Back to Pen and Paper: Recovering Assessment Questions from Computer-Based Examination Applications

Full Paper

Shalabh Saini
Department of Information Systems and Cyber Security, The University of Texas at San Antonio, USA
shalabh16932@gmail.com

George Grispos
Lero–The Irish Software Research Centre
University of Limerick, Ireland
george.grispos@lero.ie

Charles Zhechao Liu
Department of Information Systems and Cyber Security, The University of Texas at San Antonio, USA
charles.liu@utsa.edu

Kim-Kwang Raymond Choo
Department of Information Systems and Cyber Security, The University of Texas at San Antonio, USA
raymond.choo@fulbrightmail.org

Abstract

Computer-based examination platforms are increasingly utilized to deliver examinations. The advantages of implementing such a solution over a traditional pen-and-paper approach include support for remote candidates, reduced financial costs, shorter grading time and reduced carbon footprints. While the benefits are attractive, the nature of computer-based examinations makes them more susceptible to cheating. One conjecture previously discussed in the literature is that candidates could recover questions from hard drives. However, minimal research has been conducted to affirm these assumptions. Hence, this research investigates the extent to which questions can be recovered from a computer-based examination and makes recommendations to counter such problems based on the Routine Activity Theory. The study contributes to the discipline by empirically demonstrating that questions can be recovered from three computer-based examination applications using freely available computer forensic tools, and highlighting the need for additional security measure to enhance the creditability of the computer-based examination applications.

Keywords


Introduction

The integration of Information and Communications Technology (ICT) into the educational domain has introduced a variety of benefits, opportunities and challenges. Educational practices that were once undertaken on paper have now been replaced by digital solutions. One particular example that has benefited from this digital revolution is the mechanism used to administer and deliver tests and examinations (Rowe 2004; Wahid et al. 2015). This revolution has seen traditional ‘paper-pen’ tests and examinations being replaced by computer-based examination platforms, including the use of mobile devices (Nikou and Economides 2016; Rowe 2004; Wahid et al. 2015). There are several benefits for implementing computer-based examinations including support for remote candidates, reduced financial costs, shorter grading time and reduced carbon footprints as a result of less paper being used. The use of computer-based examinations has also assisted universities to extend their degree programs to students around the world through the implementation of Massive Open Online Courses (Dreier et al. 2014).
Although the benefits of computer-based examinations are attractive, several researchers have argued that fundamental problems exist with the use of these examinations in real-world contexts (Dreier et al. 2014; King et al. 2009; Rowe 2004; Ullah et al. 2012). These concerns focus on the security vulnerability of the software used to administer the examinations (Balasundaram 2011; Ullah et al. 2012), as well as the increased possibility of cheating in these examinations (Balasundaram 2011; Dreier et al. 2014; King et al. 2009; Rowe 2004). Security concerns regarding computer-based examinations were recently exacerbated in October 2016 when the Ontario Education Quality and Accountability Office (EQAO) confirmed that a Distributed Denial of Service attack was responsible for the cancellation of the online Ontario Secondary School Literacy Test (Thomson 2016). In a later press release, the EQAO announced that the next literacy test in March 2017, would be administered in paper format only (The Education Quality and Accountability Office 2016).

The remote nature of many computer-based examinations means that they are also more susceptible to various kinds of fraud. As a result Mitra and Gofman (2016) argue that greater integrity is needed for assessments that are delivered using computer-based examination platforms. Dreier, et al. (2014) add that such examination platforms should “provide a comparable guarantee of security and not only against students that cheat, the main concern of exam authorities, but also against other frauds including those perpetrated by the authorities themselves”. To this effect, researchers have identified various approaches that candidates could employ to exploit the vulnerabilities in assessments deployed using these computer-based platforms (Bella et al. 2016). One of these approaches is that candidates could, potentially, use computer forensic tools to recover examination questions and answers from hard disk drives and computer networks (Rowe 2004), and exploit these resources either during or after the examination. However, while Rowe (2004) has discussed this technique in the past, minimal empirical research has actually been conducted to examine and affirm these assumptions. This situation prompts research into the following hypothesis:

*Freely available computer forensic tools can be used to recover examination questions from a computer-based examination application.*

To test this hypothesis, we designed an experiment to examine if freely available computer forensic tools could be used to recover assessment questions from a computer-based examination application on a Windows-based laptop. The contribution of our study is two-fold. First, it validates the theoretical conjecture that assessment questions can be recovered from three computer-based examination applications using freely available computer forensic tools. Second, it highlights the need for additional security mechanisms to prevent candidates from cheating by recovering questions from computer-based examination applications.

The next section discusses related work concerning the security and fraud in computer-based examinations. We then present the experiment design used in this paper, our findings, and the impact of our findings. We also explain how the Routine Activity Theory (Cohen and Felson 1979) can be used as the underlying theoretical lens to design countermeasures. The last section concludes this paper and presents future work.

**Related Work**

Computer-based examinations attempt to make use of various platforms including the Internet, computer applications, web browsers and mobile devices to deliver assessments and examinations to candidates (Nikou and Economides 2016; Rowe 2004; Wahid et al. 2015). Examinations delivered using these platforms are usually undertaken away from a university, college or training provider. As a result, there is the potential that examination candidates could compromise a particular platform and cheat during an examination. Hence, various research efforts have discussed the possibility to cheat using computer-based examinations and attempted to provide various security solutions to counter this problem.

Rowe (2004) argues that there are several reasons why it is easier to cheat in computer-based examinations, including the instructor not knowing who is actually taking the exam, the assessment being undertaken at any time of the day, which can increase temptation, and students being more technology savvy than their instructors and therefore knowing how to exploit computer programs to cheat. Rowe (2004) goes on to discuss three of the most serious problems involving cheating in computer-based examinations, including getting assessment answers in advance, unfair retaking or grade changing, and
Unauthorized help during assessments. Wahid, et al. (2015) surveyed the literature and attempted to classify the methods that can be used to obtain answers illegally during a computer-based examination. These techniques include browsing the internet for answers, using internet messenger applications for obtaining answers, students sitting together in close proximity, retrieving questions and answers from local or external storage devices (such as hard drives), and reading a book or tutorial whilst taking the examination.

Separately, Kennedy, et al. (2000), Rogers (2006), and Stuber-McEwen, et al. (2009) have conducted studies to evaluate cheating in online and computer environments as compared to their traditional counterparts. Kennedy, et al (2000) explored student and faculty views concerning cheating and distance learning. The results of their study indicated that both faculty and students believe it is easier to cheat in distance learning classes because someone else could be responsible for the completion of assessments or that students could purchase/download papers off the Internet. Rogers (2006) reported that many of the faculty in her survey, who were utilizing online testing, had in fact indicated that they had experienced students who had either cheated or been suspected of cheating during an online exam. Further complicating the problem, 81.8% of the faculty respondents who have used online examinations or quizzes stated that they administer these assessments in an unproctored environment (Rogers 2006). In contrast to Kennedy et al. (2000) and Rogers’ (2006) findings, Stuber-McEwen, et al. (2009) reported that their ‘Student Academic Dishonesty Survey’ results indicated that students that were enrolled in online classes were less likely to cheat than those enrolled in traditional courses. As a result, the authors concluded that the amount of academic misconduct among online students might not be as prevalent as believed in the literature.

In order to counter some of the cheating concerns with computer-based examination platforms, several researchers have proposed a variety of security solutions. Agulla, et al. (2008) developed a web-based application that offers biometric authentication based on face-recognition. The application was developed so that it can be integrated into existing examination systems. Eplion and Keefe (2005) undertook a study that aimed to detect and prevent cheating on their own online examinations. After identifying and confirming that cheating was taking place within their examinations, Eplion and Keefe proceeded to implement a number of measures which included limiting access to the examination by requiring a unique userID and password, setting a time limitation to conclude the actual examination, and analyzing the student’s answers based on IP address location (Eplion and Keefe 2005). Castella-Roca, et al. (2006) developed and implemented a secure electronic examination management system that uses cryptographic schemes to ensure that a high level of security is available for all stages of an examination. However, a limitation of their work is that any examination will still need to take place in a supervised environment. Further work was needed to evaluate how this system could be implemented in a less restricted setting (Castella-Roca et al. 2006). Alternatively, several authors have developed frameworks to help guide the secure development and delivery of computer-based examinations (Dreier et al. 2014; Pan et al. 2004; Ullah et al. 2012; Wahid et al. 2015). Many of these approaches propose the use of authentication to ensure that the correct individual has attempted the assessment, as well as preventing candidates from communicating through a computer network. While previous research has established that it could be easier to cheat on computer-based examinations and attempted to provide security solutions, minimal research has examined the extent to which a computer-based examination application can be compromised using freely available computer forensics tools for the purpose of extracting examination questions.

**Experiment Design**

An experiment was developed to test the research question of this study. The experiment was broken into six stages, namely: 1) preparing the environment and installing the examination software, forensic tools and various applications; 2) creating mock examinations using the examination software; 3) creating a forensic image of the environment; 4) converting the forensic image of the environment into a virtual machine; and 5) executing the virtual machines created and 6) extracting the examination questions, until no new questions are recovered. The overall experiment process is illustrated in Figure 1.
In order to prepare the test environment, a laptop computer was formatted and Windows 10 was installed onto the laptop. After completing the Windows installation, various freely available forensic tools and applications were installed onto the laptop – see Table 1. It must be noted that all the tools were executed on the laptop with elevated privileges.

<table>
<thead>
<tr>
<th>Tool Name</th>
<th>Version</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forensic Toolkit (FTK) Imager</td>
<td>3.4.2</td>
<td>Used to create a raw forensic image of laptop hard drive</td>
</tr>
<tr>
<td>Live View</td>
<td>0.7b</td>
<td>Used to create a VMware virtual machine configuration files from a raw forensic disk image</td>
</tr>
<tr>
<td>VMware vCenter Converter</td>
<td>6.0</td>
<td>Used to create VMware virtual machines using the forensic images and configuration files generated using Live View</td>
</tr>
<tr>
<td>VMware Player</td>
<td>6.0.3</td>
<td>Execute the VMware virtual machines created using Live View and VMware vCenter Converter</td>
</tr>
</tbody>
</table>

Table 1. Forensic Tools and Applications

After installing the tools and applications, the laptop was connected to the Internet and used to download two computer-based examination applications, Test Generator Lab version 6.0, produced by EPractize Labs, and Exam version 16.9, produced by Exam-Software. These computer-based examination applications were selected for two reasons. First, these two applications are available to trial without financial cost. Second, neither of the selected computer-based examination applications places any restrictions on the number of questions or setup of the examinations in their trial versions. Although a number of alternative computer-based examination applications may also fulfill these criteria, these two systems were most readily available to the authors and are considered highly representative of other equivalent products in the market at the time of the study.

After installing and activating the two computer-based examination applications, we then proceeded to create two mock examinations using the applications. Using the Test Generator Lab software, we created a pool of twenty (20) examination questions, which were largely based on a freely available Test of English as a Foreign Language (TOFEL) practice test. We setup this mock examination with ten (10) questions

1 Available from [www.examboat.com](http://www.examboat.com)
and imposed a ten-minute time limit on taking the actual examination. With regard to the test application offered by Exam software, we then proceeded to create a mock examination consisting of a pool of fifty (50) questions. These questions covered a range of topics including mathematics, English and general knowledge, with participants only being presented ten (10) questions in a ten-minute time limit. After creating the question pools, an executable program was created for each mock examination that could be distributed to a candidate to undertake the examination.

Prior to executing the candidate’s executable program, the test laptop was powered down and an E01 forensic image of the laptop’s hard disk drive was created using FTK Imager, using the default options for this application. The resulting E01 forensic image was then transferred to a separate Windows 10 desktop to create a bootable virtual machine. The process to create a virtual machine involved two phases. In the first phase of the conversion process, FTK Imager was used to emulate the E01 image as a physical drive. Live View was then executed and the physical drive was loaded into the application in order to generate VMware virtual machine configuration files. The default initialization parameters were used during this phase of the conversion process. The second phase of the conversion process involved opening the configuration files and E01 forensic image with VMware vCenter Converter Standalone to create a VMware bootable virtual machine. We allocated 20 GB of hard disk space and 2.5 GB RAM for the virtual machine. With regard to the virtual hard disk drive, we opted to use the ‘SCSI LSI Logic SAS’ disc controller option to prevent a ‘blue screen of death’ error from occurring (EMC Corporation 2010). In addition, virtual networking options were removed from the bootable virtual machine to restrict the examination software accessing the Internet once the virtual machine was powered up.

The last step in the process was to power-up each virtual machine, execute the candidate’s executable program and attempt the particular examination. This was done in order to extract the examination questions. Video recording software was used to capture the contents displayed on the screen, i.e. the question that would be presented to the candidates. This was repeated for each question in the examination and at the conclusion of the examination the virtual machine was powered down. The procedure of creating a copy of the virtual machine and recovering examination questions using the video recording software was then repeated. If a particular question was not presented to the authors in a previous recording of the virtual machine, the question was then captured using the recording software. The process of creating virtual machines and extracting questions was repeated until no new examination questions were recovered from a virtual machine. Screenshots from the various video recordings were then used to produce a document containing the recovered questions.

Mock Examination Findings

A summary of the results from the analysis of the two computer-based examination applications is presented in Tables 2 and 3.

<table>
<thead>
<tr>
<th>Attempt Number</th>
<th>Observations</th>
<th>Number of Unique Questions Recovered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All ten questions were attempted and recorded – 0 duplicates</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>All ten questions were attempted and recorded – 0 duplicates</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>All ten questions were attempted and recorded – 10 duplicates</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>20</strong></td>
</tr>
</tbody>
</table>

Table 2. Test Generator Lab Version 6.0 Results

Our findings demonstrated that it is possible to record and recover all of the questions that were included in the pool of questions created for the two mock examinations. More specifically, in the analysis of the Test Generator Lab, all twenty questions from the mock examination pool of questions were recovered.

Note that installing the examination generation program on the same laptop that runs the candidate’s executable program does not affect the recovery of test questions. Generating the candidate’s executable program on a different computer will yield the same results as produced in this study.
after copying and analyzing two instances of the virtual machines. In the third copy created, no further new questions were recovered and our analysis showed that we had recovered all twenty (20) questions from the question pool in this software.

<table>
<thead>
<tr>
<th>Attempt Number</th>
<th>Observations</th>
<th>Number of Unique Questions Recovered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All ten questions were attempted and recorded – 0 duplicates</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>All ten questions were attempted and recorded – 0 duplicates</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>All ten questions were attempted and recorded – 4 duplicates</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>All ten questions were attempted and recorded – 3 duplicates</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>All ten questions were attempted and recorded – 2 duplicates</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>All ten questions were attempted and recorded – 1 duplicate</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>All ten questions were attempted and recorded – 10 duplicates</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>50</strong></td>
</tr>
</tbody>
</table>

**Table 3. Exam Version 16.9 Results**

With regard to the Exam software, six (6) copies of the virtual machine running this computer-based examination application were needed to recover the fifty (50) questions in the mock examination pool. An interesting observation from the analysis of this computer-based examination application is that the number of unique questions recovered from the analysis of the copied virtual machine increased with each additional virtual machine that was analyzed. This finding supports our idea that in order to extract the maximum number of examination questions from computer-based examination application, virtual machines should be copied and analyzed until no further unique questions are recovered. In summary, all fifty (50) questions from the mock question pool for this examination were recovered in six (6) iterations of the analysis, with the seventh copy in our analysis revealing no further questions.

**Professional Certification Examination – Case Study**

In order to further evaluate our approach and its use to recover questions from computer-based examination applications, we conducted a case study of a professional certification that uses such a system to deliver examinations to candidates. In order to protect the identity and susceptibility of the professional certification, we have chosen to present our results and analysis anonymously. However, to aid the discussion of our case study, we will refer to the professional certification as ProCert.

ProCert is a professional certification that is awarded by an organization based in the United States. According to the issuing organization, it is the largest institution in its field with over 75,000 members. In order to undertake the ProCert examination, after undertaking the necessary training or studying, candidates are required to obtain access to a computer as the organization does not make use of testing centers. Examination candidates are therefore offered the opportunity to take the ProCert examination at any day or time. After a candidate has been approved to take the examination and paid the required fee, they are provided with a web link to download a computer-based application that is used to deliver the ProCert examination. In addition, candidates are emailed a unique activation key, which is used to activate the examination when the candidate is ready to begin the assessment. According to the professional organization’s website, the ProCert examination contains four sections with a total of 500 questions, which are generated from a master question pool. A ProCert candidate is required to complete all four sections of the examination within one month of activating the actual examination. Candidates are required to complete a section after starting it, but do not need to complete all four sections at once. The organization claims that a number of security controls are in place to ensure that the integrity of the examination is upheld, but it does not provide any details on these controls.

After downloading the ProCert examination software and receiving the activation key from the organization, we then proceeded to install the software and activate the examination. We then replicated
our approach, as discussed in the Experiment Design, to create a forensic image and a virtual machine. We took a pragmatic decision to breakdown our analysis of the ProCert examination system into the four sections. The proposal was that a virtual machine would be created, powered up and then used to record and recover questions from the first section of the ProCert examination. A new copy of a virtual machine was then created and the first section of the examination was re-attempted, documenting all the questions that did not appear in the previous analysis and discarding any duplicate questions. This process was repeated until no new questions were recovered from the first section of the examination. After the analysis of the first section was completed, we then proceeded to the second section. The process of creating virtual machines and recovering examination questions until no new questions were recovered was then repeated for the second, third and fourth sections of the ProCert examination. The synthesis of our analysis of the ProCert computer-based examination is presented in Table 4.

Table 4 shows that a total of 1001 questions were recovered from the analysis of the virtual machines. In summary, 247 questions were recovered from section one, 251 questions from section two, 250 questions from section three and 253 questions from section four. In the analysis of all four sections, seven copies of the virtual machine were required to recover new questions. In all four sections, the analysis of the eighth virtual machine did not reveal any new questions for the examination. An interesting finding from our analysis was the identification of a unique number for each question recovered from the examination. From our analysis of the recovered questions and these unique numbers, it appears that each question in the master question pool is given one of these unique identifiers. The recovery of this question identifier allowed us to easily remove any duplicate questions, when they were recovered and recorded. While this number is likely to have been implemented for practicality and performance benefits, our experiment has shown it can also be used to help a malicious individual identify and remove duplicate questions quickly using our approach.

<table>
<thead>
<tr>
<th>Section Number</th>
<th>Number of Attempts</th>
<th>Number of Unique Questions Recovered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>247</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>251</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>250</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>253</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>8</strong></td>
<td><strong>1001</strong></td>
</tr>
</tbody>
</table>

**Table 4. ProCert Case Study Results**

**Countermeasures: A Routine Activity Theory Approach**

The results of our experiment have highlighted the need for taking additional steps to minimize the impact of recovering questions from master question pools included with computer-based examination applications. The fact that all examination questions can be recovered in our three experiments in a relatively streamlined process raises serious concerns about the vulnerabilities of these computer-based examination applications and the creditability of their test results. In particular, once the exam questions have been recovered, malicious individuals could distribute these questions further and profit from their illegal access to these resources, which extends the detrimental impact from individual examination to a greater scope. Hence, serious countermeasures need to be implemented based on the vulnerabilities identified in our study.

Based on our experiment findings, we propose five recommendations that could be implemented to avoid candidates recovering examination questions using our approach. We frame our recommendations using the Routine Activity Approach (Cohen and Felson 1979), which proposes that crime occurs when a suitable target is in the presence of a motivated offender and is without a capable guardian. Therefore, when designing countermeasures, we need to target one or more of these areas, namely: increasing the

---

3 This last attempt is technically a verification that no further new questions can be recovered.
effort required to offend (reduce opportunity); increasing the risk of getting caught (enhance guardianship); and reducing the rewards of offending (reduce motivation) – see Table 5.

Our first recommendation to counter the problem identified in our research is that examination providers remotely monitor and supervise candidates who take computer-based examinations, to avoid any violations. For example, examination providers could enforce the compulsory installation of an offline monitoring application, which would execute alongside the computer-based examination. The purpose of this monitoring application is to act as a virtual invigilator, recording and monitoring any candidate activity involving the computer-based examination (i.e. enhancing guardianship). For such an approach to work, the monitoring application would need to be executed as soon as the computer-based examination application is installed and when the examination is activated. This would then record any attempts by candidates to duplicate the examination questions.

Our second recommendation is that examination providers implement a one-time activation code within their computer-based examination applications. In other words, when a candidate first attempts to start a question or section of questions, the computer-based examination application would attempt to ‘call home’ and activate its one-time code (i.e. enhance guardianship). More importantly, enforcing a one-time activation code per examination means that a candidate can only use the activation once and prevents a candidate using a virtual machine and executing multiple ‘calls home’ (i.e. reduce motivation).

Our third recommendation is that examination providers impose a time limit on the computer-based examination. This countermeasure could be used together with our recommendation that computer-based examination applications ‘call home’. This solution would involve the examination provider setting a time limit when the examination attempts to ‘call home’ and then refusing to grade an examination after the limit has expired. Alternatively, if the time limit expires the candidate could also be forced to request a new activation key from the examination provider (i.e. reduce opportunity), as well as new set of questions (i.e. reduce motivation). Enforcing such time limit controls would restrict candidates from creating virtual machines and then executing the machine to recover all the questions.

Our fourth recommendation is that computer-based examination providers should prevent the execution of their examination application in a virtual machine environment. For example, Lau and Svajcer (2010) as well as Raffetseder, et al. (2007) have discussed how several families of malware now include virtual machine detection and execute differently to avoid security researchers performing any form of malware analysis. This technology could also be embedded within computer-based examination applications to avoid candidates executing the examination in virtual environments and extracting questions using the approach described in this paper (i.e. enhance guardianship, and reduce opportunity).

Our final recommendation is to restrict candidate’s access to the examination’s master question pool. The success of our approach on the mock and the ProCert examinations was because the test generation software used to serve the examinations included the question pools. As a result, whenever a new assessment was started, a new set of questions were drawn from the pool and further questions could be recovered and recorded if the same process could be replicated on a virtual machine. Hence, an

<table>
<thead>
<tr>
<th>Countermeasures</th>
<th>Routine Activity Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote monitoring and supervision</td>
<td>Reduce Opportunity</td>
</tr>
<tr>
<td>Enforce one-time use of activation code</td>
<td>Enhance Guardianship</td>
</tr>
<tr>
<td>Impose time limit on both the entire test and individual test questions</td>
<td>Reduce Motivation</td>
</tr>
<tr>
<td>Prevent examinations executing in virtual environments</td>
<td></td>
</tr>
<tr>
<td>Do not provide entire question pool to candidate</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. A Routine Activity Theory Approach to Designing Countermeasures

Our fourth recommendation is that computer-based examination providers should prevent the execution of their examination application in a virtual machine environment. For example, Lau and Svajcer (2010) as well as Raffetseder, et al. (2007) have discussed how several families of malware now include virtual machine detection and execute differently to avoid security researchers performing any form of malware analysis. This technology could also be embedded within computer-based examination applications to avoid candidates executing the examination in virtual environments and extracting questions using the approach described in this paper (i.e. enhance guardianship, and reduce opportunity).

Our final recommendation is to restrict candidate’s access to the examination’s master question pool. The success of our approach on the mock and the ProCert examinations was because the test generation software used to serve the examinations included the question pools. As a result, whenever a new assessment was started, a new set of questions were drawn from the pool and further questions could be recovered and recorded if the same process could be replicated on a virtual machine. Hence, an
alternative solution is that the computer-based examination generated for each candidate should be unique and only include the questions for that candidate and not the entire question pool (i.e. reduce opportunity). This would reduce the incentive to duplicate the environment with the anticipation of recovering more questions than what is currently presented in the examination, and exploit them either in the current examination or future settings.

**Conclusions and Future Work**

The results from the analysis of the mock examinations and the case study of ProCert indicate that it is possible to recover assessment questions from computer-based examination applications. In our analysis of two mock examinations using Test Generator Lab Version 6.0 and Exam Version 16.9, we were able to record and recover all the questions we had used to populate the question pools for these computer-based examinations. The paper also highlighted how our approach can be used to extract 1001 questions from a master question pool within a computer-based examination application used by a popular professional certification examination that has been attempted by potentially thousands of candidates around the world.

In order to counter the use of our approach for recovering questions from computer-based examination applications, we proposed five recommendations including implementing remote monitoring of examinations, enforcing software restrictions on the computer-based examination application, imposing time limits on the test, prohibiting the use of the virtual machine environment, and not provide entire examination pools to the candidate. These recommendations were designed to ‘reduce opportunity’, ‘enhance guardianship’ and ‘reduce motivation’, the three areas prescribed in the Routine Activity Theory (Cohen and Felson 1979).

We hope that the results of our research encourage organizations, universities and certification providers, who offer computer-based examinations, to perform a thorough investigation of their examination applications in virtual environments to prevent candidates recovering questions using the approach discussed in this paper. Moreover, we encourage examination providers to examine our recommendations to enhance the credibility and integrity of their computer-based examinations.

Future work will include an investigation into other professional examinations in a variety of testing environments (e.g. mobile devices, online, within a computer network), as well as an extension of the forensic techniques used to recover and record examination questions. For example, previous research (Cahyani et al. 2017; Grispos et al. 2014; Grispos et al. 2015) has shown that information can be recovered from residual data generated by mobile devices. Similarly, we anticipate that examination questions could be recovered from residual data stored in mobile devices used to administer and deliver tests and examinations. Therefore, further work may attempt to use additional types of computer forensic tools with the objective of identifying what residual data concerning a computer-based examination can be recovered from a variety of devices. Further research is also needed to investigate the suitability of our countermeasures as viable approaches for preventing the recording and recovery of questions using computer forensic tools. This could take the form of a case study with an organization, such as the one that distributes ProCert. The aim of the case study is not only to evaluate the feasibility of the countermeasures, but to also identify and examine the complications and challenges that could arise from their implementation.

**Acknowledgements**

The second author was partially supported by SFI Grant No. 13/RC/2094 and ERC Advanced Grant. No. 291652 (ASAP).

**REFERENCES**


