Beyond Lockdown: Towards Reliable e-Assessment

Bastian Küppers[1], Florian Kerber[2], Ulrike Meyer[3], Ulrik Schroeder[4]

Abstract: In modern university education, lectures and accompanying exercises and tutorials incorporate digital elements to keep up with a general trend of digitalization. These digital elements spread from incorporating online learning management systems into the lectures to the usage of computers and tablets in exercises and tutorials. Despite e-Assessment being a valuable component in form of self-tests and formative assessment, the trend of digitalization has not yet been transferred on examinations. Retaining examinations on paper is often caused by reservations against e-Assessment, but also financial reasons interfere with the establishment of e-Assessment, because maintaining a suitable IT-infrastructure for e-Assessment is expensive in terms of money as well as administrative effort. Bring Your Own Device is a potential solution to this issue, but also poses new challenges regarding the integrity and reliability of examinations, hence potentially boosting the existing reservations. A common approach for securing e-Assessment is the usage of a so-called lockdown software on the students' devices, which is intended to ensure that students cannot execute impermissible actions during the examination. This paper exposes the disadvantages of current lockdown approaches in the Bring Your Own Device setting and outlines a novel alternative to securing e-Assessments. It thus contributes to reducing reservations against e-Assessment.

Keywords: E-Assessment, Digital Examinations, Bring Your Own Device, BYOD, Lockdown.

1 Introduction

Retaining examinations on paper is often caused by reservations against e-Assessment. These reservations concern for example the fairness or reliability of digital examination systems [VS09]. Especially the students have to accept e-Assessments before it can be introduced to higher education [FPH09, TE11]. Hence, when introducing e-Assessment to an institute of higher education, security is an often-discussed issue [VS09]. In any technical study program and in computer science programs in particular, the risk of fraud and cheating is presumably higher than for other study courses, because the particular students are studying the tool for e- Assessment - the computer. Thus, a proper mechanism to secure e-Assessment has to be used to ensure that it can be carried out reliably. Existing mechanisms, however, have some drawbacks especially in Bring Your Own Device (BYOD) settings. These include requiring administrative access rights to the BYOD

[1] RWTH Aachen University, IT Center, Seffenter Weg 23, 52074 Aachen, kueppers@itc.rwth-aachen.de
[2] RWTH Aachen University, IT Security Research Group, Mies-van-der-Rohe Str. 15, 52074 Aachen, kerber@itsec.rwth-aachen.de
[3] RWTH Aachen University, IT Security Research Group, Mies-van-der-Rohe Str. 15, 52074 Aachen, meyer@itsec.rwth-aachen.de
[4] RWTH Aachen University, Learning Technologies Research Group, Ahornstraße 55, 52074 Aachen, schroeder@cs.rwth-aachen.de

device and cross platform use on the usability side and a potentially incomplete threat model on the security side. In this paper, we therefore propose an alternative approach called LogDown. Our approach is based on a client-server architecture in which suspicious behaviour is logged and reported to the server. Thus, cheating during the eAssessment is discouraged as it will be detected. The rest of the paper is structured as follows: Related work is briefly discussed in Section 1.1., Section 2 details the drawbacks of lockdown software and Section 3 sketches our novel approach. We close with a conclusion and an outlook in Section 5.

1.1 Related Research

Several different approaches to ensure reliability already exist. For example online proctoring via remote-desktop software or surveillance over the webcam [Fr10]. These approaches induce, however, a lot of effort, because invigilators have to be available to review the remote desktop session or the webcam, keeping aside possible privacy issues. Therefore, a so-called lockdown software is most often used to prevent illegal actions during the e-Assessment [Fr10]. Once it has been started, this software controls the students' devices in a way that only certain actions can be carried out, e.g. starting a particular program or visiting white-listed webpages. Examples for existing software solutions are the Safe Exam Browser (SEB), which is an open source project developed at ETH Zürich[5], and LockDown Browser, which is a closed source application by the company Respondus[6]. These tools, however, have some drawbacks, which introduce vulnerabilities that potentially render these tools attackable. While corresponding attacks require some technical effort and administrative access to the computer the tools are running on, e-Assessment on a centrally managed IT-infrastructure is not is not particularly vulnerable to these attacks. For BYOD e-Assessment, however, these vulnerabilities have to be overcome, since students usually have administrative access to their devices and time to prepare their devices before an examination.

2 Drawbacks of Lockdown Software

The drawbacks with lockdown software concern mainly two main aspects: usability and security. Both will be discussed in the following paragraphs. While the below mentioned drawbacks in usability could be acceptable, the drawbacks regarding security can potentially sabotage the fairness and reliability of e-Assessments.

2.1 Usability

Since lockdown software tries to control the working environment completely, it has to

[5] http://safeexambrowser.org
[6] http://www.respondus.com/products/lockdown-browser

interact with the operating system inevitably. Therefore, the software needs administrative privileges, at least while installing the software, in order to secure the working environment (see for example the manuals of SEB [ET17] and LockDown Browser [Re17]). In a BYOD setting, however, it cannot be guaranteed that students have administrative privileges on their devices. For example, a student employee, who is allowed to use a laptop provided by the employer, does not necessarily have administrative privileges on that device. Therefore, a lockdown application, which is intended to run on the students' devices flawlessly, must not require administrative privileges in order to establish a fully functional and secure BYOD scenario for e-Assessment.

Additionally, the available tools are not designed for cross-platform use. SEB supports Windows and MacOS, LockDown Browser can be used on iOS and ChromeOS additionally. To our knowledge, however, none of the available tools supports Linux, let alone all common platforms. Therefore, some students may not even be able to use a lockdown software on their accustomed operating system.

2.2 Security

Lockdown software attempts to provide a secure working environment by preventing potential security threats. Since SEB is open source, it is easy to get an insight into the methodology that it uses to implement this strategy. Therefore, SEB will be discussed here, but the results seem equally valid for other available lockdown software, because the feature set for that type of software is similar for all available products [FL13].

SEB uses many of the possibilities that MS Windows offers in order to secure the working environment. The implemented measures range from creating a new desktop object, which does not provide the default GUI, up to the use of windows hooks in order to get aware of occurring events. This approach is a negative security approach [Tr09], i.e. a threat model has been defined and for each threat, a counter measure was implemented to prevent the threat. The general problem with negative security is that it does not help to build trust, because it is hard to prove that every potential threat is covered in the threat model [As13].

Indeed, it seems that the threat model of SEB does not cover all potential threads [Sø16]. In fact, we carried out some tests ourselves and were able to inject a DLL file into SEB by using common DLL injection techniques [HB99]. The injected DLL allowed us to potentially execute any code. We used it to open a new window that displays information about the topic of the examination as a proof-of-concept. SEB could of course include our particular approach into the threat model, but that does not solve the problem in general.

3 An Alternative Approach: LogDown

Due to the previously discussed drawbacks of lockdown software using the example of SEB, we conclude that lockdown software does not offer suitable security for e-Assessment, especially not in a BYOD setting. Therefore, we propose an alternative

approach, which makes use of a client-server architecture. Our approach does not try to prevent cheating, but to detect suspicious behaviour, which is then reported to the server (hence the name 'LogDown'). Therefore, no deeper embedding into the operating system is necessary and thus no administrative privileges are required. Additionally, since there is only a loose coupling with the operating system, porting LogDown to a new platform is a lot easier.

LogDown is designed in a way that it only monitors itself, not the whole working environment. Broadly speaking, it checks if any communication channel, e.g. keyboard or screen, shows suspicious behaviour. To accomplish this goal, several things have to be checked, for example running processes or the contents of the clipboard. As opposed to using a threat model, this approach uses more of a safety model.

In order to make the approach taken with LogDown work, it has to be ensured that an unaltered version of LogDown is running on the students' devices, otherwise attacks like DLL injections or code injection in general would render LogDown useless. Since the students' devices are untrusted platforms from the examiner's point of view, there is virtually no possibility to ensure that students have not altered the LogDown executable when they are starting it. Since the executable could be altered, implementing a self-check, which is executed at the start, is pointless. It is, however, possible to verify that the executable has not been altered at run-time.

In order to do that, several steps have to be taken. First, every started instance of LogDown has to register itself at the server, which in turn provides the assignments of the examination. Hence, no registered version of LogDown means no assignments. In order to be able to match the registered instances of LogDown with the corresponding students, public key cryptography is used. All students have to register their key-pair prior to the examination. That key-pair is then used to establish a TLS connection with client authentication between LogDown and the server. This connection is held active throughout the whole examination and takes up the role of a heartbeat measurement. Should this connection break down for any reason, this is immediately reported to the examiner, who can then take action as needed and e.g. instantly stop the examination for the student in question. This is necessary, because a breakdown of the connection could have also different reasons than a cheating attempt, e.g. a problem with the Wi-Fi connection. Therefore, an automated termination of the examination could be wrongful.

To verify the integrity of the client, the server has a set of verification binaries. These verification binaries take up the role of a challenge-response authentication between the server and the LogDown executable. At random intervals, the server selects a random verification binary as challenge and sends it to the client to be executed with the expectation to receive a timely answer. A late or wrong answer as well as no answer at all will result in the immediate notification of the examiner, as this possibly indicates a tamper attempt. Restricting the time for an answer to the server and sending out the verification binaries in random time intervals is necessary, as the verification binaries provide integrity checking code as well as the answer of the challenge, which is sent back to the server in

case of a successful check. Without these measures, analysing the binary and extracting the correct answer without an actual genuine LogDown executable would become feasible. To prevent pre-computations of challenge results as well as raising the bar for potential attackers, the verification binaries could be personalized or compiled just-in-time with random optimizations or code fragments. This way, a potentially compromised client would be unable to collect all possible verification binaries. Additionally, a successful verification allows reusing the verification binary in question, as the integrated self-check ensures the sanity of the platform. Therefore, no verification binaries are collected. A failed test could indicate a potential security breach and therefore the leak of the binary used, but due to the massive amount of potential binaries as well as the knowledge of the actual actor, due to identification via the TLS certificates, the number of attempts is restricted to one per person. In order to prevent leakage of the contents of the examination, the assignments are only sent to the students' device after the first successful execution of a verification binary. Figure 1 shows the workflow of LogDown.

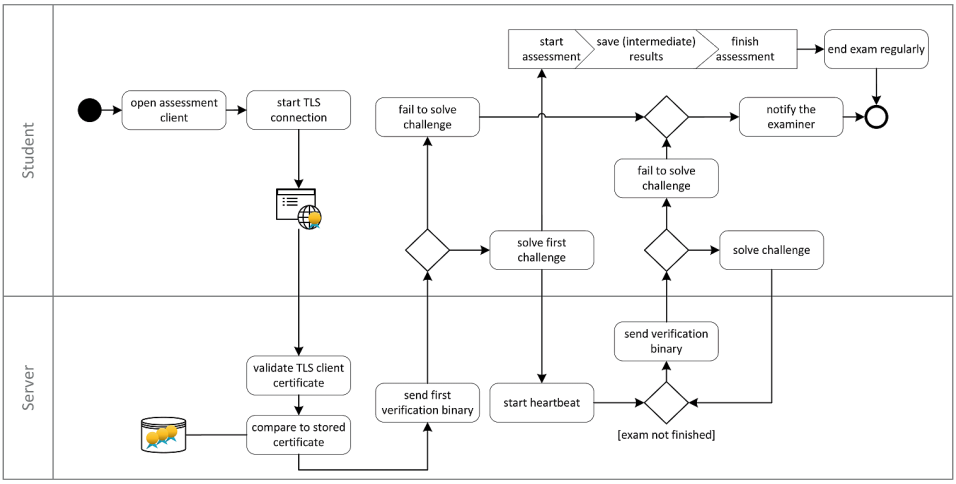

Fig. 1: Workflow of LogDown.

Unlike other hacking scenarios, where the aggressor can act out of anonymity, our approach removes this cover and therefore creates a psychological inhibition threshold, since even a single unsuccessful attempt to cheat is detected and can result in consequences for the student. Therefore, we expect that the willingness to cheat in an examination will be low in comparison to a regular lockdown application, despite the students' not being constantly monitored. Since the students can never know whether a verification binary is executed at a time or when the next execution is due, cheating is very risky. That situation resembles the state-of-the-art in paper examinations, where students could have crib sheets, but would risk to be caught by an invigilator when using these.

4 Summary and Future Work

In this paper, we discussed the general approach of conventional lockdown applications for e-Assessment systems, and pointed out their drawbacks and insecurities, especially in a BYOD scenario. Based on the results of this discussion, we proposed an alternative approach using a monitoring client-server architecture, which utilizes integrity checks at random time intervals. This does not only decrease the intrusiveness of the e-Assessment, it also increases the overall security due to its on-the-fly self-checks. Therefore, our approach potentially leads to a higher acceptance of e-Assessment and BYOD. Our next steps will be to implement the presented approach prototypical and run test in order to become aware of potential pitfalls. Especially the portability for different, common operating systems will be an issue that has to be addressed while implementing and testing.

Bibliography

[As13] Association, Information Resources Management: Crisis Management: Concepts, Methodologies, Tools and Applications. IGI Global, Hershey, PA, USA, 1st edition, 2013.

[ET17] ETH Zurich, Educational Development and Technology (LET): Windows User Manual. Website, 2017, http://safeexambrowser.org/windows/win_usermanual_en.html, Stand: 17.07.2017.

[FL13] Foster, David; Layman, Harry: Online Proctoring Systems Compared, 2013.

[FPH09] Fluck, Andrew; Pullen, Darren; Harper, Colleen: Case study of a computer based examination system. Australasian Journal of Educational Technology, 25/4, 2009.

[Fr10] Frank, Ariel J.: Dependable distributed testing: Can the online proctor be reliably computerized? In (Marca, David A., Hrsg.): Proceedings of the International Conference on E-Business. SciTePress, S.1, 2010.

[HB99] Hunt, Galen; Brubacher, Doug: Detours: Binary Interception ofWin32 Functions. In: Third USENIX Windows NT Symposium. USENIX, S. 8, July 1999.

[Re17] Respondus Technical Support: Knowledge Base. Website, 2017, http://support.respondus.com/support/index.php?/Knowledgebase/Article/View/147/25/i need-administrator-rights-to-install-respondus-lockdown-browser-windows, Stand: 17.07.2017.

[Sø16] Søgaard, Thea Marie: Mitigation of Cheating Threats in Digital BYOD exams, 2016.

[TE11] Terzis, Vasileios; Economides, Anastasios A.: The acceptance and use of computer based assessment. Computers & Education, 56/4, S.1032–1044, 2011.

[Tr09] Trost, R.: Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century. Pearson Education, 2009.

[VS09] Vogt, Michael; Schneider, Stefan: E-Klausuren an Hochschulen: Didaktik - Technik – Systeme - Recht - Praxis, 2009.