Global Information Assurance Certification Paper

Using Virtualization in Internal Forensic Training and Assessment

GIAC (GCFA) Gold Certification

Author: Courtney Imbert, courtneyimbert@gmail.com
Advisor: Chris Walker
Accepted:

Abstract

The field of forensics requires continual, on-the-job training in new devices, techniques, and file systems. Virtual environments offer unique advantages in digital forensics investigation. This paper will explore the ways information security professionals can use virtualized environments and simulation to reconstruct a sample event or device, distribute them throughout a team, and manage them as a tool to provide small scale training and assessment to forensic employees or teams.

1. Introduction

Continual training is a critical part of forensics work. Formal training and education in forensics are irreplaceable, but training has the most value when supplemented with hands-on laboratory work to reinforce concepts and apply practical skills (Ananthapadmanabhan, Frankl, Memon, & Naumovich, 2003). This is known as "scenario based training" (SBT) and is an important part of the active learning process (Norton, et al., 2012). A survey has shown that IT learners, in particular, have a preference for "kinesthetic learning" - learning by doing (Benoit, 2012).

Similar to physical crime scene investigation, part or all of a digital case can be reconstructed using a virtual environment. Reconstruction is proving helpful for examining, classifying, determining, and testing sequences of digital events (Arnes, Haas, Vigna, & Kemmerer, 2006). Guided by learning objectives, the use of a similar process can also serve the purpose of providing a low-risk, controlled environment for learning new forensics skills.

Simulations enhance trainee of engagement compared to traditional learning activities (Crellin, Adda, & Duke-Williams, 2010). In a compendium of recent studies, business students who learned through simulation either scored higher on exams or felt that they learned more than students in traditional lectures (Faria, 2001). Through simulations and other learning "games", trainees develop strong links between theory and practice.

Off-the-shelf online digital forensics scenario-based training is increasingly popular. Many organizations, software providers, and professional groups have created free simulations or sets of data for the purpose of practice, training, assessment, and competition. Some examples include those shown below:

- Digital Corpora (http://digitalcorpora.org/corpora/scenarios/)
- NIST's CFReDS Project (http://www.cfreds.nist.gov/)
- WireShark (https://wiki.wireshark.org/SampleCaptures)
- DFTT (http://dftt.sourceforge.net/)
- SANS DFIR NetWars (https://www.sans.org/netwars/dfir-tournament)

Courtney Imbert, courtneyimbert@gmail.com

There is no substitute for formal forensics training and education, especially regarding forensics fundamentals and theory. However, sometimes formal training cannot fulfill the immediate needs of the forensics team. Teams sometimes need to "fill in the gaps" on specific technologies or software with in-house training. Small scale, custom developed training and assessment scenarios can be helpful for learning custom, brand-new, or boutique technologies and skills, or for getting a new team member up to speed rapidly.

This paper provides a possible framework for creating a small scale virtual simulation for the purpose of in-house forensic training. The process of creating a training program and simulation can quickly shift into a deep dive in learning, instructional game design, training, and assessment theories and best practices! Therefore, this list of steps is focused on a small scale, practical process: a rapid development, Agile-style method for creating informal, standalone training modules based on simple forensics objectives and outcomes. The framework for creating a virtualized forensics simulation is based on the author's firsthand experience creating environments for the purpose of small scale training and assessment, research into the development of digital forensics virtual laboratories, and interviews with others who have developed in-house forensics or information security training.

Virtualization offers some exciting opportunities for training developers. In addition to being portable between systems (even remote ones), virtual environments provide manageability and several other benefits. Therefore, although the development process could apply to a hardware-based training center as well as a virtual one, the development process shown here is described in the context of a virtual environment.

2. A Brief Overview of Virtualization

A hypervisor, sometimes known as a virtual machine monitor (VMM), is a software package uses underlying physical hardware to emulate an operating system (Brueckner, Guaspari, Adelstein, & Weeks, 2008). Typically, the emulated operating system is called a *guest machine,* while the underlying operating system and hardware are referred to as the *host machine.* Hypervisors and hosts can emulate multiple guests with a variety of operating systems. Type 1 hypervisors run guests directly on host hardware. Type 2 hypervisors are software packages that abstract guest machines through a conventional operating system. Trainees are

likely to complete forensic training on a variety of computers that are not fully designated as virtual machine hosts. Therefore, Type 2 hypervisors are better suited to the small scale, individually distributed forensic simulation discussed here.

Some examples of Type 2 hypervisors include VMWare Workstation or Server, Oracle Virtual Box, and Microsoft Virtual PC (Vanover, 2009).

Though the simulation development example focuses on virtualization of traditional devices (e.g., computers), it is also possible to create a virtual simulation of mobile phones and other non-traditional devices. For information on this, see Appendix B.

3. Benefits and Drawbacks of Virtualized Simulations

The most obvious benefit of virtualization is the lower resource cost. Virtualization is frequently used in crime scene reconstruction and research, since it is highly flexible and requires far fewer resources than a similar physical configuration (Arnes, Haas, Vigna, & Kemmerer, 2006). Most Type 2 virtual machines do not require specific hardware configurations, so trainees can use their own computers for training rather than having to budget for new ones. In addition, though the initial cost of creating a training scenario may be high due to subject matter expertise and time, the one-time cost is distributed over multiple replays (Crellin, Adda, & Duke-Williams, 2010).

While physical simulations may be the most accurate representations of forensic cases, they are often impractical to manage and distribute to a team. With virtualization, a small network of virtual devices and resources can be passed throughout a team with just a link to the cloud. A virtual training module can be portable, completed at an employee's convenience at any location.

Simulations are designed to be "safe", providing many of the effects of a potentially high-stakes scenario without the fear of failure or damage. Virtual machines are particularly well-suited to this. They have snapshots - a system and data captured and stored at a particular moment in time. Snapshots are helpful for capturing moments like the start of an incident and key points throughout. This feature permits learners to test a hypothesis or technique with lower

risk. If the trainee's attempt fails, he could revert to a previous machine state and try another approach, creating a "disposable" learning experience (Shavers, 2008).

Virtual scenarios make it simpler to collect performance and assessment data after the simulation is completed (Hancock, Vincenzi, Wise, & Mouloua, 2008). Trainers can have complex, formal assessment tools that monitor the trainee's progress and provide support. Trainers can also answer questions about the tasks they completed, or simply have trainees submit their virtual machines and resulting data. "Resetting" the training for the next group requires only sending links to copies of new virtual machines.

However, virtualization is not a panacea for forensics training. Many of the same elements that present a challenge in a physical environment continue to exist in a virtual one. Restrictive licensing, staffing, and hardware can continue to be expensive and prohibitive, even in a virtual environment (Scott & Koonts, 2013).

Not all learning objectives are best suited to a virtual environment. The designer must determine if virtualization is appropriate for the objectives of the training. For example, a learning outcome of connecting to a hard drive and collecting a physical image from it would not be an appropriate target for virtualization. Some malware even has anti-forensic capabilities that prevent it from running properly in a virtual environment (Zeltser, 2009), making it more difficult to simulate accurately in a VM.

Virtualized environments provide the most benefit in training objectives that involve behavioral analysis of volatile artifacts or other "live" analysis. Most static file analysis can be simulated without creating a virtual environment, for example by providing an image file. However, a virtual machine can easily be packaged together with related images, log files, and other components to create a realistic, comprehensive "case file" for the trainee.

4. Steps to Creating a Small Scale Virtual Simulation

In the tradition of Agile programming, these steps can be followed iteratively and seemingly out of order. For example, testers can review walkthroughs while a designer develops the simulation using ongoing test results to refine the script. The designer may already have access to a case he's sure his team will encounter again, using the data in his scenario and taking

it into account when creating the conceptual model. It is best to read through these instructions and use them to create a training development plan, rather than following them from start to finish.

4.1. Define Learning Objectives

The first step in developing a training module for a forensics team is to determine the objectives of the training. A team should perform a "needs analysis" and prioritize training in areas that the target audience has a knowledge gap in, that present a high risk, and that the target audience will encounter most frequently (Jones & Valli, 2009). Because the area of training development is a vast one, additional resources for creating and working through training plans are included in Appendix A.

A new technology, technique, or recent forensics case may make the knowledge gaps and therefore the learning objectives, clear. For example, if the forensics team acquires a new tool, a simulation could be used to assess or practice the use of the tool. A new model phone found in the field might require a team to learn and practice new techniques. In fact, the training designer may even start with a sample data set - for example, a recent case file - that can be used to help guide the design of the simulation. In some cases, an internal change in procedure could also spur a need for training.

The scenario most commonly encountered in technical fields like digital forensics is *problem-based* (Errington, 2010). This type of scenario requires the learner to integrate theory with practical knowledge to solve a problem or complete a task. It typically requires skill in decision-making and critical analysis.

Here is a sample list of forensics learning objectives for a beginning examiner, to be used as an example throughout this paper:

- Search through locally stored email artifacts, using EnCase
- Identify the date a file was created on a computer
- Find a malicious process masquerading as a system process on a live system

It is entirely possible for a simulation to have a single objective. Fewer clearly-defined objectives make a simulation faster and easier to develop and maintain over time. With single-

objective exercises, an organization could even build a library of modular exercises to train on as they are needed, with a common underlying infrastructure.

Whatever the objective, it is important for the designer to understand the full technical details, or work closely with a subject matter expert who does. Because scenario-based training is interactive, a learner has many more options than a recorded lecture or multiple-choice exam. Learners can find an inefficient way to complete the objectives, become confused with a lack of guidance, or find themselves sidetracked by unexpected components. One or more experienced subject matter experts can help create meaningful learning objectives, and avoid potential pitfalls later in the process.

4.2. Create a conceptual model

Once the designer finishes developing learning objectives, it's time to create a premise for a scenario that connects the learning objectives together into a story or "game". Simplicity is best, particularly for beginning training designers. No model can perfectly replicate a live event; for the sake of training, it is enough to include just enough components to link to a list of learning objectives (Becker & Parker, 2012).

Here is an example of a premise that utilizes the list of learning objectives in our example:

> *An attacker emails a malicious file to a victim. The victim downloads and runs the file, resulting in a malicious hidden process.*

The premise is then expanded into related events and outcomes. The designer can document these with a table, a flowchart, or storyboard. As shown in the following table, each learning objective is linked to a trigger event in the scenario. Trainees can detect trigger events by fulfilling a forensics training outcome - a task that they can practice or learn with the help of live or self-directed training.

Conceptual Model: "Forensicating" a Phishing Attack

Learning Objective	Trigger Event	Outcome
Collect and search through local email artifacts	Attacker emails a malicious file to a victim	Locate .PST local mail files in a Windows 8 computer with Microsoft Outlook 2013 Use EnCase to load the PST, expand or index it, and locate a specific email with a word search
Identify the timestamps of a file on a computer	Victim downloads a file from an email	Use EnCase or another acceptable technique to locate and document the metadata timestamps of an executable stored in the "Downloads" folder
Find a malicious process masquerading as a system process	Victim runs the executable, resulting in a persistent running process	Use Process Monitor against the victim computer to identify information about a running rogue process

Table 1: Learning Objectives, Events, and Outcomes

By creating this information, the designer creates a master document he can refer to as a guide while developing the script and written training materials, as well as a checklist he can use to assess the effectiveness of the training against the learning objectives. A qualified trainee should be able to complete the tasks associated with each objective. In a well-designed, simple scenario, each task is compartmentalized; if a trainee fails at one task, he should still be able to complete the other ones.

4.3. Create a script

At this point, it is time to create an "operational model" - a list of players, actions, elements, and behaviors that the simulation will include. Simulations are typically ordered according to a sequence of events (Becker & Parker, 2012), so it is most helpful to create a

"script" of actions that the developer can follow to generate the scenario. This script can be created using a spreadsheet, flowchart, storyboard, or checklist. For example:

Order	Who	Action	Computer
1	Attacker	Creates a phishing email to send to the target via Anonymous-Mail-Server, attaching file "salaries.xlsx.exe"	Attacker-PC
2	Victim	Opens email in Outlook, double-clicks attachment, saves to download folder	Target-PC
3	Victim	Double-clicks attachment to open it (attachment creates persistent, rogue svchost process)	Target-PC

Table 2: Simulation Script

4.4. Design the requirements and architecture

There are potentially three different categories of guests in a virtual scenario: attackers, targets, and third parties, such as logging servers or certification authorities (Arnes, Haas, Vigna, & Kemmerer, 2006). Target networks can be as simple as a single computer exposed to the rest of the network, or they can be more complex, hidden behind a virtual gateway with security measures in place.

In the requirements, the designer must include a method of collecting data related to the scenario. Are there any log services that should be on while following the script? Should a device collect network traffic during some or all of the events? The data to be collected should be the data the trainee will use to fulfill his objectives. In addition to the hosts related to the scenario, there may be designer-controlled guests that perform services like automated monitoring, logging, and assessment. The trainee will also need access to an analysis host, a device used for performing the investigation. This may already exist on the trainee's computer, or he may need to be provided with it as part of the training package.

Here is an example of a list of resources, using the preceding steps as a guide:

Name	Description	OS / software	IP Address	Visible to trainee?
Attacker-PC	Attacker's computer (attacker)	Kali Linux Email client	192.168.2.3 (vmnet1)	No
Target-PC	Target's computer (target)	Windows 7 Home Outlook	192.168.3.2 (vmnet2)	Yes
Anonymous-Mail-Server	Mail server (3rd party)	Ubuntu +Postfix	192.168.2.2 (vmnet1)	No
Trainee-PC	Trainee's computer, to be used for forensic analysis (host)	SIFT Kit +Encase	N/A	Yes

Table 3: Resource Planning

As in a real forensic investigation, it is typical that the forensic expert has access to devices belonging to only a subset of the affected parties in an incident (Brueckner, Guaspari, Adelstein, & Weeks, 2008), which is why there is a column listing the visibility to the trainee. Though the Anonymous-Mail-Server and Attacker-PC will be created, used, and stored, they will not be included in the training package.

Virtual machines share hardware resources, a trait that makes them efficient to use in a low-resource setting. Unfortunately, that characteristic can also make them liable to compromise the integrity of other VMs, or even the host machine. As in any forensic investigation, the designer must make certain to minimize risk and prevent cross-contamination between host and guest systems and networks (Zeltser, 2009). Among other security measures, this can be done by using the analysis host only for designated tasks, maintaining security patches throughout the system, disabling shared folders, and configuring the virtual network to disconnect it from the host and other external networks (Arnes, Haas, Vigna, & Kemmerer, 2006).

As discussed in section 2, there are several hypervisor options available for creating a simulation. The designer and trainees should both use a similar software package to minimize technical support problems.

4.5. Implement the scenario and collect data

Next, it is time to create the scenario with the finalized conceptual and operational models. The first step is creating the environment required for the scenario, using the requirements document. This step is often done by creating a virtual machine and allocating resources to it, installing an operating system, installing additional software or updates on the operating system, and configuring it for the scenario. Many open-source operating systems have VM images available for download, optimized for virtualization. It may not even be necessary to create a new VM.

In order to fulfill the events in the scenario script, it may be necessary to play some parts of the scenario semi- or fully automatically using a custom script. Scripts may include operating system scripts, Python, Perl, or other programming language scripts.

It is a good practice to create a copy or snapshot of the virtual machine once it has been created and configured prior to initiating the events in the script. The designer may also create snapshots at milestones or critical points in the script.

Ensure any capture devices or logs are activated. Once the base image is created and configured with logging or capture tools enabled, follow the script, taking notes and screenshots along the way to generate the scenario files and data.

4.6. Review and sanitize data

Once the designer has created the scenario, he should review the generated data and, if necessary, sanitize it, removing sensitive and extraneous information from any trainee-facing scenario files. If the simulation designer is using real-world data for training, he must ensure the data is reviewed and anonymized before releasing it for training or research. Sensitive data could include PII, network addresses, and cryptography keys. Cleanup and editing can also make simulated data more realistic or plausible.

If the evaluation of training is high-stakes, data editing can also increase the integrity of assessment results. For example, the training proctor can present logs with different data values to each trainee, while the tasks and objectives remain unchanged. This practice maintains the

integrity of the assessment, provides signatures for DLP to detect or investigate exam exposure, and protects against collusion (Lorenzetti, 2010).

Developers can easily leave irrelevant remnants of the development process on virtual machines, such as installation files, administrator credentials, and command history. The simulation designer should take care to document temporary files or data throughout the development process, and clear those before distributing the virtual machines to trainees. Designers should also back up a "clean copy" of the completed virtual machine and remove temporary files related to development, like snapshots.

Microsoft or Unix log files can be edited using tools like Microsoft Log Parser (http://www.microsoft.com/en-us/download/details.aspx?id=24659) and lr_anonymize (http://manpages.ubuntu.com/manpages/lucid/man1/lr_anonymize.1.html). In some cases, software includes built-in tools for anonymizing logfiles (Splunk Enterprise, 2015), or custom scripts can be created to substitute values.

Several options exist for editing and cleaning up network packet captures at multiple network layers, with capabilities ranging from changing timestamps, substituting IP addresses, to simply eliminating duplicate packets. Options include editcap (https://www.wireshark.org/docs/man-pages/editcap.html), Scapy (http://www.secdev.org/projects/scapy/), tcprewrite (http://tcpreplay.synfin.net/wiki/tcprewrite), and WireEdit (https://wireedit.com/).

4.7. Create documentation and training materials

As with most projects, documentation is best created and updated throughout the development process, rather than at the end. A documentation package should be visible to anyone involved in the development of the training, and includes all planning, checklists, storyboards, spreadsheets, diagrams, and scripts associated with development, as well as lists of problems, changes, or troubleshooting steps. Training documents and troubleshooting for support staff are also important to the success of the project.

Written training materials include supportive reading, worksheets, and questions. This part of a simulation can take a significant amount of time to develop. The guidance given to trainees can range from an outline of tasks or outcomes to a step-by-step walkthrough of each

activity. The learning objectives developed at the beginning of the process should guide the training materials, and the level of detail should be tailored to the audience. For further resources on developing quality training materials, see Appendix B.

4.8. Verify and validate

Validation relates to the content of the simulation; does it fulfill the learning objectives? Is the scenario realistic? Verification is the process of determining whether the simulation is technically correct and working (Becker & Parker, 2012).

For validation, complete a walkthrough with one or more people who are knowledgeable about the purpose of the simulation and can provide input. Typically, the designer of the simulation steps through the simulation, with a panel of subject matter experts who can offer feedback and ask questions. The designer should complete a walkthrough at every step of the development process, by using flowcharts, storyboards, development documents, or the finished project itself (Becker & Parker, 2012).

The designer can verify the simulation by testing it along every step of the process. After verification is complete, potential members of the trainee pool and subject matter experts should "play-test" the training, making notes of any inconsistencies or errors. As play-testers uncover problems, the problems can be corrected iteratively, but a final play through should not turn up any unexpected glitches.

4.9. Package and distribute the training

Finally, it is time to package the data for storage and distribution. A typical package for a trainee would include a written introduction with base rules and support information, written training materials (including any assessments or evaluations), and virtual machines or technical files. The training developer may provide support for live training or email/phone support for self-paced training.

Development materials, planning documents, and documentation should be safely stored and organized for future reference and training development. Developer can often re-use a "base image" or snapshot of a configured operating system for future development.

Courtney Imbert, courtneyimbert@gmail.com

Distributing virtual machines to remote locations can be a challenge. Virtual machines consistently consume large amounts of disk space, since they contain a full operating system and related data. VMs can easily be 10-15 GB or larger. These can take a long time to transfer over remote connections. There are a several options to facilitate the distribution of virtual machines and large training files.

.It is typical for the size of a virtual machine's hard drive capacity to be determined at the time of creation. Virtual machine platform software may offer some other capabilities, like disabling the pre-allocation of unused disk space, and "splitting" a virtual machine disk into a series of smaller files (VMware, 2015). It is better to plan disk capacity and options in advance during the requirements planning process, to avoid having to recreate a virtual machine or resolve problems later.

After a developer finalizes the virtual machine, he can run the VM or training package through compression software to decrease size for distribution. However, this may result in a single large file that is difficult to distribute through unreliable network connections. Designers may want to consult with their IT group to find the best way to distribute files to trainees. For example, the IT group may make cloud storage available, or schedule downloads for low-traffic time slots.

4.10. Collect and Analyze Training Data

After the training period is complete, it is important to collect any assessment data and feedback, and use that for future development. Debriefing after any and all phases training simulations is shown to increase future performance for all involved parties (Faria, 2001).

After the training is complete, a survey or feedback memo should be issued to trainees. The survey should consist of a maximum of 10-12 questions. The survey can include questions like:

- What have you learned in this training that you did not previously know?
- Does this seem like a realistic scenario?
- Did you encounter any technical problems with the simulation?
- Is the simulation fun? (Trainees are more likely to complete training that is fun.)

Courtney Imbert, courtneyimbert@gmail.com

- How can future training be made more effective for you?

The results of the survey can be used to guide minor changes to improve the experience for the next "batch" of trainees, and or drive the process to improve development on the next simulation.

It is widely considered a best practice to perform an evaluation of learning objectives to assess the effectiveness of training (Edens, Bell, Arthur, & Bennett Jr, 2003). Evaluations do not have to be complex - they may be simple questions at the end of each section of training that can be answered only by completing the assigned tasks and fulfilling the training outcomes. Simpler evaluation results are typically faster to "grade" than more complex results like large data files or reports, and grading can even be automated. However, simple or quantitative answers may be easier to share with other participants, creating the potential for cheating. It is important to have a remediation plan in case training isn't effective for one or more trainees.

5. Conclusion

The increasing amount of available data, the rapid evolution of devices and software, and a shortage of cybersecurity professionals (Libicki, Senty, & Pollack, 2014) contribute to a critical need for timely forensics training.

Though formal training is important, forensics departments need a process to develop in-house training on custom, brand-new, or obscure technology or new techniques. "Hands-on" work in the form of scenario-based training and assessment provides the ability to apply training to real-world situations. For the most effective training simulation, the designer must use a development process, including:

- Determining the learning objectives of the department
- Creating a conceptual model
- Creating a script for the scenario
- Defining training requirements
- Implementing the scenario
- Collecting the necessary data for training, reviewing and sanitizing the data
- Verifying and validating the scenario

Courtney Imbert, courtneyimbert@gmail.com

- Packaging and distributing training
- Closing the project with feedback and evaluation data

These steps are iterative, and can be completed using an "Agile"-style development process.

Simulation training is not easy or cheap to develop, issue and distribute, and can present a challenge for small scale teams. However, it can be made easier with virtual environments. Virtual environments include components that are particularly well-suited to training, like portability and scalability, easy networking, and the ability to revert to a previous state with snapshots.

Courtney Imbert, courtneyimbert@gmail.com

Appendix A. Helpful Training Development Resources

Books:

- **Designing and Developing Training Programs**
  Janie Fisher Chan; Pfeiffer Essential Guides, 2009. ISBN-13 978-0470404690
- **The Fifth Discipline: The Art & Practice of the Learning Organization**
  Peter M. Senge; DoubleDay, 2006. ISBN-13 978-0385517256
- **The Guide to Computer Simulations and Games**
  Katrin Becker & J.R. Parker; John Wiley & Sons, 2012. ISBN-13 978-1118009239
- **How to Measure Training Results: A Practical Guide to Tracking the Six Key Indicators**
  Jack J. Phillips, Ron Drew Stone; McGraw-Hill Education, 2002. ISBN-13 978-0071387927
- **Workplace Learning: Principles and Practice**
  Robert W. Rowden; Krieger Publishing Company, 2006. ISBN-13 978-1575242682

Websites:

- **Developing Clear Learning Outcomes and Objectives, The Learning Management Corp.**
  http://www.thelearningmanager.com/pubdownloads/developing_clear_learning_outcomes_and_objectives.pdf
- **How to Engage Learners with Scenario-based Learning, Learning Solutions Magazine**
  http://www.learningsolutionsmag.com/articles/1108/how-to-engage-learners-with-scenario-based-learning-
- **Training Material Development Guide, Swedish Civil Contingencies Agency**
  https://www.msb.se/RibData/Filer/pdf/26433.pdf

Courtney Imbert, courtneyimbert@gmail.com

Appendix B. Virtualizing Non-Traditional Devices

Increasingly, forensics departments encounter a requirement to collect data on mobile devices, "smart appliances", and other non-computer devices (Murphy, 2009). As a result, forensics teams require frequent training in these technologies, since consumer brands release new hardware versions frequently, and software can be updated weekly or more often. Purchasing and distributing these devices for training can be cost-prohibitive, and difficult to manage centrally.

Fortunately, there are ways to distribute "virtual phones". If the learning objectives support it, the training data can include a pre-collected forensic image of the device, ready for analysis. For dynamic analysis objectives that require a device to be powered on and running, some mobile devices can be emulated in a virtual environment.

The Santoku distribution of Linux (https://santoku-linux.com/) provides a suite of pre-installed mobile device emulators for Android, Blackberry, and Windows phones. It also includes forensic utilities for networking, malware analysis, and decompilation/disassembly of individual applications. The training developer can distribute a Santoku VM as part of the training or use another emulator tool to create a "virtual phone". This enables him to distribute only those virtual files (Santoku Linux, 2015). Here are some emulator options for mobile devices. Though these are primarily designed for application testing, they can be used for forensics and the reverse engineering of malware.

Courtney Imbert, courtneyimbert@gmail.com

Mobile Device Emulators

Mobile Device OS	Emulator	Website
iOS (iPhone, iPad)	XCode iOS Simulator (requires App developer privileges)	https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/iOS_Simulator_Guide/Introduction/Introduction.html
Android	Android SDK Manager	http://developer.android.com/sdk/index.html
Blackberry	Blackberry Smartphone / PlayBook Simulator	http://us.blackberry.com/sites/developers/resources/simulators.html
Windows	Windows Phone Emulator	https://msdn.microsoft.com/en-us/library/windows/apps/ff402563(v=vs.105).aspx

Links to up-to-date resources for mobile device security, including lists of emulators, are available at https://mobilesecuritywiki.com/.

Courtney Imbert, courtneyimbert@gmail.com

References

Ananthapadmanabhan, V., Frankl, P., Memon, N., & Naumovich, G. (2003). *Design of a Laboratory for Information Security Education.* Brooklyn, NY: Polytechnic University.

Arnes, A., Haas, P., Vigna, G., & Kemmerer, R. (2006). *Digital Forensic Reconstruction and the Virtual Security Testbed ViSe.* Trondheim, Norway: Norwegian University of Science and Technology.

Becker, K., & Parker, J. (2012). *The Guide to Computer Simulations and Games.* Indianapolis, Indiana: John Wiley & Sons.

Benoit, N. (2012, May 3). *IT Students Learning Trends*. Retrieved from UAFS.edu: http://info.uafs.edu/blog/bid/143301/IT-Students-Learning-Trends

Brueckner, S., Guaspari, D., Adelstein, F., & Weeks, J. (2008). *Automated computer forensics training in a Virtualized Environment.* Ithaca, NY: Air Force Research Laboratory.

Crellin, J., Adda, M., & Duke-Williams, E. (2010). *The Use of Simulation in Digital Forensics Teaching.* Portsmouth, UK: Higher Education Academy Subject Centre for Information and Computer Sciences.

Edens, P., Bell, S., Arthur, W., & Bennett Jr, W. (2003). *Effectiveness of Training in Organizations: A Meta-Analysis of Design and Evaluation Features.* Retrieved from rice.edu: http://www.owlnet.rice.edu/~antonvillado/courses/13c_psyc630002/Arthur,%20Bennett,%20Edens,%20&%20Bell%20(2003)%20JAP.pdf

Errington, E. P. (2010). *Preparing Graduates for the Professions Using Scenario-Based Learning.* Mt Gravatt, Australia: Post Pressed.

Faria, A. (2001). The Changing Nature of Business Simulation & Gaming Research: A Brief History. *Simulation Gaming*. Retrieved from http://sag.sagepub.com/cgi/content/abstract/32/1/97

Courtney Imbert, courtneyimbert@gmail.com

Hancock, P., Vincenzi, D., Wise, J., & Mouloua, M. (2008). *Human Factors in Simulation and Training.* Boca Raton, FL: CRC Press.

Jones, A., & Valli, C. (2009). *Building a Digital Forensic Laboratory: Establishing and Managing a Successful Facility.* Burlington, MA: Butterworth Heinemann and Syngress Publishing, Inc.

Libicki, M. C., Senty, D., & Pollack, J. (2014, June 19). *Hackers Wanted: An Examination of the CyberSecurity Labor Market.* Retrieved from Rand.org: http://www.rand.org/pubs/research_reports/RR430.html

Lorenzetti, J. P. (2010). *Promoting Academic Integrity in Online Education.* Madison, WI: Magna Publications.

Murphy, C. (2009). *Developing Process for Mobile Device Forensics.* Retrieved from sans.org: https://digital-forensics.sans.org/media/mobile-device-forensic-process-v3.pdf

Norton, G., Taylor, M., Stewart, T., Blackburn, G., Jinks, A., & Razdar, B. (2012). Designing, developing and implementing a software tool for scenario based learning. *Australasian Journal of Educational Technology*, 1083-1102.

Santoku Linux. (2015, October 28). *HOWTO get started with Android SDK in Santoku Linux*. Retrieved from Santoku Linux: https://santoku-linux.com/howto/development-tools/howto-get-started-with-android-sdk-in-santoku-linux/

Scott, S., & Koonts, T. (2013, October 27). *Technical implementation of a virtualized forensic distributed processing network using NGD NETLAB.* Retrieved from CPCC.edu: https://www.cpcc.edu/pd/funding/college-fellows-1/ARF2013.pdf

Shavers, B. (2008, November 27). *A Discussion of Virtual Machines Related to Forensics Analysis.* Retrieved from Forensics Focus: http://www.forensicfocus.com/downloads/virtual-machines-forensics-analysis.pdf

Splunk Enterprise. (2015, November 1). *Splunk documentation: Anonymizing data samples*. Retrieved from Splunk.com:

Courtney Imbert, courtneyimbert@gmail.com

http://docs.splunk.com/Documentation/Splunk/6.2.0/Troubleshooting/AnonymizedatasamplestosendtoSupport

Vanover, R. (2009, June 24). *Everyday Virtualization: Type 1 and Type 2 Hypervisors Explained.* Retrieved from Virtualization Review: https://virtualizationreview.com/blogs/everyday-virtualization/2009/06/type-1-and-type-2-hypervisors-explained.aspx

VMware. (2015, October 30). *Configuring Hard Disk Sotrage in a Virtual Machine*. Retrieved from VMWare: https://www.vmware.com/support/ws45/doc/disks_config_ws.html

VMware. (2015, November 1). *Defragmenting, shrinking, and cleaning up VMware Fusion virtual machine disks*. Retrieved from VMware Knowledge Base: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1001934

Zeltser, L. (2009). *Introduction to Malware Analysis.* Retrieved from www.zeltser.com: https://zeltser.com/media/docs/intro-to-malware-analysis.pdf

Courtney Imbert, courtneyimbert@gmail.com