Mobile Authentication for Software Engineering

A case study of research and development student projects

Stan Kurkovsky
Department of Computer Science
Central Connecticut State University
USA
kurkovsky@ccsu.edu

***Abstract*—This paper presents an ongoing work on integrating computer security topics into project-based software engineering courses. We describe two projects, in which students follow an incremental delivery software process and a scrum-like software process to design and implement two different authentication systems for mobile devices. The project topics and the software processes were matched to better reflect the nature of the implemented systems.**

Keywords—software engineering; mobile computing; computer security; authentication; student projects

I. INTRODUCTION

Security awareness has become an essential part of our everyday lives, in which we rely on computing systems for the widest possible array of tasks ranging from banking and shopping to social interactions and entertainment. Software manufacturers, regardless of their number of employees, the scale or complexity of their software, or the nature of their software product, are now expected to make their software trustworthy, reliable, and secure. At the same time, academic programs producing software engineers are not always up to speed in terms of incorporating computer security essentials into their curriculum.

A recent report by Ponemon Institute [14] indicates that a security breach leading to a cybercrime costs a US company an average of $15.4 million in 2015, a 19% increase from 2014. A report by Software Engineering Institute [17] produced for the US Department of Homeland Security explains that the vast majority of these breaches were, at least partially, attributed to 'unintentional insiders,' which include both end users and computing professionals responsible for operating and developing software systems. These reports underline the importance of security awareness across all strata of the society and professions. Furthermore, they also serve as a reminder that the current software engineering workforce may not be up to par when it comes to possessing knowledge and skills related to computer security.

Information Assurance and Security has been identified as a new knowledge area in the Computer Science Curricula 2013 developed jointly by ACM and IEEE Computer Society [1]. As noted in the Curricula [1], this knowledge area "is added to the Body of Knowledge in recognition of the world's reliance on information technology and its critical role in computer science education." Presently, most computer science programs implement this element of the curricular guidelines by offering a standalone course in computer security. Consequently, computer security may end up being treated in isolation from many other areas of the curriculum. Without making a computer security course mandatory, this may also allow the program to produce some graduates who will never be exposed to the fundamentals of computer security.

Many computer education practitioners believe that computer security topics should be embedded into a broad range of courses and supported by practical exercises that would help students solidify their knowledge and skills. Some courses, such as software engineering, are especially well-suited for giving a special consideration to computer security topics. To this end, many standard software engineering textbooks, such as those by Pressman [15] and Sommerville [18] intertwine information assurance and computer security elements throughout the coverage of many other topics and areas of software engineering.

This paper reports on our experience in embedding a substantial amount of computer security content into a project offered as a required part of senior-level (4th year) software engineering courses. We describe two semester-long student projects completed by various student teams over the last few years. Both projects focus on implementing an authentication mechanism for mobile devices: an experimental combination of behavioral and password-based method (rhythm tapping) and a well-established biometric method (iris identification).

II. RELATED WORK

The need to integrate computer security into the computing curriculum has been recognized long before it has been designated as a standalone Knowledge Area in Computer Science Curricula 2013 [1]. From the practical viewpoint, adding a single computer security course has been the simplest and the most common way of adding the coverage of computer security into the curriculum [13]. A number of approaches ranging from playing board games [7] to participating in research-oriented projects [10] have been proposed to embed computer security concepts into the computing curriculum. Another approach is to incorporate computer security topics throughout a number of existing courses, which is illustrated below.

Siraj et al [16] developed a framework of instructional modules focusing on different aspects of computer security that can be easily integrated into a broad variety of traditional computer science courses. The framework aims to ensure that computer security is treated as a multidisciplinary science, which is achieved by making connections with such traditional areas of computing as database management and operating systems principles. The framework's modules are intended to help faculty who have little to no background in computer security to easily add coverage of the relevant topics into existing and well-established courses.

Guo et al [9] designed an Android security labware aimed to promote the study of mobile security. Similarly to the work described above, this labware consists of several learning modules suitable for inclusion in mobile computing courses. Taken together, the modules offer a comprehensive coverage of mobile device security and privacy, as well as mobile application, network, and communication security. Although highly applicable to any course related to mobile computing, this labware is very difficult to adapt in any other context, e.g. a traditional software engineering course unless the students already have a substantial background in mobile application development.

These two sample projects on integrating security concepts into the curriculum by using learning modules illustrate a common broader problem. Offering a single computer security course or several courses providing the coverage of the topic expose students to a number of computer security concepts. However, this approach does not provide students with an integrative experience, which would use a highly practical context to apply student knowledge and skills in computer security along with their knowledge and skills of core computer science areas. In this paper, we describe the experience of integrating the topic of user authentication, a computer security area of high practical importance and relevance to the student everyday experience, into a large project completed by students in upper-level software engineering courses.

III. Alternatives to Traditional Passwords

Internet-enabled mobile devices have completed the process of transforming from a simple tool for personal communication to an omnipresent complex device kept by many people with or on them 24 hours a day. Today, mobile devices are used to conduct everyday activities, such as online banking, electronic payments, keeping personal and professional contacts, and accessing many online services that previously were used exclusively on desktop computers. Consequently, many of us end up keeping a substantial amount of personal data on mobile phones and tablets. This kind of information may include address book entries, financial data, and login records. Such a fundamental evolution of the mobile device functionality, as well as their wide popularity and availability at different price points, inevitably increases the chances of device theft and loss. Furthermore, the severity of consequences from misusing the information stored on these devices as a result of loss or theft increases exponentially. Despite many technological advances made by the industry and reported in many published research works in this area, security features of most mobile devices remain as ineffective as they were a decade ago. Password or PIN protection do not provide an adequate degree of security because passwords can be forgotten or stolen, if stored inappropriately. The biggest threat to password-based authentication, however, comes from the fact that many users choose simply not to use their PINs of passwords, or fail to replace the default values with a personalized PIN or password on a newly purchased device. Current research studies suggest that many mobile users view PIN-based security techniques as an inconvenience or something that they find cumbersome [19]. At the same time, using non-text passwords or biometric characteristics may offer convenient solutions to the problem of user authentication offering a reasonable balance between ease of use and strength of security.

Biometric technology uses one or a combination of several physical characteristics including voice, fingerprint, face, iris, retina, ear shape, palm, veins, etc. Behavioral characteristics including handwriting [2] and walking [11] could also be used in biometrics. Facial recognition has been implemented by a number of Android device manufacturers such as Samsung [4], but the reliability of current implementations remains subpar. Fingerprinting technology has also been widely adopted by major hardware manufacturers.

Except for fingerprinting, which requires special-purpose hardware to obtain a high quality fingerprint sample, most biometric and non-password authentication techniques are well-suited for adoption as a source for hands-on experimentation and implementation in the computing curriculum. Generally, these authentication methods are used in an identification mode when a large database is searched exhaustively to find a match, as well as in a verification mode when an acquired sample is matched to one or several previously observed records. The primary objective of any authentication method employed on a mobile device is to protect the device from an unauthorized access. In a typical scenario, only a single individual owns and uses a mobile phone, a tablet, or a similar device. Consequently, a mobile authentication system will be employed mostly in the verification mode when a biometric sample taken from the person attempting to turn on the device or log in to it is matched with that of the rightful owner of the device. The objective of this kind of mobile authentication system will not be to compare multiple samples; as a result, the computational complexity of the system can be reduced substantially, which further improves its applicability in the educational context.

IV. Case Studies

User authentication is one of the core topics in computer security making it a worthwhile candidate as a subject for student projects. Studying and implementing user authentication techniques on mobile devices allows students to practice a very important theoretical concept in a socially-relevant context. Here, we describe two projects that were used in upper-level software engineering courses with a heavy emphasis on computer security. In particular, these projects were designed to provide students with an in-depth exposure to different approaches to authentication on modern mobile devices.

Both projects described below were completed by several student teams. Each team worked independently on their own implementation of the project. For each project, the teams followed different software processes, as described below.

V. RHYTHM TAPPING-BASED AUTHENTICATION

Tapping-based authentication has been described in a number of research reports. Wobbrock [20] describes TapSongs, a system in which users can unlock their mobile device using a rhythmic sequence of binary values generated by pressing/depressing a single button, such as one of the volume control buttons located on the cord of many earbud models. Marques et al [12] implemented a modification of this approach by simplifying the authentication enrollment procedure and specifically testing it in an environment where the users can authenticate without looking at the device. This authentication technique makes it suitable not only for touch-screen mobile devices, but also for low-end devices without a touch screen. Furthermore, this approach may be especially relevant to the emerging class of IoT (Internet of Things) devices with no screens, such as smart smoke detectors, door locks, and security cameras [8].

Zheng et al [21] take a broader view on tapping-based authentication. From the perspective of behavioral biometrics, each user has individual traits reflected in the timing, acceleration, strength, and size of each finger taps. Taking into account the larger number of behavioral features, this approach results in a higher quality of recognition yielding lower false accept and false reject rates.

All three research reports described above applied a feature vector similarity measure to compare the enrollment sample provided by the authentic user with the sample representing the current login attempt. Another common advantage of these tapping-based authentication schemes is that they make it convenient for the user to create and use an easy-to remember authentication sequence, typically based on a melody or a jingle. At the same time, aural and visual eavesdropping over the users practicing this kind of authentication technique typically results in a low rate of successful login attempts by an impostor.

Several student teams completed this project in parallel. All teams followed a scrum-like software process [3]. The steps outlined below were initially identified as larger user stories ('epics') to be placed in the product backlog. Students quickly realized that each one of them was a good candidate to be broken down, with each epic producing enough user stories to be implemented during a single sprint. This resulted in three sprints, which was a good fit for a single-semester course project. All teams completed their implementations on the Android platform using Java.

A. Tapping sequence capture

A single tapping event on a typical touch-screen device can be broken down into two components, touch-down and touch-up, reflecting the beginning and the end of applying the finger pressure to the screen. A touch event can also be described by the amount of pressure, the size of the screen area in contact with the finger, its coordinates, and duration. Time intervals between each two consecutive touch events are essential for implementing this authentication scheme. The implementation also needs to be sensitive to timeouts to determine whether the user is done entering their authentication sequence or whether they decided to abort the current attempt. It is not necessary to use all of the above-mentioned parameters of a touch event for the purposes of constructing a feature vector to match with the login record. The key to authentication using tapping sequences lies in comparing the time intervals between the touch events. However, adding additional parameter could improve the accuracy of the technique. It was left up to the student teams to determine the final selection of parameters to be used based on several iterations of experimenting with the resulting system.

B. Comparison and normalization of tapping sequences

The designed authentication system must be able to enroll a new user by recording or learning a feature vector (a tapping sequence) that would uniquely identify the correct user. Given the non-discrete nature of each feature vector of parameters describing a tapping sequence, the vectors cannot be expected to match precisely. It is very likely that some of the parameters of each touch event described above would exhibit some variations. That requires taking into account some practical considerations regarding the enrollment sequence. Instead of requiring the user to enter a single enrollment sequence, it may be useful to have the system acquire several repetitions of the tapping sequence. This way, each parameter can be supplemented with its standard deviation. During the verification phase, these standard deviations could be taken into account in the feature vector similarity score calculations. Furthermore, from time to time, the user may enter the correct sequence, but at a different pace, sometimes a little slower, sometimes a little faster. The authentication technique may be extended to normalize the sequences entered at the verification phase by stretching or squeezing each time interval between the subsequent taps so that the overall duration matches that of the enrollment sequence.

It may be possible to reject an authentication attempt outright before applying the feature vector similarity measure. For example, a mismatch between the number of taps in two sequences is an immediate indication that they will not match. Students we also encouraged to identify any other conditions leading to the same outcome.

C. Experiment with different characteristics of the tapping sequence

This student project was specifically designed to allow students to experiment, within the frameworks of the given software process, with different elements and extensions of the implemented system. For example, it was suggested to attempt distinguishing between the taps made with different fingers by using the touch coordinates. Some teams chose to spend one of the sprints as a spike, during which they specifically explored the viability of these different modifications to the baseline authentication scheme. Doing so not only provided student teams with an opportunity to apply different agile practices in this project, but it also allowed them to further improve the authentication precision of the implemented system.

VI. SIMPLE IRIS RECOGNITION

Daugman [6] describes the theoretical principles of iris recognition technology that enabled the development of many commercially successful and robust biometric authentication systems over the last two decades. Students were asked to implement a simplified process of iris recognition specifically modified to work on mobile devices with limited resources. This

approach does not require any additional or specialized hardware, while resulting in authentication quality metrics that are on par with those of proposed and existing systems described in the current research literature [19].

This project was offered to several teams of 4th year students. Each team worked independently following a simple incremental delivery software process. Since the algorithmic procedure of iris recognition method described below can be divided into a number of discrete steps, functional requirements of each step were a good candidate to be implemented during each iteration. Students performed a thorough integration testing ensuring that each new layer of functionality works correctly with the previous layers. Some teams chose to implement the system on Android platform using Java, while other teams used C# on Windows Mobile.

A. *Image acquisition, eye detection, and segmentation*

Most mobile phones and tablets have front-facing cameras, which makes it very easy to acquire an image of the eye. If the captured image contains the entire face or its portion, it needs to be segmented to localize the portion containing one of the eyes. There are a number of efficient techniques for detecting and localizing the eye that are straightforward to implement.

B. *Iris localization*

High frequency noise in the image can be filtered out with a 3x3 median filter. Consequently, inner and outer boundaries of the iris are identified. A preliminary step involves standard edge detection applied to the noise-reduced image. Circular Hough transform can be used to identify the circular outlines of the iris boundaries in the resulting image containing iris edge highlights. Circular Hough transform can be easily simplified by assuming that the pupil is situated in the middle area of the image of the eye that has already been divided into segments during the preceding step of the process. Furthermore, the boundaries of the pupil are represented by the inner and outer circles that are approximately concentric. This provides an additional element of leverage to further simplify the computations. Occlusions of eyelids and eyelashes can be ignored to further reduce the resulting computational complexity of the entire process.

C. *Normalization and feature extraction*

Biometric samples containing the iris images typically come in slightly different sizes since each image may represent a different person, and could have been acquired using a different distance from the camera, using different lighting conditions, and using varying camera angles. The resulting extracted image of the iris requires normalization in order to make the images more suitable for the processing and the subsequent steps of the process. The majority of biometric authentication systems described in the literature that use iris recognition use a special technique of 'unfolding' a circular image of the iris and transforming it from polar to Cartesian coordinates. The resulting image is a rectangle with its vertical axis corresponding to the radial coordinate of the original circular image, and the horizontal axis corresponding to the angular coordinate of the original. Such a transition results in the inner iris boundary lying at the rectangle's top and the outer iris boundary lying at the bottom of the rectangle. This transformation provides a way to efficiently apply Gabor filter in order to obtain the iris texture information (the feature vector) from the biometric sample. Student teams working on this project were advised to bypass the two steps described above. The recognition process can be significantly simplified by using polar coordinates without transforming a circle into a rectangle on a Cartesian plane, which allows creating a feature vector directly from the extracted image of the iris.

D. *Feature vector matching*

It was recommended to the teams to use 32 sectors and 8 tracks in order to partition the resulting extracted image of the iris. Each pixel of the extracted iris image could be mapped using polar coordinates to its corresponding location at the intersection of the correct track and sector. This is followed by computing the average value of the gray scale color for each one of the resulting 256 sections, which forms a complete feature vector. While in the verification mode of the authentication technique, the feature vectors representing different biometric samples are compared with each other using a thresholding technique, e.g. Hamming Distance. Student teams used the average difference between the two feature vectors to establish a measure of vector similarity.

VII. Effect on Student Learning

Formal and informal feedback collected from students who participated in software engineering projects to implement either of the above systems strongly indicated that the students appreciated the opportunity to work on a computer security project grounded in a strong theoretical foundation. Furthermore, students responded very positively to the opportunity to make their own contributions to the design and implementation of various features of the described authentication techniques by making them more robust. As with many integrative projects offered in senior software engineering courses, students appreciated the opportunity to bring together their knowledge and skills from various areas of computing, such as algorithms, operating systems, database management, and apply them in the context of a single system.

VIII. Summary

The importance of providing students with an integrative experience that allows them to apply a broad range of skills and knowledge acquired during their course of study is very difficult to dispute. Choosing a practical, important, and socially-relevant context for such an experience helps keep the students engaged in the coursework while providing them with an opportunity to work on projects that are similar to those they are likely to experience upon graduation once they become professional software developers. Software projects incorporating or focusing on various computer security topics meet all of the above criteria. Providing future professional software engineers with an opportunity to experiment with well-established or implement novel authentication techniques for mobile devices has a great potential to solidify their skills in research and development.

REFERENCES

[1] ACM/IEEE-CS Joint Task Force on Computing Curricula (2013). Computer Science Curricula 2013. ACM Press and IEEE Computer Society Press. DOI: http://dx.doi.org/10.1145/2534860.

[2] R. Blanco-Gonzalo, O. Miguel-Hurtado, A. Mendaza-Ormaza, and R. Sanchez-Reillo, "Handwritten signature recognition in mobile scenarios: Performance evaluation," Proc. *2012 IEEE International Carnahan Conference on Security Technology,* pp. 174-179, 2012.

[3] J. Campbell, S. Kurkovsky, C.W. Liew, and A. Tafliovich. "Scrum and Agile Methods in Software Engineering Courses." In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (SIGCSE '16). ACM, New York, NY, USA, pp. 319-320, 2016.

[4] K. Choi, K.-A. Toh, and H. Byun, "Realtime training on mobile devices for face recognition applications," *Pattern Recognition*, vol. 44, no. 2, pp. 386-400, 2011.

[5] P. Corcoran, "Biometrics and consumer electronics: a brave new world or the road to dystopia?" *IEEE Consumer Electronics Magazine*, vol. 2, no. 2, pp. 22-33, 2013.

[6] J. Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.

[7] M. Gondree, Z. Peterson and T. Denning. Security through play. *IEEE Security & Privacy*, *11*(3), pp.64-67, 2013.

[8] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, vol. 29, no. 7, September 2013, pp. 1645-1660.

[9] M. Guo, P. Bhattacharya, M. Yang, K. Qian, and L. Yang, "Learning mobile security with android security labware," In *Proceeding of the 44th ACM technical symposium on Computer science education*, pp. 675-680. ACM, 2013.

[10] S. Kurkovsky, T. Carpenter and C. MacDonald. Experiments with Simple Iris Recognition for Mobile Phones. *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, Las Vegas, NV, 2010, pp. 1293-1294.

[11] J. Kwapisz, G. Weiss, and S. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74-82, 2011.

[12] D. Marques, T. Guerreiro, L. Duarte and L.Carriço. Under the table: tap authentication for smartphones. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*, pp. 33-39. British Computer Society, 2013.

[13] L.F. Perrone, M. Aburdene, and X. Meng, "Approaches to undergraduate instruction in computer security," In *Proceedings of the American Society for Engineering Education Annual Conference and Exhibition, ASEE*. 2005.

[14] Ponemon Institute, 2015 Cost of Cyber Crime Study, October 2015.

[15] R. Pressman, B. Maxim, Software Engineering: A Practitioner's Approach, 8TH Edition, McGraw-Hill, 2015.

[16] A. Siraj, S. Ghafoor, J. Tower, and A. Haynes, "Empowering faculty to embed security topics into computer science courses," In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pp. 99-104. ACM, 2014.

[17] Software Engineering Institute, Unintentional Insider Threats: A Foundational Study, Technical note CMU/SEI-2013-TN-022, August 2013.

[18] I. Sommerville, Software Engineering, 10th Edition, Pearson, 2016.

[19] Q. Tao, R.N.J. Veldhuis, "Biometric Authentication for a Mobile Personal Device," in Proc. *2006 Annual Conference Mobile and Ubiquitous Systems*, pp. 1-3, 2006.

[20] J. Wobbrock. "Tapsongs: tapping rhythm-based passwords on a single binary sensor." In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pp. 93-96. ACM, 2009.

[21] N. Zheng, K. Bai, H. Huang and H. Wang. You are how you touch: User verification on smartphones via tapping behaviors. In *2014 IEEE 22nd International Conference on Network Protocols* (pp. 221-232). IEEE, 2014, October.