CSci 4271W: Development of Secure Software Systems

Project 2 **due: May 2nd, 2022**

Ground Rules. This is an individual assignment that each student should complete on their own. It's OK to help other students with understanding the concepts behind what we're doing in the project, or to help with technical difficulties, especially if you do so in venues like Piazza and office hours where the course staff are also present. But don't spoil the assignment for other students by telling them the full details of your answers: everyone should have the experience of figuring those out for themselves. There will be one online submission for the project on Monday, May 2nd, the last day of regular classes. The submission will be online, accessible from the course Canvas page, and the deadline time will be 11:59pm Central Time. You may use external written sources to help with this assignment, such as books or web pages, but don't get interactive help with the assignment from outside sources. You **must** explicitly reference any external sources (i.e., other than the lecture notes, class readings, and course staff) from which you get substantial ideas about your solution.

About this project. In this project, you will describe a design for and analyze the security of a system to let students take exams online. We will follow common usage in calling this kind of service "online proctoring", since it is an online substitute for monitoring a student taking an exam in person. You should write your report as if you are working for a company that is considering building and selling software or an online service for test proctoring. Your report will help the company plan its implementation and marketing efforts by envisioning how the system will work and what kinds of security properties it can be expected to provide. A key purpose of online proctoring is to prevent or detect student cheating, but cheating might take many forms. So a major part of the project is threat modeling to determine what cheating or other security concerns are important in the design of the system, and how the threats can be mitigated.

Online exam proctoring motivation. Many aspects of education and learning are migrating to taking place online rather than in-person: online was a significant trend even before being accelerated and highlighted by the COVID-19 pandemic. Various aspects of classes and learning differ in the challenges that arise when trying to perform them online. Video recordings of lectures can substitute for attending a lecture in person, though it may be harder to ask questions. On the other hand, science labs that require specialized equipment would be difficult to present online. When it comes to testing, it doesn't seem too hard to use a computer to collect student answers to questions of many kinds. But when the results of a test have high stakes for a student's grades, it is more challenging to ensure that students taking a test online conform to the rules of the test. For instance the rules for a test might prohibit students from using a calculator, collaborating with other students, looking up answers in a textbook, or having a friend take the test on their behalf. But just recording the answers that a student types on a keyboard, for instance, likely wouldn't be able to prevent or detect these prohibited activities.

Online proctoring proposes to address this need, to make the experience of taking a test online more like taking a test in person in the extent to which prohibited activities can be prevented or at least detected. The use of such a system/service would be a requirement

for taking a class online, and students would be asked to complete additional steps before, during, and after the exam to demonstrate compliance with the exam rules. For instance, the student might have to record video of themselves while taking the test to be monitored live or checked after the fact. One goal of such a system is to increase the confidence that students were following the rules of a test, to at least as high a level as taking a similar test in a traditional in-person classroom, or maybe even better.

Your job. For this assignment, you should imagine yourself as working for a company that is considering developing and selling a software system or a service (implemented with software), for providing online proctoring. The company's management is pretty sure they want to pursue this market, but they haven't decided on all the details of what they should build. Student cheating and some of the other risks associated with proctoring feel like kinds of security problems, so the company has asked you to use your security expertise to envision the best way to build such a system. One part of what you need to do is to sketch out a more concrete design for how the proctoring will work, such as what information should be collected, how it should be stored and/or processed, and what the software that does this will generally look like.

Your design should be practical enough that the company has a good chance of being able to implement it, but you also need to think carefully from a security perspective about what can go wrong and to what extent those risks can be mitigated. You should treat this as a kind of threat modeling, where a big part of the challenge is to foresee the variety of different threats that the system might face. Students attempting to cheat are one kind of threat, but you'll need to think more specifically about what different kinds of behavior might constitute cheating and what the system can do about them. You should also consider other scenarios that could hurt the company in other ways, such as interfering with revenue from the service, making students or educational institutions dissatisfied, or leading to negative publicity.

You can think of your security analysis as having two main audiences: the division of the company that will build the software, and the one that will market and sell it. For the development team, it is most important to provide advice that will help them build an implementation that meets as many security goals as possible. For the threats you identify, explain whenever possible how they can be either completely prevented or at least mitigated by features of the system or implementation choices. For the marketing team, you want to give them an idea of what level of security the system will be able to provide. For instance, you might determine that some kinds of cheating can be reliably prevented and others cannot. You want to give a balanced perspective on the likely strengths and weaknesses of the product, which the marketing team can use to decide who to sell the product to, how much to charge, and what the documentation should say.

Assumptions around your solution. To focus the scope of your project, here are some assumptions you should make about the product your company will be building and the context where it runs.

We've used the words system, service, and product mostly interchangeably. You are going to write some software, and you will charge money for the privilege of using that software. But it may be a mix of software that is downloaded and executed directly on the student's device(s), and services that run on your company's servers and are accessed over the network. If there are security implications to the choice between these approaches, that should be part of your design.

As a core use case for your product, imagine an undergraduate student at a 4-year public university taking a midterm or final exam in a medium to large course, who is taking the course from home because of a COVID-19 lockdown. Probably your product will be usable more broadly than this: for instance younger students, continuing education students, students in other countries, shorter quizzes, or other variations. But it's more important to focus your design on the core use case that considering such variations.

Your company's direct customer will be the educational institution like the university that is offering a class. Probably the institution will use money from student tuition to purchase your product and make it available to students. The students won't have much choice about using your product, at least if they aren't able to take a test on-campus. Students' satisfaction with the product is relevant, but matters only indirectly: for instance if student experiences are too bad, the university may switch to your competitor's product next year.

You can assume that the students using your product have a laptop running Microsoft Windows, a smartphone running Android, and a good enough Internet connection for live video (e.g., DSL or a cable modem). This intentionally simplifies away an issue that is a challenge in reality [1].

Part of the selling point of your product to the university is that it shouldn't require much new effort or involvement from the university or its staff. For instance, if some sort of monitoring is required, it should be done by an automated system created by your company, or people employed by your company. Your system might ultimately report an incident of cheating to the instructor for a course, but it should only do so if you are pretty confident that there really was cheating.

You should consider a broad definition of cheating in your analysis. You could review the definition of academic misconduct from the syllabus of this or your other courses for ideas, but if there are new creative ways of cheating in the online environment, you would want to be able to stop them even if they are different from cheating that used to happen in person. To put it another way, the adversary for cheating is a student who wants to get a good grade on the exam even though they don't understand the material well enough to get a good grade by taking the test according to the rules. Anything a student could do to achieve that could be relevant cheating.

Design and security analysis. The two main parts of your report should be describing the design of an online proctoring system that is a good design from a security standpoint, and then analyzing its security by describing the possible threats and to what extent each one is prevented or mitigated. Of course the design has to be at a higher level than would part of actually building the system. But you should describe the most details about the aspects of the design that are most important for security. A team of other people with relevant software engineering experience will use your design as the starting point for their implementation, and they will be able to fill in the rest of the technical details that you leave out. You should assume that these developers are basically competent, so you don't need to waste time making general suggestions that all developers should know about, like that they should test their code or that they should not dereference null pointers. But it is more valuable as the security expert to point out situations where one aspect of the implementation is important for security: for instance, situations where an incorrect approach would seem to work fine in normal testing but be vulnerable to attack.

The security analysis part of the project is intentionally similar to the ideas of threat modeling we've done in class. Thinking about assets to protect and attacker motivations, modeling the data flow between parts of a system, and enumerating threats from the STRIDE taxonomy are all likely relevant. (Because the system isn't implemented yet, it's a bit more like some of our earlier examples than the already-implemented system from Project 1.) However, it is more important that your security analysis give realistic advice in the context of the project's scenario than that you adhere precisely to any specific process.

Even though it would probably work well as a structure for your final report to present the design of the system first, and then your security analysis of it, you should probably go back and forth between design and analysis when you start working on the project. You don't have the perfect design to start: start with a design that might be imperfect, and consider what threats it might be vulnerable to. If you see a threat that could be prevented or mitigated by changing the design, make the change to the design and then update the security analysis to match. One of the key benefits of doing this kind of design and security analysis before building the real system is that it is faster to modify your design.

Use of outside research. Online proctoring systems similar to the hypothetical one in this project already exist in the real world: you might even have already used some of them. You are allowed to use your knowledge of real products, or to research about them, as you think about what features your system for this project should have. However, just because a feature has been included in a real system doesn't mean it's a good idea, and this project is primarily about your security judgement, not researching existing systems. If you learn about a feature that a real system has, you should only include it in your design for this project if you agree with and can articulate in your own words what security (or other) purpose it fulfills. Be sure to apply appropriate skepticism to the marketing materials for commercial systems, which are likely to focus on advantages and leave out disadvantages. Also, if you got a design idea from a real system, you should acknowledge it.

Formatting your report. Your sole submission for this project will be written report, in a similar format as the initial submission for Project 1. Your report should be 4-5 pages long, formatted for US-standard "Letter" paper (8.5 by 11 inches) with one-inch margins. The main text of your report should use a Times, Times Roman, or Computer Modern Roman font, 10 points high, and double spaced. (By comparison, these instructions use single-spaced 12 point Computer Modern Roman on letter paper with one-inch margins, so your document should take up the same area of the page, but should have a smaller font with more space between the lines.) The expectation of 4-5 pages refers to the text of you report. Your report should probably also include some figures, but you should put them at the end, after the 4-5 pages of main text, and they will likely take up more pages. Your report should be labeled with your name and UMN email address.

Writing is part of the purpose of this assignment and about half of what you will be graded on, so be sure to allow time for quality writing, including revising, checking spelling and grammar, and so on. You should write in a relatively formal style like a report you were writing in business, but your priority should be explaining your technical points clearly.

To acknowledge any external sources you used, cite them using end notes. At the place in the main text of the report where you used external information, write a number for the external source in square brackets. At the end of the report (not counted in the page limit), list the information about each external source with enough specificity that a reader

could find and consult that source if they wanted to. You can use your own judgment as to the order in which the end notes are numbered (e.g., in the order they appear in the text, or alphabetical, etc.) and the precise format of the information about the source. These instructions contain one such end note as an example.

References

1. `https://floridacollegeaccess.org/news/study-shows-1-in-4-college-students-unreliable-internet-makes-coursework-difficult/`