Automated Testing of a Cyber Training Environment within an Agile Development Process

Michael Schwartz, Glenn A. Martin, Shehan Sirigampola, Steven Zielinski, Bruce Caulkins

University of Central Florida

Orlando, FL

mschwart@ist.ucf.edu, martin@ist.ucf.edu, ssirigam@ist.ucf.edu, szielins@ist.ucf.edu, bcaulkin@ist.ucf.edu

ABSTRACT

The Persistent Cyber Training Environment (PCTE) is a Department of Defense (DoD) initiative to provide the federal workforce with the ability to perform cyberspace operations training. PCTE offers experimentation, certification, assessment and development of cyber capabilities, techniques, and procedures for operations that cross networks and boundaries. The long-term goal of PCTE is to provide a cloud-based training platform that serves as a collective training network (e.g., Cyber Guard) to support the cyber workforce at the individual and organizational levels through hardware and software integration that organizes training, event planning, and after-action reviews into an interconnected training event. The University of Central Florida (UCF) Institute for Simulation and Training (IST) supports PCTE development by conducting testing the system as it evolves to changes in technology, tactics, techniques, and threats. The development of automated testing scripts to evaluate a software product being built with an agile process has many benefits and challenges, which are covered in this paper. Ranging from changes to the product across a sprint schedule to needs of tasking, development of automated tests across software updates and within various software installs have provided many lessons learned in such an approach. Cyber training systems make extensive use of virtual machines and container technologies as well that provides additional challenges.

ABOUT THE AUTHORS

Michael Schwartz is a Ph.D. student in UCF's School of Modeling, Simulation, and Training. His research focuses on multi-modal displays, workload, and performance. A second line of research centers on behavioral cybersecurity of wearable and Internet of Things (IoT) devices. He has an M.S. in Modeling and Simulation from UCF.

Glenn A. Martin is a Research Assistant Professor at UCF's Institute for Simulation and Training. He earned a Ph.D. in Modeling and Simulation in 2012 from the University of Central Florida. He pursues research in adaptive and intelligent training, game-based learning, multi-modal simulation, and interactive high performance computing.

Shehan Sirigampola is an Assistant in Simulation in the University of Central Florida Institute for Simulation and Training. His research interests focus on virtual environments and visualization. More specifically his work examines viability of virtual environments for training, distributed simulation, real-time rendering and parallel visualization. Shehan received his B.S. degree in Computer Science from the University of Central Florida in 2011.

Steven Zielinski is an Assistant in Simulation at UCF's Institute for Simulation and Training. He has worked on numerous projects, including augmented reality, intelligent training tools, and real-time virtual environments. He has a B.S. in Computer Science from the University of Central Florida.

Bruce Caulkins is a Research Assistant Professor at the Institute for Simulation and Training at UCF, focusing on cybersecurity-related research and instruction. He is also the Program Director for UCF's Modeling and Simulation of Behavioral Cybersecurity graduate certificate program. He is a retired Army Colonel with over 28 years of experience in tactical, operational, and strategic cyberspace operations. He earned his Ph.D. in Modeling and Simulation at UCF in 2005, focusing on anomaly detection within intrusion-detection systems.

Automated Testing of a Cyber Training Environment within an Agile Development Process

Michael Schwartz, Glenn A. Martin, Shehan Sirigampola, Steven Zielinski, Bruce Caulkins

University of Central Florida

Orlando, FL

mschwart@ist.ucf.edu, martin@ist.ucf.edu, ssirigam@ist.ucf.edu, szielins@ist.ucf.edu, bcaulkin@ist.ucf.edu

INTRODUCTION

Cybersecurity is a matter of economic, national, and global concern. The 21st century is marked by a trend toward increased reliance on cyber-physical systems. Cybersecurity is as important as physical security for military, medical, and utilities applications; however, cyber threats are more pervasive and persistent than physical threats. Worldwide, there is a need train millions of workers to meet the ever-evolving nature of cyber attacks (CSO, 2015). Anticipating future threats and security needs is difficult for cybersecurity professionals. The dynamic nature of cyber threats requires continuously evolving threat mitigation and prevention tools, tactics, and techniques. Current trends toward increased connectivity and automation in military and industrial systems offers new threat vectors for hostile entities to exploit. The evolution of cyber threats combined with the need to provide training at scale for novice employees and continuing education for experienced members of the cyber workforce presents a challenge to current security systems. Cybersecurity education programs must find innovative and effective methods of delivering educational content to respond effectively (Caulkins, 2009).

Automation is increasingly being used to maximize efficiencies in the delivery of cybersecurity education. Initiatives such as Advanced Distributed Learning's (ADL) Personalized Assistant for Learning (PAL) seek to deliver personalized education and training to the cyber workforce throughout their careers (Nicholson, Massey, O'Grady, & Ortiz, 2016). Performance characteristics and after-action reports can be automatically generated from exercises performed in simulation testbeds, thereby reducing the time needed for post-simulation analysis and report generation (Nevmerzhitskaya, Norvanto, & Virag, 2019). Automatic vulnerability detection can be performed by software programs, such as Security Administrator's Integrated Network Tool (SAINT) and Tiger (Caulkins, 2009). Automatic setup of cybersecurity testbeds is becoming more common; however, automated generation of test scenarios is often performed manually by security professionals (Beuran, Chinen, Tan, & Shinoda, 2016; Beuran, Tang, Pham, Chinen, Tan, & Shinoda, 2018). Automated script testing of cybersecurity testbeds in development is another area of need. Using automatic scripts can offload tedious and repetitive tasking from cyber professionals, thereby speeding up the development of cybersecurity scenarios in a manner that scales across the workforce. All of the above capabilities are leveraged in robust modeling and simulation-based cybersecurity training programs to continuously test networks and systems and train the Cyber Mission Force (CMF). Proactive cyber defense practices will aid cybersecurity specialists in strengthening networks and devices against known and novel attacks (e.g., polymorphic attacks).

Automated Attacks

Cyber criminals are using automated modes of attack. The general public became aware of automated cyber threats in September of 2016 when the Mirai botnet infected 65,000 Internet of Things (IoT) devices within 24 hours. Compromised IoT devices are leveraged by Mirai to continuously scan the Internet for other devices to infect with malware. Infected devices, which are now a botnet, monitor a command and control server that indicates the intended target. The botnet then disables websites and servers in a distributed denial of service (DDoS) attack. The Mirai source code is published as open source on Github and has spawned many variants. The automated attacks have continued and proliferated as a result. The asymmetry between cyber attacks and defenses is a defining principle of cybersecurity. Cyber defenders must be trained to think like adversaries and use similar tools to prepare for attacks and protect critical infrastructure. Automating the creation, development, evaluation, and analysis of cybersecurity training scenarios are ways of keeping pace with automated attacks and the U.S. military's need to innovate the process of educating cybersecurity professionals.

Automated Testing Scripts and Agile Software Development

Cross-functional teams engage in agile software development by completing production sprints, periods of swift design, development, testing, delivery, and iteration. Stakeholders, including the U.S. military, are seeking to automate parts of the agile development process to meet rapidly evolving cyber threats. Automated software testing scripts have been developed to aid in the rapid and continuous validation of cybersecurity testbeds. An example automated testing process engages in five steps: 1) obtaining relevant files from the version control system, 2) building tests, 3) loading and running tests, 4) retrieving and storing the results, and 5) taking result-dependent next steps. Implementation of these steps will vary according to the system hardware and development tools. Automating the testing process does not eliminate the need for manual testing; however, it does free up team resources for other important tasks. Integration testing is performed continuously to ensure code performs as intended. The automated testing process helps teams meet delivery targets in a fast-paced market using fewer resources and with fewer bugs released. This reduces the need to fix version defects at a later date and an increased cost. Developers and testers can use automated testing to rapidly identify and correct defects and improve quality. Project managers can use automated testing to verify code is working correctly and ready for delivery.

The Persistent Cyber Training Environment (PCTE) is a Department of Defense (DoD) platform for cyber mission force (CMF) training, education, and mission rehearsals for cyberspace activities. PCTE capabilities include supporting individuals and teams in setup, planning, training and review in one interconnected platform. Connectivity is a primary advantage of the PCTE as different organizations have different technological capabilities. For example, the cyber systems used by the various United States military branches and its allies are unlikely to be identical across installations. The PCTE allows for organizations with diverse capabilities and aligned security goals to interoperate in effective cybersecurity training. In this paper we describe how automated testing scripts are being used in an agile development environment to accelerate the improvement and continued evolution of the PCTE.

In cybersecurity, a vulnerability is a weakness or susceptibility to attack that can be exploited by a hostile entity to intrude into an information system and gain unauthorized access to information (U.S. Department of Defense, 2001). Cyber threats are actors (e.g., hostile individuals or nation-states) who attempt to leverage information communication technologies (ICT) to breach cyber defenses (U.S. Department of Homeland Security, 2008). Cyber threats are using automation to initiate billions of attacks each year, such as attempting to gain access to internet of things (IoT) devices by using bots that deploy commonly used passwords.

PCTE Updates

PCTE is under active development using an agile software development process. The project performs regular releases (about every six months). However, within the agile process the project conducts required sprints typically four weeks in length. At the end of each sprint, each vendor drops releases for their components that includes new features development and known bugs fixed. As the system is updated during each sprint, a series of tests are performed to ensure existing functionality. The result is a system that is under frequent and rapid changes.

As with many projects, PCTE performs regression testing and new feature testing at each software drop (e.g. each sprint). However, PCTE encompasses multiple Remote Compute and Storage (RCS) servers, each running the PCTE system in its entirety. RCSes will be installed at various locations around the United States of America for use by different parts of the Cyber Mission Force. In addition, each RCS can also include multiple zones, or sandboxes (again, each a full installation of the PCTE system). This necessitates an additional testing step.

As each zone within each RCS is updated, a series of smoke tests is also performed. These tests check basic functionality of the PCTE system and tests the most important features. By their nature, smoke tests are not exhaustive, but do check the most critical functions. However, smoke tests are extended and enhanced into a full end-to-end test suite as well.

Finally, PCTE also goes under scalability testing. As suggested by the name, this focuses on ensuring that the system can perform adequately when used by a large number of users. The tests scale up the number of simulated users performing tasks on a single installation while measurements are taken of system load, RAM usage, and latency.

While these various forms of testing are performed manually by humans, we are implementing them using automated scripts as well. This will allow a reduction in cost and effort, and also allow additional runs of the tests (for example, regression tests could be run nightly). Currently, our automated testing efforts focus on the user experience. Therefore, we have created automated tests that login as users and perform various tasks. We have largely pursued smoke tests and scalability tests to this point.

To achieve automated testing, we have used the Katalon Studio tool. This allows the development of test scripts that can be run with user start-up actions or by an automated scheduler. The scripts access the PCTE site, click on HTML elements as a human user would, and a report on functionality and timing produced. To support scalability testing with many users, Selenium Grid is added (Katalon Studio is compatible with it) to leverage multiple computers to log in multiple users in a parallel fashion.

Lessons Learned

While automated testing has added much to the testing process, using such a process within PCTE has had some challenges. First, the agile development process and its rapid additions and changes to the PCTE system creates a situation where testing scripts can quickly become out-of-date. HTML elements can move or change (such as a button to a slider) either in function or in location. This requires a similar labor need and sprint schedule to maintain the automated scripts.

In addition, as discussed PCTE uses many sandboxes across many RCSes. Each sandbox typically can have a slightly different version (especially during development). So automated scripts either have to handle slightly varying versions or they must focus on only one version (likely, the latest). We have chosen the latter approach to this point.

Furthermore, scalability testing requires sufficient performance needs (number of computers, RAM). In order to test a large number of simulated users, a relatively larger set of test client computers are needed. For example, to simulate 10,000 users all using the system in parallel, a number of test client machines are used to drive them. In addition, running such tests in parallel can require a large amount of RAM for Katalon Studio and Selenium Grid Hub. As we scale the number of users, we continue to adjust these needs.

Lastly, to address all the issues and lessons learned, we have also leveraged standard software development techniques. The automated test scripts have common functions that they can leverage (e.g. login a user). This abstraction/encapsulation has helped with automated testing script development as it does with any software package.

Challenges

PCTE leverages virtual machines and containers heavily. Similarly, automated testing does as well. The testing scripts are run within a container and their reports stored for display by a separate web server container. Security concerns can be a challenge for automated testing in this context. As a Government Information System, PCTE follows necessary standards. While any issues can and are addressed, sometimes such solutions can take some dissection and consideration. For example, files must be labeled appropriately. In addition, access to the PCTE system through Virtual Private Network (VPN) can be challenge for automated scripts running nightly. To this point, we have run the scripts directly within one sandbox zone in order to address this latter issue.

Conclusion

Continuous attacks require constant defense. PCTE affords the CMF with the ability to have nonstop training and evaluation take place across devices, installations, borders, and services. Cybersecurity training is no longer like playing capture the flag and instead represents the distributed, uninterrupted nature of cyber attacks. PCTE templates, such as Cyber Forge, are downloaded and used by Army, Navy, and Air Force teams around the world to simultaneously take part in joint exercises. Exercises take place at the scope and scale necessary to train the CMF for current and future threats. PCTE comprises more than 150 virtual machines and continues to grow to meet the size required to keep the CMF up to date. Cyber Forge examined how the system performed across time zones. Cyber Anvil, a version B prototype, was the first major user PCTE event. Members of the Army, Navy, Coast Guard, Marine Corps, Air Force Reserve, and Air National Guard participated in a 72-hour exercise from locations distributed around the world. While Cyber Anvil was a start, the exercise was essentially rapid prototyping. Cyber Anvil took place at seven distributed sites across five time zones and involved almost 100 participants. Trainees took part in elastic skills building, a cyber team hunt scenario, and individual forensic skills training. Additional value was created from participant criticisms and PCTE managers are continuing to seek feedback in future iterations. This rapid DevOps process allows developers to quickly respond to input from the operational the operational community. More testing is needed to validate and verify individual and collective training for future exercises. Future tests will continue to examine the system performance, software utilization, and load balancing capabilities. The ability of PCTE to support including more teams mid-exercise is another area of interest. Hardware and software integrity across the platform is being analyzed. The capacity for PCTE to support key stakeholders, such as training managers, also needs to be examined.

Cyber Valhalla was another opportunity to test and iterate PCTE in preparation for a January 2020 release (PCTE 1.0) with additional updates planned. The PCTE program plans to spend almost $500 million on the program through 2025. The U.S. Army is conducting a cyber innovation challenge (CIC) series, which will provide funding to incorporate new technologies into the PCTE. A technical management dashboard and white cell exercise control are already slated for incorporation into the platform. Continued development is planned for Joint Qualification Requirements (JQRs), the checklists DoD personnel use to know when to deploy a cyber weapon system, such as a defensive kit deployed when responding to a cyber threat. The future of PCTE is difficult to predict due to the reactive nature of cybersecurity in response to cyber threats; however, several trends are planned to continue in PCTE development. An increased focus on connected, but distributed operations is likely to persist as U.S. Cyber Command looks to integrate cyber responses across military branches. Automation will be expanded to assist with script testing as well as creation and evaluation of scenarios. Continued expansion of management tools, such as system health dashboards, is likely to continue as the military seeks to increase the ability of individuals to perform multiple operations at once (e.g., one person controlling multiple unmanned aerial systems). The military is also realizing the value in seeking and incorporating stakeholder feedback early and continuously throughout the system development process. Agile development will continue to enhance PCTE development and help secure critical infrastructure for the United States and her allies.

References

Beuran, R., Chinen, K. I., Tan, Y., & Shinoda, Y. (2016). Towards effective cybersecurity education and training.

Beuran, R., Tang, D., Pham, C., Chinen, K. I., Tan, Y., & Shinoda, Y. (2018). Integrated framework for hands-on cybersecurity training: CyTrONE. *Computers & Security*, *78*, 43-59.

Caulkins, B. D. (2009). Proactive self-defense in cyberspace. ARMY WAR COLL CARLISLE BARRACKS PA.

CSO. (2015, July 28). *Cybersecurity Job Market to Suffer Severe Workforce Shortage*. Retrieved from CSO Online: http://www.csoonline.com/article/2953258/it-careers/cybersecurity-job-market-figures-2015-to-2019-indicate-severe-workforce-shortage.html

Nevmerzhitskaya, J., Norvanto, E., & Virag, C. (2019). High impact cybersecurity capacity building. In *The International Scientific Conference eLearning and Software for Education* (Vol. 2, pp. 306-312). " Carol I" National Defence University.

Nicholson, D., Massey, L., O'Grady, R., & Ortiz, E. (2016). Tailored Cybersecurity training in LVC environments. In *MODSIM World Conference, Virginia Beach, VA*.

U.S. Department of Defense, DOD Dictionary of Military and Associated Terms, Joint Publication 1-02 (Washington, DC: U.S. Department of Defense, October 17, 2001), 587.

U.S. Department of Homeland Security—U.S. Computer Emergency Readiness Team, “Cyber Threat Source Descriptions,”http://www.us-cert.gov/control_systems/csthreats.html (accessed December 6, 2008).