. VTE requires no client software other than a web browser with an Adobe Flash[TM] plug-in installed, making it highly accessible to students from within university networks, at work, or at home. The VTE lab management platform maintains reusable virtual lab configurations that can be rapidly created and deployed on-demand requested through the VTE website. Each user is given an individual instance of a virtual lab to interact with, with the goal of providing an immersive, personalized experience to users with learning value beyond that of watching lectures, reading content, or interacting with simulations. (Greenberg, 2004) A list of tasks to be performed in a given lab is integrated into the virtual lab interface, keeping the students focus within the lab environment.

Further VTE development has extended this user-friendly, intuitive interface to virtual lab technology to lab authors as well as students. By putting the power of virtualization for training and education into the hands of instructors, without requiring expertise in virtualization technologies or platforms, the VTE team hopes to allow authors to focus their efforts solely on content development, and to engage authors who might otherwise be remiss to develop virtual labs. Authors are given an intuitive visual interface for creating virtual lab networks and configuring associated virtual machines. Lab instructions are created as the virtual machines are configured allowing authors to perform the lab to verify functionality, while marking points of progress to be saved as fallback states for students who have made irreversible errors while completing the lab. Giving authors complete control over virtual lab creation allows educational content to be created and deployed faster, while the easy-to-use toolkit provided by VTE allows all domain experts to participate in content creation without esoteric knowledge or formal training.

2 SYSTEM DESIGN

The ultimate goal of virtual labs is to provide a completely immersive, secure training environment to anyone, anytime, anywhere. (Greenberg, 2004) To achieving this goal, a system must be designed with three primary attributes:

- Automated content delivery,
- Broad accessibility,
- An intuitive, functional interface.

The following sections will discuss how VTE arrives at the first two goals. Descriptions of the user interfaces in VTE will be presented in later sections featuring the student and author experiences in the VTE system.

2.1 Architecture and Storage

A complex lab management system is necessary to facilitate on-demand automated delivery of virtual labs. The platform driving VTE is made up of four primary architectural components:

- Web tier,
- Application tier,
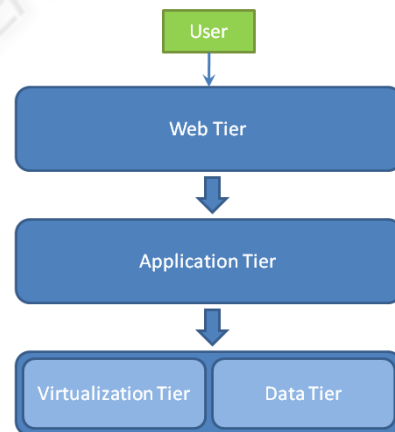- Virtualization tier,
- Data tier.

Figure 1: shows a basic representation of the system architecture.

The web tier contains web servers that host the VTE website and accept user commands. All data presented to and received from the user passes through these servers. The application tier contains servers that run the VTE application, which manages the configuration of virtual labs and communication with the ESX hosts to issue the commands to create and destroy virtual machines as required. The

virtualization tier is a scalable pool of virtualization hosts, consisting of VMware ESX servers in the present implementation of VTE. These servers accept commands from the VTE application and host the virtual machines and virtual network components that make up each virtual lab. Ultimately, a user is connected from their web browser, through the web and application tiers to the ESX server which hosts the virtual machine they are currently accessing. The data tier contains database servers to hold all data for lab configurations, user accounts, and all other stored application information. Also contained in the data tier are storage systems holding data for the virtual machines used in VTE labs.

System data for each virtual machine is stored in one or more data files, called virtual disks, which simulate the hard disk of a physical computer. Virtual disks contain the operating system and any other information stored on the virtual machine, such as software applications or data files, in binary format. These virtual disks, known as base disks, are opened in read-only mode by virtual machine instances using them and can thus be used to define numerous identical virtual machines. Base disks must remain concurrent between all users to provide identical lab experiences, thus no data can be written to them while they are being referenced by virtual labs. Any changes made by the user during a virtual lab session are stored not in the base disk, but in a separate change file unique to that user. These changes can be stored separately and when a lab session is closed and applied again to the virtual machine when the session is re-opened, creating a stateful experience for the student while maintaining the integrity of the base disk. In addition to managing lab deployment, the VTE lab management system handles user data storage and retrieval, providing a seamless experience for the student.

2.2 User Access

VTE virtual labs are networks of virtual machines deployed on remote VMware ESX hosts. When a student requests a lab, a new instance of each virtual machine in the lab configuration is automatically created and powered on. When the virtual machines have booted into their operating systems, the user is given a secure connection to the desktop of a virtual machine through their web browser, which appears as the familiar desktop interface of the operating system installed on the virtual machine. Mouse and keyboard interactions are handled from within the

browser by Adobe Flash^{TM} and passed through to the virtual machine.

To facilitate delivery to students on secure networks, such as military personnel, the remote display session is altered to connect to the server over port 80 or port 443, the standard ports used to deliver web pages on the internet. In highly secure networks, all other ports may be blocked by firewalls, and thus unavailable to deliver remote display session data. This adaptation is vital in allowing VTE to deliver training to U.S. government and military personnel, and is also of benefit for organizations with highly secure network practices.

3 STUDENT EXPERIENCE

Upon logging in to the VTE website, students can register for courses or search for available pieces of content. Once registered for a course, students can view the content items of that course, including virtual labs. (Figure 2)
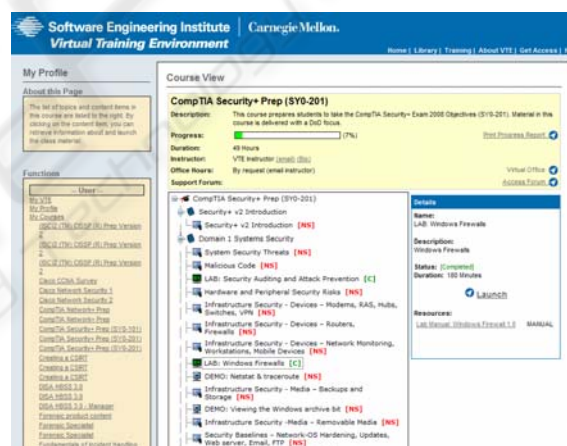

Figure 2: Student course view in VTE. Clicking the "Launch" button in the right-hand panel will launch the selected lab, sending the user to the lab interface screen.

Clicking the "Launch" button initiates the process of generating an instance of the specified lab, and redirects the user to the virtual lab interface. (Figure 3) The virtual lab interface is designed to be a unified, immersive environment for all information and functionality necessary to complete a virtual lab. On the left side of the interface is a task list, containing ordered descriptions of tasks to be performed, including tables, images, and other metadata. Tasks expand and collapse as the user records their progress through the lab. On the right side of the screen is a lab network diagram showing

icons for each virtual machine or virtual switch and the network connections between them. Double-clicking on a virtual machine creates a secure connection to the desktop of the machine and presents it in a window for the user to interact with.
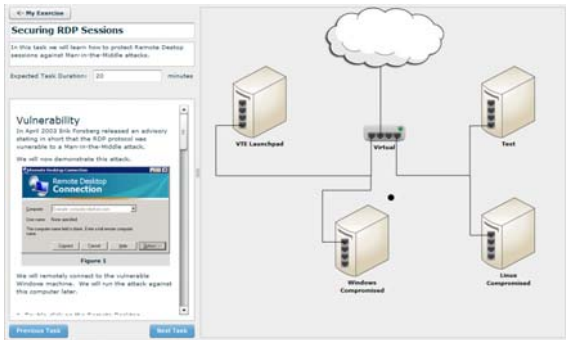

Figure 3: Lab user interface. On the left is a task panel, on the right is a network diagram of the virtual lab environment.

This interface is intuitive and easy to understand. It allows the user to visualize the network they are working with, and interact with each virtual machine individually while following an outline of tasks to complete, all without straying from the same browser window. This allows the user to maintain focus in one place, creating an immersive environment conducive to learning and skill retention.

When a lab has been completed, the user can elect to finish the lab and receive credit for completion. Should the user want to end a work session before all tasks are complete, the user has the option to save their progress and exit the lab. The lab management system will then take over, storing all changes the user has made to the base virtual machine disks since the session began in preparation for recreating this state for the user the next time they launch the lab.

4 VIRTUAL LAB AUTHORING

A streamlined virtual lab delivery system allows rapid, automated deployment of reusable software environments to students, eliminating expensive and cumbersome construction of secure network environments and computer systems for each learning session. However, this advancement is shallow without effective methods for creating virtual lab content to deliver. Virtualization technology and the usage of platform-specific virtualization tools are complex domains for which content authors are unlikely to possess expertise. Furthermore, should advances or changes to proprietary virtualization technologies require a change in underlying virtualization technology, content authors should not be expected to learn esoteric new tools to accomplish the same tasks in an updated system. Abstracting the underlying virtualization technology away from the content authoring interface solves both of these issues. To realize this goal, the VTE team has developed a powerful, yet intuitive system for the creation and configuration of virtual labs. A reference for specific pieces of terminology pertaining to the VTE virtual lab system can be found in Table 1.

Table 1: Definitions for VTE virtual lab terminology.

Virtual Lab	A network of virtual machines and associated Exercises.
Exercise	A unit of learning in VTE. Consists of one or more tasks.
Task	An individual action a student must complete as part of a larger exercise.

4.1 Shared Authoring Resources

Creating a new virtual machine is often a time-consuming process, as it requires installing and configuring an operating system, configuring network settings, and installing any necessary software applications. To streamline this process for virtual lab authors, VTE allows virtual machines to be shared among multiple labs. When an author assigns virtual machines to a lab, they may use existing virtual machines previously created by other authors and saved in the VTE system, rather than creating new virtual machines of their own.

Archived virtual machines will have operating systems and often a collection of software applications already installed. Common virtual machine configurations such as domain controllers, web servers, or vulnerable client systems may be used from the archives without any additional changes, saving the author a great deal of time. Even if an author elects to design his own virtual machines, they can use any archived virtual machine as a starting point, including stock installations of common operating systems provided within VTE. These options increase the speed and efficiency of virtual lab creation, allowing more quality content to be created and delivered to students.

4.2 Interface

A primary goal of the VTE system is to provide as much quality training as possible to learners. A functional, intuitive interface for the design of virtual labs is necessary to allow educators to effectively create high quality content for students to consume. The virtual lab authoring interface in VTE has been designed for ease of use and efficient lab creation, removing the cumbersome aspects of creating virtual machines and networks using traditional virtualization tools designed for IT professionals, in order to spur content creation.

4.2.1 Virtual Network Configuration

To create a lab, an author logs in to the VTE web site and is directed to a virtual lab interface similar to that of a student completing a lab. This design leads the author to experience the lab in the same way as the student, in an effort to help the author to design the lab to communicate the desired information as effectively as possible.

Figure 4: Virtual lab network configuration interface. Authors select virtual machines and network components by name from the left-hand panel and drag them into the lab configuration. Lines are dragged between components to create a network connection.

An author's first step is to design the network configuration of virtual machines that make up the lab. As seen in Figure 4, authors can select virtual machines and networking components from a list and drag them into the lab area, connecting lines between them to indicate a network connection. This network diagram system will be familiar to those in IT and computer security fields, which comprises a large percentage of the authors for VTE labs. The author may optionally alter the configuration of the

virtual machines selected to meet the needs of the lab being created. Often this includes installing software, changing network settings, or adding data to the machine for students to use to test various software tools.

4.2.2 Exercise Creation

Figure 5: Author interface with task creation panel on left. Instructions can be edited and previewed in the same format/interface as students will view them.

Once virtual machines have been configured and the network configuration has been set, the author initiates the exercise building stage of lab development. A task interface is brought into view on the left hand side of the screen, allowing the author to create Exercises and Tasks (Table 1), as shown in Figure 5. By double-clicking the mouse on any virtual machine icon, the author can bring up a window with a connection to that virtual machine's desktop and interact with that virtual machine in the same way as with a physical computer. (Figure 6) Thus, while defining these steps to lab completion, the author is able to personally perform the desired actions on the virtual machines in the lab, verifying the expected outcome of each task.

This process also allows the author to take screenshots of the virtual machine and embed the images into the task list, by highlighting a region of screen and simply dragging it into the task panel. At the end of each exercise, the author is able to save the state of all virtual machines, creating a known good starting point for the following exercise. These saved states allow students to revert to known good lab states in the case of irreparable errors while completing the lab, rather than being forced to start at the beginning and repeat steps previously completed correctly.

Figure 6: Authoring interface with active connection to the desktop of a virtual machine. On the left is the task panel, showing the task the author is currently completing.

Authors can also re-enter this lab creation interface with existing virtual labs, to alter or fix parts of the lab. This action creates a new version of the lab, leaving the old version intact so as not to affect the work of in-progress students.

5 CONCLUSIONS

Computer education issues a unique challenge to the educational community. Hands-on training is necessary in preparing students to perform duties in fields such as cyber security, computer forensics, and information technology. However, many educational exercises, especially in security areas, pose real risks to production networks. (Bulbrook, 2006) Such exercises are better performed in an isolated network environment. An isolated physical training environment is time-consuming and expensive to build, is restricted to a single physical location, and requires additional time and expense to reconfigure for new exercises or to repair damage done during a training session.

Aside from creating an isolated physical network, a virtual lab environment is the only solution to potential security risks. In addition to safely isolating exercises from production networks, a virtual lab offers other benefits, such as:

- Access for multiple concurrent users,
- Instant deployment of lab environments,
- Quick reconfiguration of lab environments,
- Safe access from any network environment.

Students are each given access to an independent instance of the lab environment that can be accessed from campus networks, production networks, or

even from a student's home computer. Changing a lab environment requires no hardware changes, only reconfiguration of virtual machines and switches, resulting in a multi-functional lab solution that is both time- and cost-effective.

The Virtual Training Environment (VTE) was designed to deliver virtual labs for computer security, forensics, and information technology training to United States government professionals, including those in the Department of Defense and U.S. Federal Civilian Agencies. To facilitate delivery of virtual labs to users in most network environments, VTE has been designed to deliver a secure connection to virtual machine desktop sessions in a web browser, and requires no proprietary software installation on the student's system. Lab exercises and tasks have been integrated into the student's lab interface, increasing the immersion of the student by maintaining focus on a single virtual lab interface. Students are able to save and resume progress through virtual labs, functionality made possible by careful management of virtual disk data in the VTE lab management software framework.

Future work to this end includes formal studies of VTE as a learning system. The authors hope to conduct studies of the efficacy of the VTE system as a learning tool, comparing knowledge retention and performance on exams of users trained with VTE to those trained elsewhere.

In addition to a unified student interface, VTE has been designed with an intuitive interface to provide powerful virtual lab creation tools to content authors and domain experts, giving them the ability to design virtual labs for education and training exercises. Pre-configured virtual machines and a drag-and-drop interface for virtual lab network design provide a swift and efficient process for designing training labs. The unification of highly accessible virtual lab delivery mechanisms and powerful, intuitive lab authoring tools allows VTE to provide a completely user-driven virtual lab system. Content creation and consumption is all handled by the software and users, allowing learning to proceed at a pace set by the user community, rather than that allowed by an overseeing organization.

REFERENCES

Bulbrook, H. 2006. Using Virtual Machines To Provide A Secure Teaching Lab Environment. J. Comput. Small Coll. 20, 1 (Oct. 2004), 332-346.

Greenberg, A. The Emerging Value of Virtual Labs: Addressing the 'Pressure Points' in Technical Training. Wainhouse Research. November, 2004. www.wainhouse.com/files/papers/wr-**virtuallabs**.pdf

Hu, D., Wang, Y. Teaching Computer Security using Xen in a Virtual Environment, vol., no., pp. 389-392, April, doi:10.1109/ISA.2008.18

Hu, J., Cordel, D., Meinel, C. A Virtual Laboratory for IT Security Education. Proceedings of the Conference on Information Systems in E-Business and Egovernment (EMISA), Luxembourg, 6-8 Oct 2004, pp. 60-71 http://subs.emis.de/LNI/Proceedings/Proceedings56/GI-Proceedings.56-5.pdf

Nabhen, R. and Maziero, C. 2006. Some Experience Using Virtual Machines for Teaching Computer Networks. *International Federation for Information Processing (IFIP).* 210 (2006), 93-104.

Robila, S. A. 2005. Distributed computing and computer security education. In *Proceedings of the 6th Conference on information Technology Education* (Newark, NJ, USA, October 20 - 22, 2005). SIGITE '05. ACM, New York, NY, 383-384. DOI= http://doi.acm.org/10.1145/1095714.1095803

VMware. 2006. Virtual Lab Automation: A Quantum Leap in IT Cost Reduction and Application Development Process Improvement. *White Paper*.

Vollrath, A. and Jenkins, S. 2004. Using virtual machines for teaching system administration. *J. Comput. Small Coll.* 20, 2 (Dec. 2004), 287-292.

Yang, T. A., Yue, K., Liaw, M., Collins, G., Venkatraman, J. T., Achar, S., Sadasivam, K., and Chen, P. 2004. Design of a distributed computer security lab. J. Comput. Small Coll. 20, 1 (Oct. 2004), 332-346.