Teaching Information Security with Workflow Technology – A Case Study Approach

Wu He
Department of Information Technology & Decision Sciences
Old Dominion University
Norfolk, VA 23529, USA
whe@odu.edu

Ashish Kshirsagar
Alexander Nwala
Yaohang Li
Department of Computer Science
Old Dominion University
Norfolk, VA 23529, USA
akshirsa@cs.odu.edu, anwala@cs.odu.edu, yaohang@cs.odu.edu

ABSTRACT

In recent years, there has been a significant increase in the demand from professionals in different areas for improving the curricula regarding information security. The use of authentic case studies in teaching information security offers the potential to effectively engage students in active learning. In this paper, the authors introduce the approach of using workflow technology to compose case studies to enhance information security education. This approach allows students from different disciplines to collaborate in a distributed computing environment in order to learn important information security principles. Two case studies simulating real-life scenarios, including one for an online banking system and one for an online grading system, are recreated within a laboratory setting using workflow technology and are then presented in information security classes. Our educational practice shows that the benefits of using workflow technology in information security education have been well received by students.

Keywords: Workflow technology, Case-based learning, Case-based instruction, Case study, Security

1. INTRODUCTION

Nowadays, information security is becoming more and more important to organizations across a variety of industries. In information security education, it has been observed that students learn best with hands-on real-life examples (Sharma and Sefchek, 2007). Teaching using case studies to simulate real-life scenarios has been considered an effective method that actively engages students. Researchers (He et al., 2013) point out that using a case study-based approach in information security courses can provide several key advantages, such as the ability to focus on the practical aspects of information security in the real world, the ability to ensure a high level of student involvement, and the ability to teach security concepts with a minimum requirement for equipment.

In recent years, many hands-on projects, including simulating virus and spyware mechanisms (Katz, 2006), breaking ciphers (Schweitzer and Baird, 2006), attacking/defending a system (Chen et al., 2011), detecting intrusion (Roschke et al., 2010), and analyzing penetration (Antunes and Vieira, 2012), have been developed and have been widely used in the information security education curriculum. However, most of these hands-on projects are relatively simple and can be carried out in isolated or virtualized lab environments, which usually do not fully reflect the complexity of real-life situations. Moreover, most of these projects require students to have strong technical skills in system/network administration and often involve heavy computer programming. These requirements might make the study of information system security difficult for students with less background in computer science.

In information security education practice, there is a great need for students to study more complicated cases that are more similar to real-life scenarios in order to better understand important security issues and techniques. Ideally,

each case study should simulate a complex real-life scenario. These case studies should enable students to visualize concepts at a high level in order to facilitate analysis and discussion. Moreover, in order to engage students and to sustain students' interest in learning information security concepts, difficult technical details should be temporarily hidden, particularly until students with less computer science background can become familiar with these concepts.

In this paper, we describe a new approach of using workflow technology to enhance information security education (van der Aalst and van Hee, 2004) by simulating complex real-life scenarios within a laboratory setting. The use of workflow technology provides interactive graphical interfaces to build sophisticated information security cases without the need for low-level programming or command-line interactions and allows for collaboration among students with different educational backgrounds. Moreover, workflow technology enables seamless integration of distributed and local services to support the composition of complex case studies. Two case studies using the Kepler scientific workflow system (Ilkay et al., 2004) are presented in this paper to show how workflows for real-life scenarios can be created and enacted. The first case study simulates the scenario of an attack on a bank account and is based on a real security incident described in the Daily Record magazine (Mann, 2012). The second case study models the situation of a coordinated attack that compromises an online course management system. The workflows for both case studies were developed by students in Computer Science and were then used to support teaching in Information Technology (IT) security courses. Feedback from students regarding the use of workflow technology in teaching information security principles and techniques is also discussed and analyzed.

2. WORKFLOW TECHNOLOGY AND ITS APPLICATIONS IN EDUCATION

Workflow is originally an administrative concept used in business operation management; it describes a business process that delivers services from one participant agent to another. A workflow is described as (Allen, 2001):

> *The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules*.

The most attractive feature of a workflow is "business automation," which enforces data validation and verification in business operations, overcomes constraints in time and space, maintains consistency, and largely eliminates possible human errors. While the idea of workflow has been widely used in many different types of businesses, the concept has extended beyond conventional business process management and is now applied more broadly in scientific computing (Deelman et al., 2009), sensor networks (Li et al., 2008), engineering (Vucina et al., 2009), image processing (Radu and Gorgan, 2007), e-commerce (Li et al., 2004), and many other areas.

The basic element in a workflow is a task, which is defined by three parameters: input description, transformation, and output description. The input description provides the information required to complete the task; the transformation is composed of a set of functions to be carried out in the task; and the output is the resulting information produced in the task that can be provided as input to the downstream tasks. Typically, a workflow consists of a set of serial or parallel tasks that perform the operations of accessing services or executing specific functions. Closely related tasks can also be organized as a sub-workflow that can be reused in the composition of other workflows. In general, the structure of a workflow can be represented as a DAG (Direct Acyclic Graph), in which arcs connect tasks. The relationship between tasks can be sequential, parallel, or selective (Yu and Buyya, 2005a).

A number of tools have been developed to facilitate the composition of workflows. In addition to the Kepler workflow used in this paper, others include Pegasus (Deelman et al., 2003), Taverna (Oinn et al., 2004), Gridflow (Cao et al., 2003), ScyFlow (McCann et al., 2006), and GridNexus (Brown et al., 2005). Typically, a workflow system provides a GUI layout that allows users to drag and drop in order to create tasks and to connect services or logic components. Given the input and output parameters, a correctly composed workflow can be compiled (Li et al., 2007) to a format described by a workflow description language or other XML-based languages. The workflow can then be scheduled and executed on various service providers or resources under the orchestration of a workflow management system (Yu and Buyya, 2005b). The effectiveness of workflow techniques has been verified and includes applications in biology, chemistry, business, and mathematics. A key advantage of workflow technology is that it extricates users from many complicated and low-level system configurations, programming, and data manipulations and thereby allows them to focus on composing the applications, such as the grid or the cloud computing infrastructures, used in today's distributed computing environment.

In addition to workflow's popularity in scientific computing and e-business applications, educators have begun to investigate the feasibility of using workflow technology to support educational practices. Van der Veen et al. (2003) have found that using workflow to transfer business applications into the educational domain has provided added value to students. Santoro et al. (2003) integrated the workflow concept to support collaborative project-based learning and achieved good learning outcomes. Hiekata et al. (2007) used a semantic web-based workflow framework to support design engineering education, which helped shorten the students' learning duration. Gorgan et al. (2009) used workflow to develop a Grid-oriented e-learning environment for supporting training and experimental practice in earth observation. Wilkinson and Ferner (2008) used the GridNexus workflow editor to teach Grid Computing classes across universities in North Carolina. Silva et al. (2011) used VisTrails, a scientific workflow and provenance management system, to support teaching in scientific visualization. Waddell et al. (2010) employed a Kepler workflow to design several simple case studies, including a distributed program code review, remote code analysis, and a web service fuzzer (a software testing tool that automatically provides invalid and unexpected random data to the inputs of a web service), and then used these case studies to teach a course in secure software engineering.

We recently used workflow technology to build sophisticated case study scenarios in order to simulate real-life examples of breaches in information security. These case studies were then used in teaching several information security courses. Each case study included a list of learning objectives, a workflow-based case scenario, and a learning task. According to Bloom's revised taxonomy (Anderson and Krathwohl, 2001), there are six levels of cognitive skills and capabilities. These are briefly described in Table 1.

Cognitive Level	Description
Remembering	Retrieving relevant knowledge from long-term memory
Understanding	Constructing meaning from instructional messages, including oral, written, and graphic communication
Applying	Carrying out or use a procedure in a given situation
Analyzing	Breaking down informational materials into components to understand the organizational structure
Evaluating	Making judgments based on criteria and standards
Creating	Putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure

Table 1: Bloom's revised taxonomy (Anderson & Krathwohl, 2001)

Each case study is designed to target certain cognitive levels of Bloom's revised taxonomy. Due to the complexity and diversity of security scenarios, it can be challenging to allow students to test each security scenario and/or to conduct security attacks and defenses in the existing classroom environment. Therefore, based on the characteristics of several different security attacks, the authors of this paper mapped case studies at different cognitive levels. The first case study targets the highest cognitive level, "Creating." The second case study targets a lower level, "Understanding." The next section provides detailed descriptions of these two case studies.

3. CASE STUDIES USING WORKFLOW TECHNOLOGY

Case Study I: A Complicated Attack Pattern for an Illegal Banking Account Transfer

In 2012, a real-life security scam with a relatively complicated attack pattern that involved an online bank account wire transfer was reported (Mann, 2012). The ultimate goal of the hacker was to transfer a large amount of money from a compromised customer's banking account to an overseas account. To accomplish this kind of attack, a hacker breaks into a bank customer's banking account to initiate the money transfer. However, recently, many banks have implemented an anti-fraud policy that notifies customers, mostly via automated phone calls or messages, if unusual activities, such as money transfers exceeding a certain amount, are detected in their accounts. Warned by the notification messages, the banking customer who is the victim would have the chance to stop the money transfer during a grace period. To be successful, then, the hacker must prevent the customer from receiving the anti-fraud notification calls or messages. In order to achieve this, the hacker must carry out a sequence of denial-of-service attacks by continuously calling the customer's phone. Bothered by numerous strange phone calls, the annoyed customer will likely turn off his/her phone and will thereby miss the anti-fraud notifications from the bank. As a consequence, if the above steps in the scam and the attack procedure are carried out successfully, the hacker will achieve his goal of transferring money out of the customer's banking account.

This case serves as an ideal example to use in class to demonstrate a complicated real-life attack pattern. Using this real-life banking case as the blueprint, the first workflow scenario was designed by simulating the above banking case while embedding within it several basic and important information security techniques that are taught in regular Information Security classes. First of all, the authors implemented a module that simulated an SQL injection attack, in order to simulate the hacker's process of stealing the user's credentials and personal information from the back-end database by taking advantage of security holes in the web server. The sub-workflow in the SQL injection module serves the educational purpose of showing how a security hole in the database of an information system can lead to vulnerability. In the denial-of-service attack module, the authors developed a sub-workflow to invoke a phone-calling web service to implement the making of consecutive phone calls. During the in-class demo, a student was asked to volunteer his cell phone to demonstrate the effect of this denial-of-service attack. A transaction validator module was also implemented, in order to reflect the changes to the banking accounts in this case study. The learning objectives of this case study include:

- To understand SQL injection attacks in computer programs;
- To describe strategies for avoiding SQL injection attacks;
- To discuss banking anti-fraud policies;
- To analyze computer programs and identify SQL injection vulnerabilities;
- To apply coding techniques to eliminate SQL injection vulnerabilities in computer programs; and
- To create SQL injection-free computer programs.

Figure 1 shows the high-level Kepler workflow that implements the banking case study and offers relevant screen shots (including SQL injection information, banking account status changes, and user notifications during the workflow execution). The high-level workflow provides a big picture of the overall attack pattern in this case study. The execution of the workflow simulates the banking attack case study step-by-step, and shows how it provided students with an opportunity to experience a simulated real-life security attack within a lab setting. At the same time, students were inspired to think about the related security techniques and policies in order to prevent such attacks. Students who were interested in technical details could explore the sub-workflow in each module to learn more about the detailed implementation

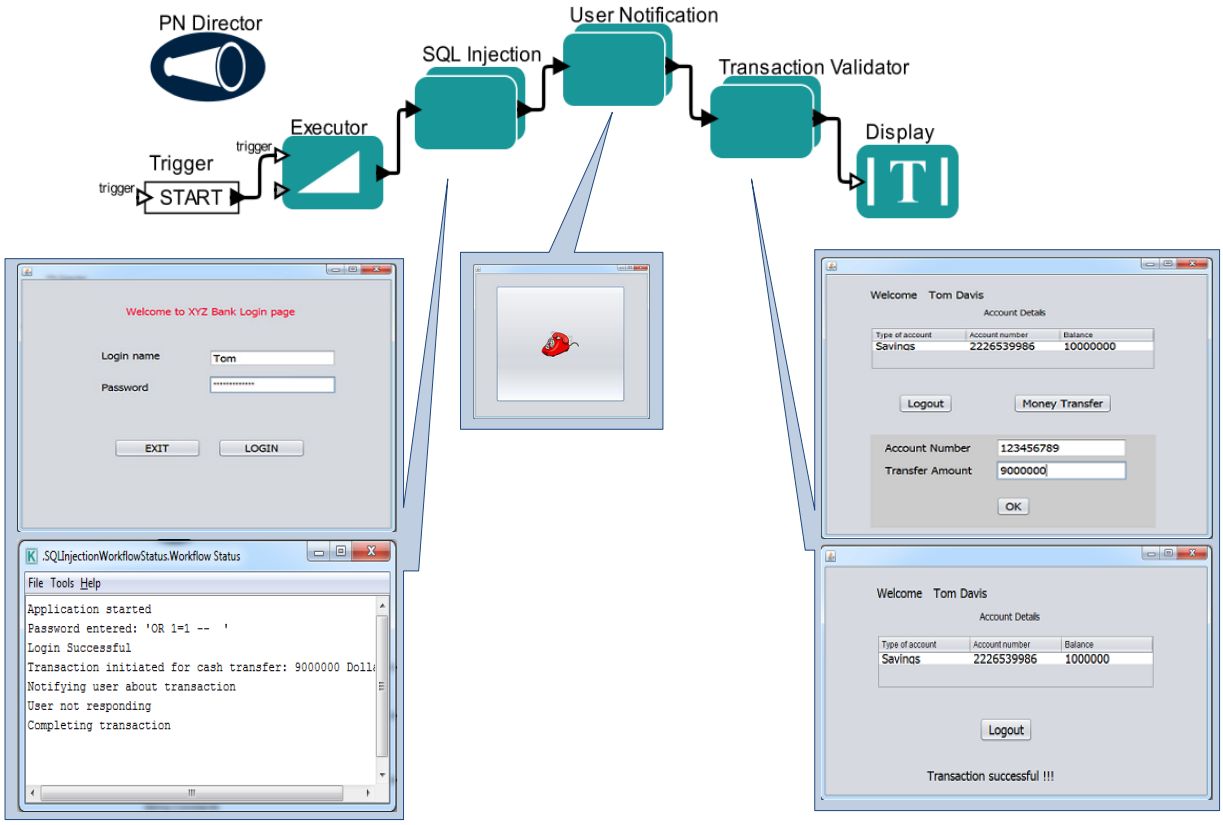


Figure 1: Kepler Workflow for Case Study I: A Complicated Attack Pattern for an Illegal Banking Account Transfer

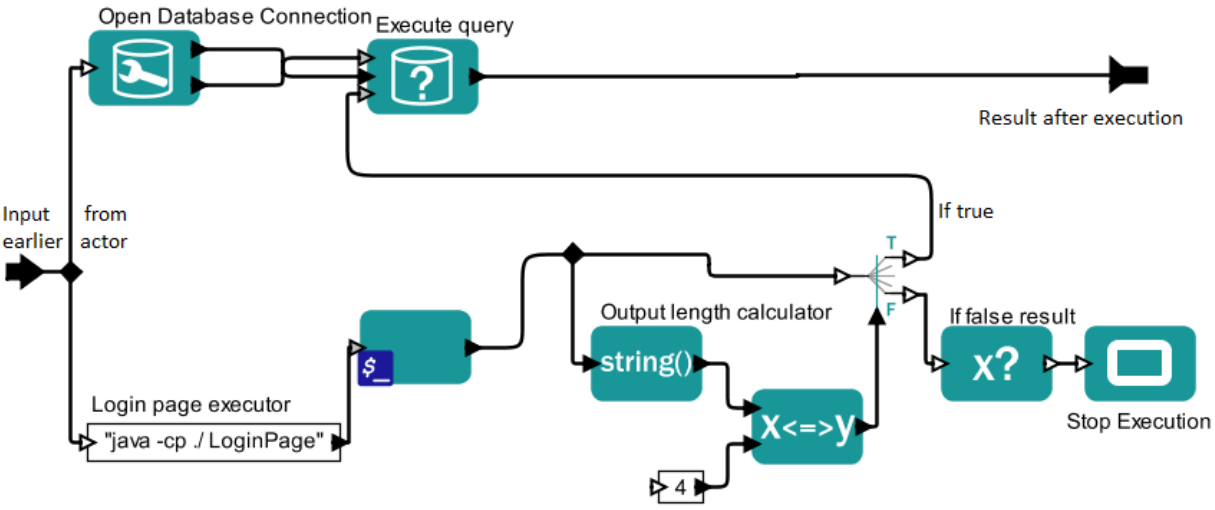


Figure 2: Implementation Detail of the SQL Injection Sub-Workflow

mechanism. An example of the detailed implementation of the SQL injection sub-workflow is shown in Figure 2.

In order to help students achieve the learning objectives, a learning task was associated with this case study. Students were asked to evaluate the web scripting programs in the case study workflow to search for potential SQL injection vulnerabilities and to provide solutions to address these security holes. The follow-up programming assignment and project involved students creating web-scripting programs that were free of SQL injection vulnerabilities. The instructor tested and graded the students' programs to determine their effectiveness against SQL injection attacks.

Case Study II: Coordinated Attacks on Online Course Management Systems

Online course management systems such as Blackboard and Moodle are widely used in higher education. Instructors use such systems to teach their online or on-campus courses, using them to store course materials, to conduct online discussions, and to manage student grades. As the popularity of online course management systems continues to grow, there has been an increased interest in attacking online course management systems or other e-learning applications (Schultz, 2013). The design of the second case study was inspired by various recent real-life attempts to break into online course management systems in order to change grades (Carr, 2013). This case study models the situation of a coordinated attack in order to compromise a test bed (which is a simulated Moodle-based online course management system used for testing purposes). The instructor discussed the ethics guidelines and issues before presenting this case study to students. Since several studies discussed ethical hacking and suggested that a university information security curriculum must include both the "defender" and "attacker" perspectives in order to meet the demand for trained security professionals to have both attack and defense skills (Bratus et al., 2010; Curbelo and Cruz, 2013; Logan and Clarkson, 2005; Pashel, 2006), both perspectives were used by students. To reduce possible ethics concerns, the actual programming codes were hidden from the students.

One way to attack online course management systems is to use the denial-of-service (DoS) attack technique. To launch a DoS attack, the attacker typically sends a very large number of connection requests to flood the target system. When the target system receives more requests for services than it can handle, it often stops working or even crashes (Whitman and Mattord, 2011). Once the target system is down, in co-ordination, the hackers can set up a fake system with exactly the same URL and user interface on the Internet and can use it for phishing. Phishing is an attempt to gain personal or financial information from an individual, usually by posing as a legitimate entity. When the course instructor then tries to log in to the fake course management system, his or her account information will be captured by the fake system. Later on, the hackers can use the instructor's account information to visit the legitimate course management system and can carry out malicious activities, such as changing grades or stealing exam/quiz questions, at will.

This second case study implements a workflow-based scenario to simulate the effect of a DoS attack and a phishing attack on an online course management system established for testing. The learning objectives of this case study are:

- To understand DoS attacks and phishing on computer systems;
- To understand the impact of DoS attacks on network and systems; and
- To identify defense mechanisms to protect against DoS attacks and phishing.

Figure 3 shows the designed overall workflow of this case study. The DoS attack sub-workflow initiates the server attack by calling a java application that attempts to establish a large number of connections to the server hosting the online course management system. When the server's resources are exhausted, the server is then unable to accept new connections. Figure 4 illustrates the detailed implementation of the DoS attack sub-workflow. Following the DoS attack, a phishing module is used to implement the impersonation of the authentic online course management server. This fake Trojan server has exactly the same user interface appearance as the original system and serves the purpose of phishing the login account information from unsuspecting instructors. This stolen information can then be used to steal test questions or to modify student grades.

In order to support the learning objectives associated with this case study, further in-class discussion about possible ways to mediate the DoS attacks can be included, in order to help students study DoS attacks and the techniques that can detect DoS attacks and can mediate the impact of such attacks. The example shown in Figure 5 demonstrates the "SiteKey" technique used to prevent phishing. By comparing the executions of the Case Study II workflow on the Moodle servers both with and without SiteKey protection, students can gain a better understanding of DoS attacks and can learn techniques that can prevent phishing.

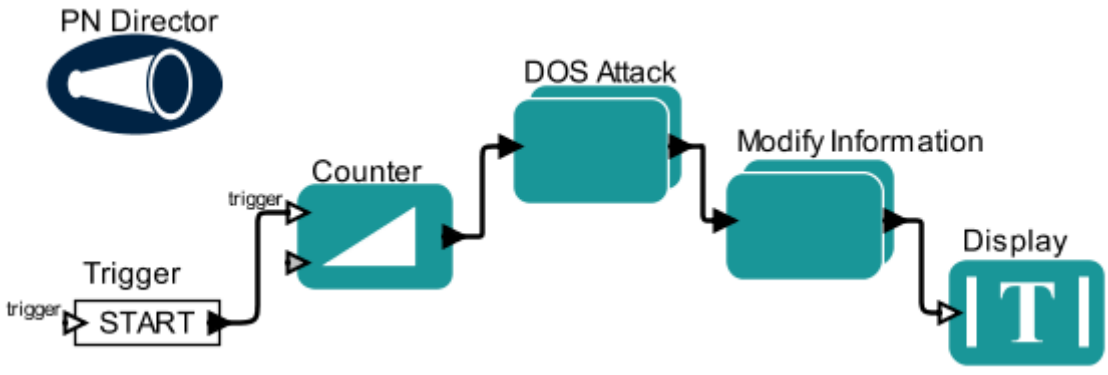


Figure 3: The Kepler Workflow for Case Study II: Coordinate Attacks on an Online Course Management System

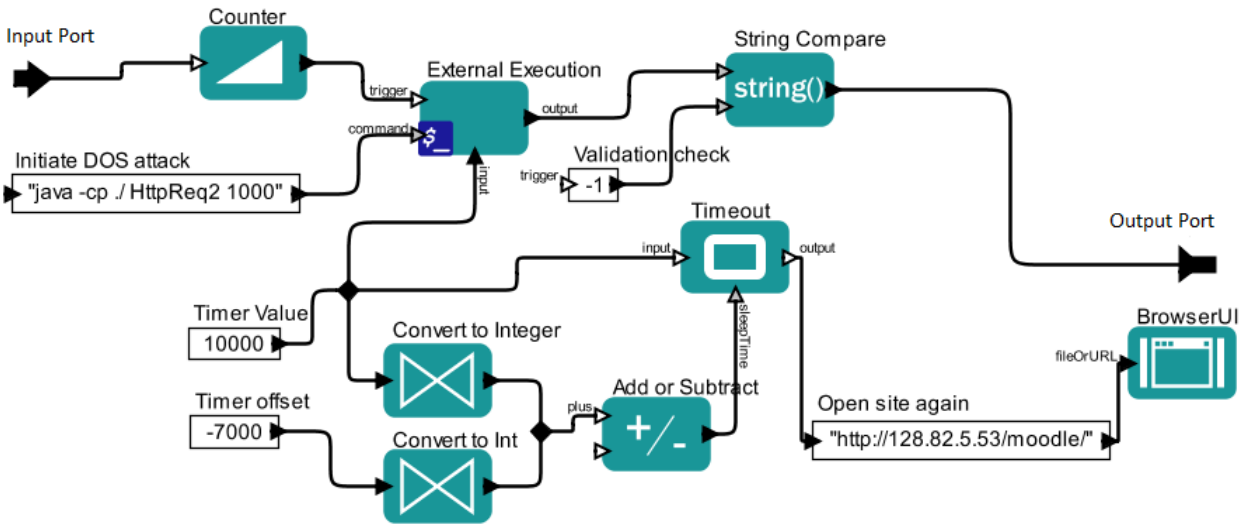
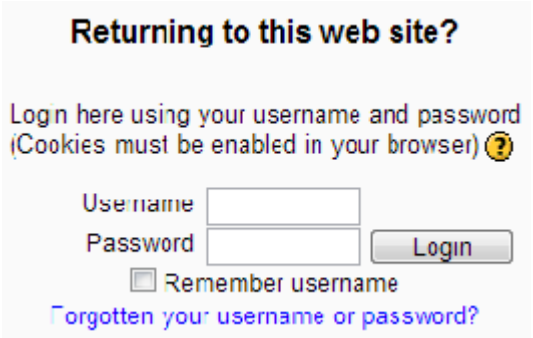
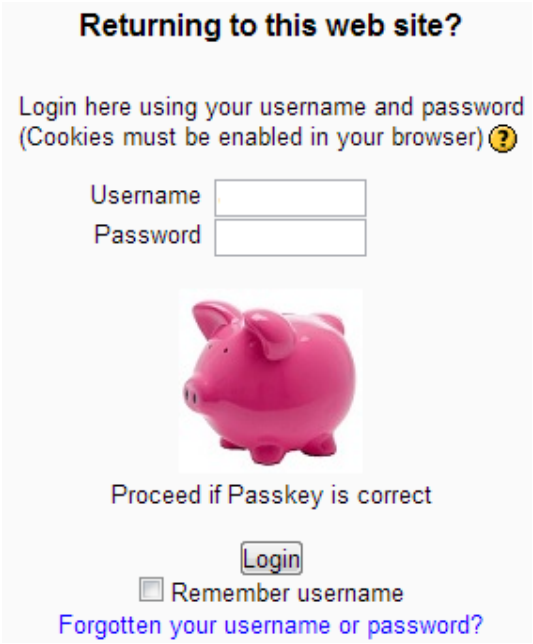


Figure 4: DoS Attack Sub-Workflow with Detailed Implementation of the DoS Attack Mechanism on a Moodle Server



(a): Moodle Server without SiteKey Protection



(b): Moodle Server with SiteKey Protection

Figure 5: Using SiteKey Protection in the Moodle Server to Prevent Phishing

4. EVALUATION OF THE WORKFLOW-BASED SCENARIOS

The two developed workflow-based scenarios were demonstrated to students in four IT courses with small class sizes (class 1 has 16 students; class 2 has 9 students; class 3 has 7 students; and class 4 has 18 students). After a demonstration of the two case studies, the students were invited to complete a survey regarding using workflow technology for teaching information security. The survey included three quantitative and seven qualitative questions, and the college's institutional review board (IRB) approved the survey. 42 students (26 undergraduates and 16 graduate students) volunteered to complete the survey. These students reported that they were new to using a workflow technology like Kepler. The results of the quantitative questions from the survey are summarized as below.

The students were asked to rate the degree of their agreement with three learning-related statements. Table 2 shows that the majority of the students either agreed or strongly agreed that using the case studies implemented by workflow technology was helpful in deepening their understanding of the information security concepts and technology.

Students also made some qualitative comments as they answered the seven qualitative questions. The majority of the students reported that they had some basic knowledge about information security but lacked an understanding of the technical details involved in security attacks such as SQL injection and denial-of-service attacks. Some students explained why they thought that scenario-based workflows were useful for learning information security principles and reported on their favorite part of the scenario-based workflows. Some noted that they liked the fact that workflow breaks down the scenarios on a step-by-step basis, making the hacking process easier to follow. Several reported that they enjoyed seeing the simulated attacking process behind the SQL injection, denial-of-service, and phishing attacks visually, and they appreciated the workflow implementation in helping them better understand the underlying concepts and techniques. One student even pointed out that the workflow-based scenarios provided an opportunity to conduct practical training without the need of multiple computers or servers. Below are some quotes from the participating students:

- *"I understand the attacks well after seeing the demo."*
- *"It gave me a better understanding of SQL injections."*
- *"Scenario-based workflows look useful in dealing with user requirements."*
- *"Very good. It visually depicts the flow of security attacks and helps improve understanding."*
- *"In the real world, you have to be able to explain security issues to non-technical managers. I think the scenario-based workflows are very important."*
- *"My favorite part is to see codes interpreted graphically."*

The participating students also provided suggestions for improving the workflow-based scenarios. They suggested that more details be added to the workflows in order to make the scenarios as similar as possible to real-life situations. They also suggested making the GUI of the workflows more user-friendly and engaging. A few students noticed that there were many actors in the workflows and wondered how much time it took to build a complex workflow. Some students expressed interest in learning how to make such workflows because they believed that the workflow-based scenarios were useful. Some suggestions from the students who participated in the survey are listed here:

- *"Make the program mimic more complex workflows based on actual sites."*
- *"Make it a little more realistic. The online banking scenario needs to deal with emails, too."*
- *"I am curious to see the translation from graphical workflow to actual code if possible."*

After the in-class demonstration of the two case studies, the instructor asked students to complete the associated learning tasks. For case study I, students in the Web programming course were required to check the programs that they had previously developed and to fix the SQL injection vulnerabilities, if any existed. Students, in the subsequent assignments and for the final project, were also required to create programs that were free of SQL injection vulnerabilities. As they graded students' assignments and tested their programs, the authors found that the vast majority of students understood the SQL injection vulnerabilities well and successfully wrote codes to prevent SQL injection. Thus, it was clear that most of the students achieved the learning objectives of Case Study I.

Statement	Strongly agree	Agree	Neither Agree Nor Disagree	Disagree	Strongly disagree
Workflow technology is very useful for learning information security concepts.	12 (28.6%)	21 (50.0%)	7 (16.7%)	0 (0.0%)	2 (4.8%)
I am interested in learning more about using workflow technology for information security.	9 (21.4%)	20 (47.6%)	11 (26.2%)	1 (2.4%)	1 (2.4%)
I enjoyed learning information security concepts using workflow technology.	8 (19.0%)	25 (59.5%)	8 (19.0%)	0 (0.0%)	1 (2.4%)

Table 2: The survey results of the learning-related statements

For Case Study II, students in the information security course were required to further research DoS attacks and were asked to discuss in class what approaches they could use 1) to detect DoS attacks and 2) to mediate the impact of such attacks. Based on their discussion, the instructor found that the students showed a solid understanding of DoS attacks, of the consequences, and of possible mediation strategies. The authors did not conduct further testing to assess students' understanding about DoS attacks, since their main interest was to ascertain how students perceived the workflow-based information security scenarios.

5. CONCLUSIONS

Case studies have often been recognized as important tools to illustrate conceptual or complex materials. Teaching information security skills effectively is difficult without ready access to adequate case studies. Often, a real-life information security situation is complicated and involves numerous steps. It can be challenging for instructors to describe such complex security situations orally in class. It can also be hard for students to understand complex security techniques and concepts without being offered visual examples. In this paper, we developed two workflow-based case studies using Kepler software to simulate real-life scenarios in information security. The two workflow-based scenarios were then introduced to students in both undergraduate- and graduate-level courses. The evaluation results show that most of the students were positive regarding the effectiveness of using workflow to teach security techniques and concepts. A limitation with this evaluation is that the sample size is small. The authors hope to conduct a large-scale evaluation with more students in the future, in order to further improve and enhance the case studies implemented by workflow.

Currently, there is a lack of ready-made information available about security-related case studies, which makes the application of case study methods in information security education challenging (He et al., 2013). The authors of this paper plan to share their developed workflow-based case studies on the Internet. Regarding future work, the authors plan to design guidelines to help interested information security instructors develop workflow-based case studies by using workflow development/management software packages. In addition, the authors are designing more workflow-based security case studies and will share them with the information security educational community once they are ready.

6. ACKNOWLEDGEMENTS

This research was supported by a Faculty Innovation Grant from the Center of Learning and Teaching at Old Dominion University, Norfolk, Virginia, USA.

7. REFERENCES

Allen, R. (2001). Workflow: An Introduction. In L. Fischer (Ed.), *Workflow Handbook* (pp. 15-38). Lighthouse Point, Florida: Future Strategies Inc.

Anderson, L. W. & Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York, NY: Longman.

Antunes, N. & Vieira, M. (2012). Defending Against Web Application Vulnerabilities. *Computer*, 45(2), 66-72.

Bratus, S., Shubina, A., & Locasto, M. E. (2010). Teaching the Principles of the Hacker Curriculum to Undergraduates. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, (122-126), Milwaukee, WI.

Brown, J. L., Ferner, C. S., Hudson, T. C., Stapleton, A. E., Vetter, R. J., Carland, T., Martin, A., Martin, J., Rawls, A., Shipman, W. J., & Wood, M. (2005). GridNexus: A Grid Services Scientific Workflow System. *International Journal of Computer and Information Science*, 6(2), 72-82.

Cao, J., Jarvis, S. A., Saini, S., & Nudd, G. R. (2003). GridFlow: Workflow Management for Grid Computing. *Proceedings of 3rd International Symposium on Cluster Computing and the Grid (CCGrid)*, (198-205), Tokyo, Japan.

Carr, D. F. (2013). The Cybersecurity Challenge on College Campuses Lies as Much with the Students as with Malicious Outsiders. *Dark Reading*. Retrieved March 10, 2014, from http://www.darkreading.com/security/hacking-higher-education/d/d-id/1109684?

Chen, G., Dong, Z. Y., Hill, D. J., & Xue, Y. S. (2011). Exploring Reliable Strategies for Defending Power Systems against Targeted Attacks. *IEEE Transactions on Power Systems*, 26(3), 1000-1009.

Curbelo, A. M. & Cruz, A. (2013). Faculty Attitudes toward Teaching Ethical Hacking to Computer and Information Systems Undergraduates Students. *Proceedings of the Eleventh LACCEI Latin American and Caribbean Conference for Engineering and Technology* (LACCEI'2013), paper 86, Cancun, Mexico. Retrieved March 10, 2014, from http://www.laccei.org/LACCEI2013-Cancun/RefereedPapers/RP086.pdf

Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., & Vahi, K. (2003). Mapping Abstract Complex Workflows onto Grid Environments. *Journal of Grid Computing*. 1(1), 25-39.

Deelman, E., Gannon, D., Shields, M., & Taylor, I. (2009). Workflows and E-Science: An Overview of Workflow System Features and Capabilities. *Future Generation Computer Systems*. 25(5), 528-540.

Gorgan, D., Stefanut, T., & Bacu, V. (2009). Grid Based Training Environment for Earth Observation. *Proceedings of 4th International Conference in Advances in Grid and Pervasive Computing (GPC),* (98-109), Geneva, Switzerland.

He, W., Yuan, X. H., & Yang, L. (2013). Supporting Case-Based Learning in Information Security with Web-Based Technology. *Journal of Information Systems Education*, 24(1), 31-40.

Hiekata, K., Yamato, H., Rojanakamolsan, P., & Oishi, W. (2007). A Framework for Design Engineering Education with Workflow-Based E-Learning System. *Journal of Software*, 2(4), 88-95.

Ilkay, A., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., & Mock, S. (2004). Kepler: An Extensible System for Design and Execution of Scientific Workflows. *Proceedings of the 16th Conference on Scientific and Statistical Database Management (SSDBM)*, (423-424), Santorini Island, Greece.

Katz, F. H. (2006). Campus-Wide Spyware and Virus Removal as a Method of Teaching Information Security. *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development*, (1-4), Kennesaw, GA.

Li, Y., Esterline, A. C., Baber, C., Fuller, K., Burns, M., Hansen, T., LeFebvre, T., Schultz, M., Govett, M., Hamer, P., & Mysore, A. (2008). A Sensor Information Framework for Integrating and Orchestrating Distributed Sensor Services. *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, (857-862), Las Vegas, NV.

Li, Y., Cai, Q., & Li, Y. (2004). Toward a Dynamic E-Commerce Automation with XML and Workflow Techniques on the Grid. *Proceedings of IEEE SoutheastCon*, (310-315), Greensboro, NC.

Li, Y., Chen, D., & Yuan, X. (2007). Trustworthy Remote Compiling Service for Grid-Based Scientific Applications. *Journal of Supercomputing*, 41(2), 119-131.

Logan, P. Y. & Clarkson, A. (2005). Teaching Students to Hack: Curriculum Issues in Information Security. *ACM SIGCSE Bulletin*, 37(1), 157-161.

Mann, S. D. (2012). Lessons Learned from a Phone Scam. Retrieved March 10, 2014, from http://thedailyrecord.com/generationjd/2012/04/23/lessons-learned-from-a-phone-scam/

McCann, K. M., Yarrow, M., DeVivo, A., & Mehrotra, P. (2006). ScyFlow: An Environment for the Visual Specification and Execution of Scientific Workflows. *Concurrency and Computation: Practice & Experience*. 18(10), 1155-1167.

Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., & Li, P. (2004). Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows. *Bioinformatics*. 20(17), 3045-3054.

Pashel, B. A. (2006). Teaching Students to Hack: Ethical Implications in Teaching Students to Hack at the University Level. *Proceedings of the 3rd annual conference on Information security curriculum development*, (197-200), Kennesaw, GA.

Radu, A. & Gorgan, D. V. (2007). Diagrammatic Description of Satellite Image Processing Workflow. *Proceedings of International Symposium on Symbolic and Numeric Algorithm for Scientific Computing*, (341-348), Timisoara, Romania.

Roschke, S., Willems, C., & Meinel, C. (2010). A Security Laboratory for CTF Scenarios and Teaching IDS. *Proceedings of 2nd International Conference on Education Technology and Computer*, (433-437), Shanghai, China.

Santoro, F. M., Borges, M. R. S., & Santos, N. (2003). Using Workflow Concepts to Support Collaborative Project-Based Learning. *Proceedings of World Conference in Educational Multimedia, Hypermedia, and Telecommunications*, (140-147), Honolulu, Hawaii.

Schultz, C. (2013). Information Security Trends and Issues in the Moodle E-Learning Platform: An Ethnographic Content Analysis. *Journal of Information Systems Education*. 23(4), 359-372.

Schweitzer, D. & Baird, L. (2006). The Design and Use of Interactive Visualization Applets for Teaching Ciphers. *Proceedings of 7th IEEE Workshop on Information Assurance*, (69-75), West Point, NY.

Sharma, S. K. & Sefchek, J. (2007). Teaching Information Systems Security Courses: A Hands-On Approach. *Computers & Security*, 26(4), 290-299.

Silva, C. T., Anderson, E., Santos, E., & Freire, J. (2011). Using Vistrails and Provenance for Teaching Scientific Visualization. *Computer Graphics Forum,* 30(1), 75-84.

van der Aalst, W. & van Hee, K. M. (2004). *Workflow Management: Models, Methods, and Systems*. Cambridge, MA: MIT Press.

van der Veen, C. B. & Jones, V. (2003). Network Applications for Group-Based Learning: Is More Better? *Interactive Learning Environments*. 11(2), 127-146.

Vucina, D., Lozina, Z., & Pehnec, I. (2009). Reverse Engineering with Shape Optimization using Workflow-Based Computation and Distributed Computing. *Proceedings of the World Congress on Engineering*, (789-794), London, U.K.

Waddell, I., Jones, N., Steed, C., Yuan, X., & Li, Y. (2010). Using the Workflow Technology in Secure Software Engineering Education. *Proceedings of 14th Colloquium for Information Systems Security Education*, (76-82), Baltimore, MD.

Whitman, M. E. & Mattord, H. J. (2011). *Principles of Information Security* (4th edition). Boston, MA: Cengage Learning.

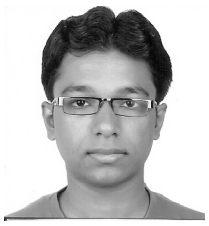
Wilkinson, B. & Ferner, C. (2008). Towards a Top-Down Approach to Teaching an Undergraduate Grid Computing Course. *Proceedings of ACM SIGCSE*, 40(1), 126-130.

Yu, J. & Buyya, R. (2005a). A Taxonomy of Scientific Workflow Ssystems for Grid Computing. *ACM SIGMOD Record*, 34(3), 44-49.

Yu, J. & Buyya, R. (2005b). A Taxonomy of Workflow Management Systems for Grid Computing. *Journal of Grid Computing*. 3(3-4), 171-200.

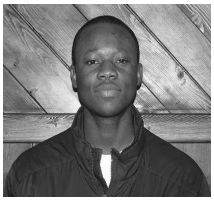
AUTHOR BIOGRAPHIES

Wu He received the B.S. degree in Computer Science from DongHua University, China, in 1998, and the Ph.D. degree in Information Science from the University of Missouri, USA, in 2006. He is an Assistant Professor of Information Technology at Old Dominion University. His research interests include Data Mining, Information Security, Social Media, Knowledge Management and Computing Education.



Ashish Kshirsagar is a Graduate student at Old Dominion University pursuing degree in Master's in the field of Computer Science. He received B.E. degree from Shivaji University, India in 2007 in Computer Science and Engineering. He has 5 years of industry experience in the Software Engineering field from 2007 to 2012.

Alexander Nwala received his B.Sc. in Computer Science at



Elizabeth City State University, Elizabeth City, North Carolina in 2011, his M.Sc. at Old Dominion University, Norfolk, Virginia in 2014. He is currently pursuing his Ph.D. at Old Dominion University under the supervision of Dr. Michael Nelson. He has developed computer vision algorithms currently used in industry. His research interests include Artificial Intelligence, Text Mining and Web Sciences.

Yaohang Li is an Associate Professor in the Department of

Computer Science at Old Dominion University. His research interests are in Computational Biology and Scientific Computing. He received the Ph.D. and M.S. degrees in Computer Science from the Florida State University in 2003 and 2000, respectively. After graduation, he worked at Oak Ridge National Laboratory as a research associate for a short period of time. Before joining ODU, he was an associate professor in the Computer Science Department at North Carolina A&T State University.