Leveraging Next-Generation Virtualization Technologies for Advanced Malware Analysis in the Classroom

Dale C. Rowe
Brigham Young University
dale_rowe@byu.edu

Laura Wilkinson
Brigham Young University
lauraw@byu.edu

Abstract

Malicious software (malware) represents an ever-increasing threat to our technological lives. In the last decade, the proliferation of malware on traditional computing devices has expanded to include mobile platforms and embedded technologies. The development and use of malware is no longer limited to computer scientists and hackers but is now becoming an integral operational capability of militaries and governments worldwide. Recalling that in 2003, a primitive MS-SQL worm resulted in the shutdown of a nuclear plant, the threats faced today and in the near future are alarming. In 2013, it was estimated that the business cost of malware exceeded $114 billion.

Technology, computing, and engineering students of multiple disciplines can be better prepared to deal with malware risks by a comprehensive study of malware. Traditional pedagogical methods typically involve isolating computers and/or networks to enable students to learn without posing a risk to connected networks. While this method does provide a relatively safe environment, modern malware is frequently dependent on a complete network connection, and isolation is no longer representative of current best practices in malware analysis.

In the last year, we have been developing a new course in malware analysis that uses innovative methods of infrastructure-as-a-service (IaaS) and network-as-a-service (NaaS) technologies to enhance student learning in an instructor controllable and inherently safe environment. We show how these approaches are leveraged to allow a variety of dynamic and static analysis techniques and how we have optimized this approach for a typical classroom schedule. We also demonstrate in detail how this solution can be implemented on a limited budget using low-cost surplus hardware. In conclusion, we contrast our implementation with traditional approaches and discuss its benefits and limitations.

Introduction

There is some debate over the precise timing, author, and indeed definition of the first computer virus; however, most will agree that the first discovered in the wild and utilizing removable media was “Elk Cloner,” written by Rich Skrenta in February 1982 at the age of

15. The virus was able to self-replicate using the boot sector of floppy drives and consisted of a payload that on occasion displayed messages during the system boot process.

In stark contrast, a recent report by McAfee shows almost 12 million new virus samples discovered in a Q4 2012 with over 3 million utilizing illegitimate public key certificates to pass as an authentic application [1]. Malware payloads today target the confidentiality, integrity, and/or availability of both services and information for a variety of end-goals that include the acquisition of intellectual property and extortion of ransom payments for user data.

As society becomes more dependent on technology in every aspect of life, criminals have a target that is becoming increasingly easier to attack. A modern cyber-criminal can use computer programs to, with relative anonymity and ease, conduct cyber-crime on an immense scale with an extremely low risk vs. reward outcome; yet as our technological lives continue to evolve, there is no sign that these risks will do anything except continue to grow. It is in the face of these alarming trends that we present our approach for advanced malware analysis and propose a wider uptake of the topic by computing disciplines [2].

Although this paper focuses on malware education within the information technology discipline [2], we believe that other disciplines can benefit from, and should consider incorporating, some of these ideas within a dynamic learning environment.

Pre-Requisite Knowledge

The foundation and strategy for our approach are rooted within the ABET Information Technology (IT) Model Curriculum [3]. While this is certainly not the only domain within which malware analysis may be located, it does provide some unique advantages when compared with traditional computer science programs. As an applied discipline, IT suggests a holistic approach to computing technology and does so from an applied perspective; these naturally lead to a systems engineering aligned model [4]. It has been our experience that systems engineering is a natural approach for IT students and provides an excellent baseline for malware analysis.

The pillars of an IT education include programming, networking, human computer interaction, databases, and Web systems. Each component pillar is connected by a common foundation and a pervasive security theme as shown in Figure 1. Before a strong understanding of malware analysis can occur, students must possess knowledge of certain foundational topics and be skilled in their application. These include an understanding of user and kernel operating system modes, the C compiler process, familiarity with instruction sets, registers, opcodes and their notation, and a solid understanding of networks with experience in OSI layer 2 and 3 configuration. Knowledge of a scripting language such as Python or Ruby is also advantageous.

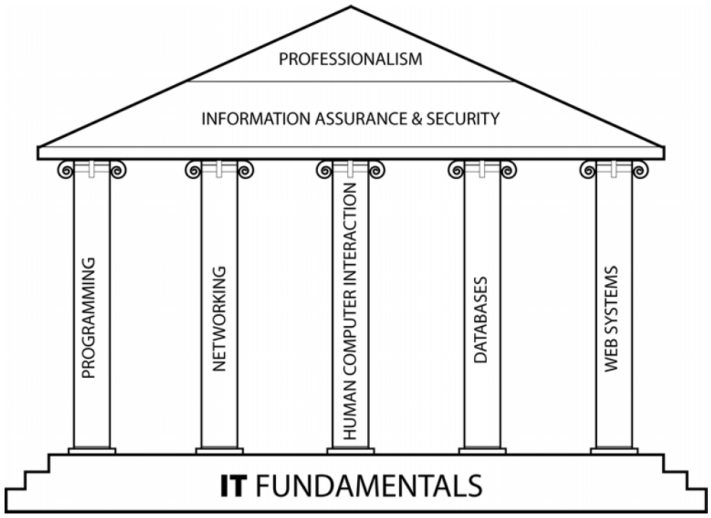

Figure 1. Pillars of an IT education [3]

Learning Outcomes

The focus of our course is to understand the malware process through the stages of creation, infection, operation, discovery, analysis, mitigation, and remediation. Table 1 lists the course learning outcomes.

Table 1. Learning outcomes

1	Analyze a computing system in an unknown state and determine the state of malware infection
2	Employ dynamic and static analysis techniques to determine the characteristics of suspected malware.
3	Identify, analyze and classify various types of malware
4	Derive effective strategies to mitigate the impact of malware and be able to evaluate their relative strengths and shortcomings.
5	Analyze vulnerabilities that may give rise to malware infection and be able to apply countermeasures to prevent infection.
6	Understand a variety of techniques used by malware to self-conceal or hide from analysis.

Instructor Objectives and Rationale

One of the challenges of many computing-based disciplines is maintaining relevancy with current technology. It is the nature of the domain that rapid obsolescence is a way of life and while the persistence of methodologies allows for a form of academic continuity, this is, at times, challenged by the rapid rate at which technology evolves. Hence, in the development of any course with envisaged longevity, particular care should be taken to minimize the course-maintenance overhead.

When discussing the topic of malware analysis, the issue becomes even more problematic. Unlike topics such as networking or operating system in which significant evolution is often seen in intervals ranging from a few years to a decade, the rate of malware evolution may be measured in months to a year. Indeed, simply understanding the scope of the malware problem may be problematic as shown by the discovery of the Flame virus in 2012; in this specific case, researchers discovered evidence suggesting that the malware had been operating completely undetected for over five years [5].

One technology in particular that has proven useful in malware analysis is virtualization [6]. The ability to effectively "sandbox" an operating system within another allows analysts to examine malware with relative safety from accidental infection. In more advanced cases, multiple operating systems may be virtualized simultaneously on a host while also providing isolated networking capabilities. The advantages offered by these approaches are significant: analysts have the ability to revert to previous known states at any time, emulate network and communications protocols, easily attach kernel-level debuggers, perform forensic file-system analysis and compare system states, to name but a few [7].

While these techniques have proven effective for experienced analysts, they still present a significant overhead in a classroom of several students. Experience has shown that even with the best of instructions, well-intended students may be careless in their execution. For example, in a closely related penetration testing course, mistakes such as inadvertently connecting to a wireless network and performing scans have been known to happen. While steps are taken to minimize the occurrence and severity of such errors, malware can present an even greater danger to connected networks.

Summary objectives may be derived as follows:

1. Students must have complete access to all levels of the operating system.
2. Students must be able to create virtual networks.
3. Instructors must be able to minimize the probability of malware leakage onto adjacent networks.
4. The system should not cause security events outside of the lab environment (e.g., IDS alerts).
5. The system must provide a low-maintenance sandbox environment that can be rapidly updated and cater to new malware variants swiftly.

The following are methods that we have previously attempted and their limitations:

Physical/Logical Isolation

Physically isolating the lab by removing the uplink connection or by logically isolating the lab VLAN has succeeded in protecting infrastructure from directly attached systems but often results in frustration at the inability to access online resources. It has been our experience that many students prefer to undertake labs on their own equipment, which often results in user-owned devices being simultaneously connected to the lab and campus wireless networks. This circumvents the isolation and re-introduces the risk of accidental malware propagation.

Lab Filtering

An attempt was made to place a firewall between the lab and outside network. However, it was found that this took significant effort to maintain and would often result in disruption to other courses sharing the same lab space. The issue of students using their own equipment and creating bridges to wireless networks remained a risk.

Dedicated Hypervisors

Perhaps the most secure solution that prevents the network bridging issue is to run virtual machines on equipment owned and managed by the department. Unfortunately, this introduces a new risk by moving the threat directly onto one of the systems we are required to protect. Even if dedicated hardware is used, this solution still presents risks of the inadvertent spreading of malware through the network. It should be noted, however, that for classroom instruction, this approach has a significant advantage in its ability to provide a constant, uniform environment that is well suited to a classroom environment. All students should see the exact same information in a debug window as the hardware, OS, and patch levels are consistent.

The Solution: Software Defined Networks

A recent "cloud" hot-topic is network virtualization. The concept of network virtualization, more commonly known as software defined networks (SDNs), is similar to that of platform virtualization in that it allows the abstraction of configuration from hardware. More specifically, SDNs allow the rapid creation of networks of networks and their association with virtualized operating systems [8] by adding a network control layer as a software-based management component.

SDNs in general are a relatively new approach to networking and in contrast to server virtualization are still viewed as immature. However, the use of SDNs is growing rapidly, and virtualization vendors such as VMWare, Citrix, Oracle, and Microsoft are implementing SDN capabilities into their commercial products. We have found that SDNs bring significant benefits to an educational environment and seem particularly well suited to handling learning activities that can pose security risks to their surrounding environment.

Using SDNs allows an instructor to maintain complete control of a student's network connections on a virtual PC. In this scenario, virtualized computers are hosted on a department-owned hypervisor with students being provided virtual console access to the system. SDN allows the creation of logically isolated network segments with highly controllable access to the outside world without impeding lab connectivity.

It also facilitates the capture of network traffic at a single one-to-many point as opposed to workstation deployments that require local capture for each student.

Requirements

Vendor-supported SDN is a relatively new feature of systems such as Cisco Unified Computing Solution (UCS) and vCenter Cloud Director. These systems tend to be too expensive for many programs. We have implemented our solution on a variety of hardware types and found it to be feasible on any platform that is capable of running a modern hypervisor such as Hyper-V or VMWare ESXi. In our current deployment, the system uses six HP-BL460-G1 servers equipped with dual Xeon 4-Core CPUs with 32-GB RAM each. These are connected to a Synology DS1610+ NAS system providing shared storage using the iSCSI protocol. These servers will comfortably support up to 30 students at an acceptable performance level with each student using 1 VM.

Architecture

The system's architecture consists of three elements or control planes, each of which must be managed in coordination with the others: the hypervisor, SDN layer, and egress/ingress. The term "infection zone" will be used to describe the malware analysis hosts and infrastructure.

Hypervisor

Our initial intent for the hypervisor was to employ VMWare vCenter 5.5 to manage ESXi installations on the physical hardware. VMWare is probably the best known virtualization platform and benefits from extensive support. Unfortunately, during our testing, we ran into issues with academic licensing restrictions that prevented us from creating a usable operational environment despite being successful in prototype designs. This led us to investigate different offerings that may provide a similar capability and culminated in a selection of Microsoft Virtual Machine Manager (VMM) 2012R2 with HyperV 2012R2 for several reasons.

Our analysis found that the out-the-box VMM feature set was much more comprehensive without requiring advanced licensing options. This became advantageous, given that our institution's MSDNAA agreement covers the use of all Microsoft System Center products and does not limit use within educational infrastructure. Effectively, this set at zero the software costs for hypervisors and their management. It should be noted, however, that the initial configuration of VMM is a much more complicated affair than vCenter.

SDN Layer

The objective of the SDN layer is to rapidly deploy dynamic LAN segments and connect these securely to the IT network while providing isolation for malware. This required a centralized model for configuring a variety of network equipment including

- Blade switches (Cisco 3020 HP)
- Router (FastIron Edge X448-Prem)
- HyperVisor networking (vDS Switch or HNV SDN)

The proposed solution uses VLANs to provide layer 2 separation in combination with the Ingress/Egress plane to provide controllable routing and deep packet analysis capabilities. We defined a range of unused VLANs and subnets within our IP space that could be rapidly setup and torn down by PowerShell and Python scripts and configured a DHCP server to allocate addresses within each range as these networks are created. SDN provisioning is achieved by a Web front-end that provides instructors with complete control using simple mouse actions provision, move and/or tear down an entire live network range in real-time.

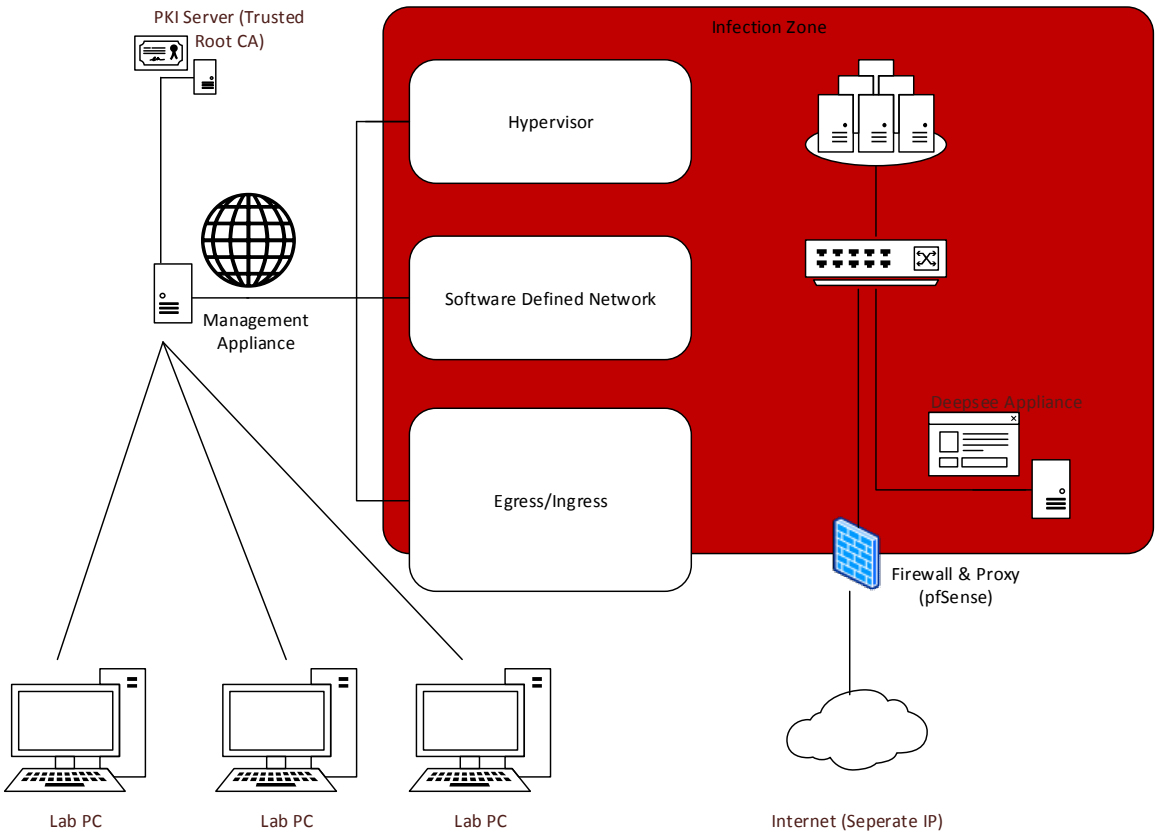

Figure 2. System architecture

Egress/Ingress Plane

Perhaps the most critical part of the system is the connection point to the rest of the department's network. It should be noted that our program already exists on a separate IP range from the rest of campus, thus inherently affording other departments a level of protection. We prototyped the system using a simple IPTables firewall running on Debian 7.1. However, we found a simpler implementation in pfSense, an open source firewall appliance that offers several advanced features.

Of particular interest in the pfSense appliance is the ability to perform Web proxying and SSL inspection to decrypt encrypted traffic. The latter requires the addition of a trusted root certification authority on each client. We have implemented this as part of our virtual machine

template and connected a Bluecoat Security Analytics Platform (formerly Solera Deepsee) virtual appliance to a mirrored traffic port to record all network traffic attempts from the infection zone. The use of the analytics platform in this way provides real-time deep packet logging and analysis capabilities to students.

To minimize alerts from campus intrusion detection devices, the campus IP address of the infection zone egress/ingress is known and provided to campus IT services. This allows for casual monitoring of activity without creating panic every time a sample sends traffic across the network.

The principal advantage of the SDN approach is the ease with which a classroom virtualized network may be connected to, or isolated from, the physical network. The instructor may, using simple scripts, connect an SDN VLAN to the physical network, allowing students to download tools, scripts, and perform updates. We have also employed firewall configuration scripts that limit access to HTTP/HTTPS traffic only to known sites, offer full HTTP/HTTPS to any site, or offer unhindered external access to all services. Once a student is ready, the malware sample is made available and the SDN disconnected from the physical world and instead connected to a catch-all DNS service and network traffic monitoring devices. Thus during the analysis, the student-managed virtual environment has no network traffic route to a physical network. Students are able to manage their environment by a console connection to the hypervisor.

Care must be taken in this approach to ensure the hypervisor is regularly patched to avoid any traversal of malware from the guest environment to the host physical system. We believe this is currently an acceptable risk and are aware of no malware currently available that is capable of attacking a Microsoft HyperV host from within a virtual machine.

A benefit of this arrangement is that students may safely access and manage the infected virtual machine from any lab system within the department network, their own personal laptop, or a home PC connected via a VPN. This greatly increases the valuable out-of-class time students can spend analyzing malware. We have also found this approach to be naturally supportive of malware research projects.

Systems Management

The management appliance is under development and acts as a provisioning and management portal that allows a simple drag-and-drop style arrangement of network segments, virtual machines, and firewall configurations. All management is performed out-of-band using separate network interfaces on each server and/or network device.

An instructor will be able to manage a library containing VM templates of standardized configurations and deploy en-masse as required for class exercises or laboratory assignments. Deployed VMs must be assigned to a VLAN and IP subnet, which are taken from a pool of available containers. The container is dragged to the main window and a template placed within it. The instructor may then specify how many VMs to create and either assign a pre-defined firewall template, manually configure firewall settings, or disconnect the SDN from the physical network.

After deployment, students may connect to the management consoles of provisioned virtual machines. This provides an additional layer of protection to lab PCs by providing distinct data paths for infection zone traffic and infection zone management. Students are also provided access to the Security Analytics Platform instance and firewall logs to examine traffic flows and perform network forensics.

Configuration and Evaluation

At the time of writing, the system has been through multiple levels of testing and design ratification and is now in the final integration stages. Backend management scripts and architectural testing have been completed, and the management appliance is near completion. We intend to conduct large-scale testing in fall 2014 with the system ready to support a new malware analysis course in winter 2015.

One of the advantages of the VM-library and SDN approach is its ability to remain up-to-date and analyze new samples on an ongoing basis without jeopardizing network security. Currently, the library includes pre-configured virtual machine templates for all major Window versions between Windows 95 and Windows 8.1 including both 32-bit and 64-bit editions and server counterparts where appropriate. Additionally are images for a variety of Linux and OSX-based systems over a 10-year timespan. Unfortunately, due to licensing restrictions, the OSX VMs may only be deployed on Mac hardware, and we are working to find a solution to this problem. We have also included several relevant security distributions of Linux such as Kali, Remnux, and SIFT.

The library also includes over 2,000 malware samples, although currently access is restricted to instructors only for the complete dataset. Finally, a variety of standard firewall configuration scripts are also included which include blocking all traffic, allowing HTTP/HTTPS only (with full SSL inspection), allowing common Windows/Linux protocols (such as SMB/NetBIOS) and unrestricted access. All of these may be selected via the Web drag-and-drop interface.

Fusion of Research and Education

The architecture described in this document also benefits a fused learning and research environment. While instructor-guided walk-throughs can be extremely useful in acquiring malware analysis skills, students also benefit from self-guided analysis and often thrive when they pursue their own choices. We propose guiding students to several online malware repositories and setting research projects to analyze samples not discussed in the course using the SDN based system. We plan to evaluate this during the first iteration of our course and report on its effectiveness along with any issues discovered.

Future Work

The use of SDN in a malware analysis environment allows a great amount of flexibility at the networking layer. We envisage expanding the architecture to allow for network layering between the analysis VM and the egress/ingress firewall. This would allow the insertion of in-line network devices such as intrusion detection systems, honeypots, and traffic emulators. It would also facilitate research into the effectiveness of such devices in a malware detection and mitigation scenario.

Conclusion

In this paper, we have presented a novel approach for utilizing software defined networks in a malware analysis classroom environment. It is believed that using SDNs in this manner will decrease the overhead in maintaining courses and open avenues to project-based research in an undergraduate environment. The security and usability advantages of the system have been discussed and compared in relative terms to existing approaches of malware analysis that rely on both standalone and networked hosts.

In conclusion, we maintain a positive outlook for the prospects of SDN in malware analysis education and believe that the capabilities it brings will be revolutionary in better preparing students to understand and defend against this ever-increasing threat.

References

[1] McAfee Labs. (2012). *McAfee Threats Report: Fourth Quarter 2012*. Retrieved from http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q4-2012.pdf

[2] Ledin, G. (2011, February). The Growing Harm of Not Teaching Malware. *Communications of the ACM*, *54*(2), 32.

[3] Lunt, B, M., Ekstrom, J. J., Gorka, S., Hislop, G., Kamali, R., Lawson, E., LeBlanc, R., Miller, J., & Reichgelt, H. (2008). *Information Technology 2008—Curriculum Guidelines for Undergraduate Degree Programs in Information Technology*.

[4] Squires, A. F., Ferris, L. J., Ekstrom, T. J., VanLeer. J. D., & Roedler, G. (2012). Defining the Core Body of Knowledge (CorBoK) for a Graduate Program in Systems Engineering: A Work in Progress. *Proceedings of the 2012 ASEE Annual Conference*.

[5] Zetter, K. (2012, May 28) Meet "Flame," the Massive Spy Malware Infiltrating Iranian Computers. *Wire*. Retrieved from http://www.wired.com/2012/05/flame/

[6] Latorre, G., & Flores, D. A. (2013). Reverse Engineering: How to Create a Basic Environment for Malware Analysis Oriented to Undergraduate Students. *Ibero-American Journal of Computing of Systems Engineering, National Polytechnic School, Ecuador*, *1*(2), 4–7.

[7] Sikorski, M., & Honig, A. (2012, February). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. San Francisco: No Starch Press.

[8] Kim, H. & Feamster, N. (2013, February). Improving Network Management with Software Defined Networking. *IEEE Commuication Magazine*, *51*(2), 114-119.

Biographies

DALE C. ROWE is an assistant professor of Information Technology at Brigham Young University and director of the Cyber Security Research Laboratory. He maintains a variety of security certifications including a CISSP. Dr. Rowe's scholarly interests include anything security related and he enjoys keeping his technical skills up to date. In 2011, he created and maintains a student Red Team that frequently conducts penetration tests as a service to the local community. In the past 3 years, he has mentored 4 cyber defense (CCDC) student teams who have received twice 3rd place, 2nd place, and 1st place in regional contests. Prior to joining BYU in 2010, he worked as a systems security architect in the aerospace industry. Dr. Rowe may be reached at dale_rowe@byu.edu.

LAURA WILKINSON is a junior student at Brigham Young University majoring in Information Technology with an emphasis in cyber security. She recently began a summer internship in California as a penetration tester for Bishop Fox. Her interests lie in big data, penetration testing, and malware analysis. Before BYU, she worked for Symantec in Utah. Laura can be reached at lauraw@byu.edu.