Watching the watchers: bias and vulnerability in remote proctoring software

Ben Burgess, *Princeton University;* Avi Ginsberg, *Georgetown Law;*
Edward W. Felten, *Princeton University;* Shaanan Cohney, *University of Melbourne*

https://www.usenix.org/conference/usenixsecurity22/presentation/burgess

This paper is included in the Proceedings of the
31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Watching the watchers: bias and vulnerability in remote proctoring software

Ben Burgess
Princeton University

Avi Ginsberg
Georgetown Law

Edward W. Felten
Princeton University

Shaanan Cohney
University of Melbourne

Abstract

Educators are rapidly switching to remote proctoring and examination software for their testing needs, both due to the COVID-19 pandemic and the expanding virtualization of the education sector. State boards are increasingly utilizing these software packages for high stakes legal and medical licensing exams. Three key concerns arise with the use of these complex programs: exam integrity, exam procedural fairness, and exam-taker security and privacy.

We conduct the first technical analysis of each of these concerns through a case study of four primary proctoring suites used in U.S. law school and state attorney licensing exams. We reverse engineer these proctoring suites and find that despite promises of high-security, all their anti-cheating measures can be trivially bypassed and can pose significant user security risks.

We evaluate current facial recognition classifiers alongside the classifier used by Examplify, the legal exam proctoring suite with the largest market share, to ascertain their accuracy and determine whether faces with certain skin tones are more readily flagged for cheating. Finally, we offer recommendations to improve the integrity and fairness of the remotely proctored exam experience.

1 Introduction

The rapid adoption of proctoring suites (software for monitoring students while they take exams remotely) is a phenomenon precipitated by the pandemic [27]. Subsequently, we find high stakes exams being administered remotely. Recent media coverage provides anecdotal evidence that these proctoring suites introduce security concerns, reduce procedural fairness, and pose risk to exam integrity [28, 32]. In this work, we aim to investigate the anecdotal claims and systematize the study of these packages. We evaluate the proctoring software on its technical merits using traditional reverse engineering techniques and perform a fairness analysis.

Historically, computerized test taking benefited from in-person proctors and institution controlled hardware. Proctors are unavailable in the remote test-taking setting, prompting institutions to turn to proctoring suites. Proctoring suites attempt to mitigate the increased risks of cheating associated with the remote environment and student hardware by installing pervasive, highly privileged services on students' computers. Such software attempts to reduce academic misconduct during exams by limiting access to online resources and system functions, such as the ability to paste in pre-written materials. Considerable privacy concerns arise since a student's laptop is not generally a single use device that only contains class related material. A student using their own hardware faces the risk of having their personal information and other sensitive information in their possession misused or leaked by the proctoring software.

Initially, the remote proctoring software was marketed and designed for tertiary institutions, however, the software has recently been adopted for medicine and law licensing exams. Inherent societal costs to illegitimately passing students are magnified in both professions, where an inept lawyer can put an individual's liberty at stake and an incompetent physician can cause significant trauma to patients. The time and monetary burden of professional education and licensing places extreme pressure on students and this, along with the benefits of passing, increases the extent to which students may be willing to risk cheating. Maintaining confidence that degrees earned and licenses obtained ensure a minimum degree of competency and knowledge is imperative. Equally important is the confidence that no individual who merits entrance into a profession has been blocked due to false cheating allegations. Applied research into whether remote exam proctoring puts either of these interests in jeopardy is lacking in current literature and merits attention from the security and privacy community.

We conduct the first systematic, technical analysis of the remote proctoring ecosystem of law school and state bar exam boards. By limiting the scope of our investigation to the single regulated profession of law, we can map out how the software is used across the entire profession, increasing confidence in our proposed recommendations. We do not, however, have reason to suspect that applying our methodology to other examination settings would yield substantially different results.

Through public data aggregation and survey, four exam proctoring suites utilized in 93% of U.S. law schools and 100% of remote state bar exams are identified: Examplify, ILG Exam360, Exam4, and Electronic Blue Book. Reverse engineering of these four proctoring suites reveals vulnerabilities of varying complexity that compromise the purportedly secure testing environments. We evaluate suites in the context of three potential adversaries: a law student; a law student with computer science experience; and an experienced reverse engineer. We discuss vulnerabilities identified with each. The proctoring suites we analyzed installed highly privileged system services with full access to user activities. Highlighting the privacy trade-off of the software packages, we find that system logs created *before an exam begins* are nonetheless transmitted to the vendor's servers during the exam.

We determine that Examplify implements a facial recognition classifier to authenticate a student against a pre-existing photograph prior to starting an exam. The classifier is then re-run repeatedly during the exam, depending on the settings selected by the exam administrator. At each interval, the classifier attempts to determine whether the student initially authenticated is the student who is presently taking the exam. We extract the name of the facial recognition system Examplify is using, 'face-api.js', and note they employ the pre-trained models that are publicly available on the face-api's GitHub. To evaluate whether one is more accurate than the other, we test these models against current off-the-shelf, state-of-the-art (SOTA) classifiers using several different datasets. We find significant accuracy concerns across the board.

We then evaluate the classifiers across subjects of different races and find concerning variability. We investigate proctoring suite terms of service and user interfaces to shed light on whether a user is giving informed consent to all types of monitoring the software performs.

We offer privacy protecting recommendations for remote exam proctoring administrators and students. We conclude with suggestions for vendors on ways to improve exam integrity while lessening the student privacy impact.

Contributions

- We survey the top 180 law schools and all U.S. state bar associations to determine their remote proctoring practices and release the dataset (Section 3.3).
- We reverse engineer four exam proctoring suites and identify the security the proctoring suite provides the institution and its student privacy ramifications (Section 4).
- We release a tool for automatically and rapidly extracting key security and privacy properties of exam suite software (Section 4.4).
- We perform a detailed evaluation of racial biases in the facial recognition model used in the software with the dominant market-share for remote proctoring in law schools (Section 5).

While we acknowledge that the attack techniques we use are generally well known, their application to this high-stakes context merits a comprehensive analysis.

Research Ethics & Limitations

While our survey involved contacting law schools and state bar associations to determine what platforms they use and how they use them, our work was exempt from IRB review as we did not collect data pertaining to individuals.

Our analysis of facial recognition systems used a data set containing images of incarcerated individuals who were not given an opportunity by the original researcher to consent to its use for research. We consulted with an applied ethicist independent of the project and determined that while we are unable to rectify the consent issue to fully uphold the principle of autonomy, use of these images is nonetheless appropriate, as our work fulfills the principle of beneficence in the following manner:

- Our work aims to aid marginalized groups and dis-empowered individuals by evaluating the privacy/bias concerns of software that powerful organizations require students to use.
- Our work does not cause additional harm to the individuals in our dataset, beyond perpetuating its use in academic literature.

Thus, while we are unable to uphold the principle of autonomy to the greatest extent, we believe our research is nonetheless appropriate.

Our analysis of facial recognition systems focuses on racial biases in algorithms, despite evidence that system performance is more closely tied to skin-tone with race as a proxy. However, as our very large reference data sets are racially coded rather than coded by skin tone, a more sophisticated distinguishing analysis was outside our scope.

We restrict ourselves to the law and regulated professions sub-sector of education. Centering our work on a limited set of products allows for timely research, considering current societal needs, and more comprehensive coverage of our chosen area.

We intentionally refrain from evaluating server side components and functionality of the software packages to avoid the accompanying legal and ethical concerns. We conducted responsible disclosure to the proctoring suite vendors, making them aware of discovered vulnerabilities along with potential remediation steps. We believe this mitigates any potential harm disclosing these vulnerabilities may have on exam integrity.

2 Related Work

Several previous studies [22, 38] have discussed different threat models remote proctoring solutions face. They have recommended security features to mitigate these new vulnerabilities such as improved authentication measures or 360-degree cameras. Slusky [34] extended this by investigating the security feature claims of 20 different exam proctoring suites, noting their strengths and weaknesses against various threat models. Teclehaimanot, ET AL. [36] studied eight John Madison University professors and determined that a significant number of their students seemingly gained an advantage on remotely proctored exams despite the use of a proctoring suite. Cohney, ET AL. [9] performed a multidisciplinary analysis of the risks faced by institutions as they rapidly adopted EdTech.

A few studies attempted to quantify how a student's perceived stress and performance varies between remote and in person exam environments. Karim, ET AL. [20] studied 582 subjects taking a cognitive test in each environment and found similar scores between the two groups but a higher perceived stress level in subjects in the remote setting. Teclehaimanot, ET AL. [36] performed a meta-analysis of test scores recorded in proctored and unproctored environments, finding a 0.20 standard deviation between the two sets.

Teclehaimanot, ET AL. [36] surveyed eight experts from different universities with experience using remote exam proctoring and found the perceived trust of the vendor and the security of the offering to be the primary factors influencing their solution adoption decision. The recent work of Balash, ET AL. [4] presented a detailed user-study of student responses to remote proctoring software in light of the pandemic. Barrett [5] discussed the risks of adopting remote proctoring software in terms of bias, test-taker anxiety, and student privacy.

Previous studies have demonstrated biases in different components of facial recognition systems [23, 39]. Singh, ET AL. [33] created targeted attacks to cause false positives by the classifier. While we do not investigate these attacks in the context of Examplify's classifier, we anticipate no reason they would not be applicable. Nagpal, ET AL. [25] evaluated several different machine learning classifiers using the Multi-PIE and Morph-II datasets and found significant racial biases. We closely structure our methodology for detecting biases in Examplify's classifier to this work. The National Institute of Standards and Technology (NIST) conducted the Face Recognition Vendor Test (FRVT) to quantify face recognition accuracy at a very large scale [17], providing methodological expertise that guides our work.

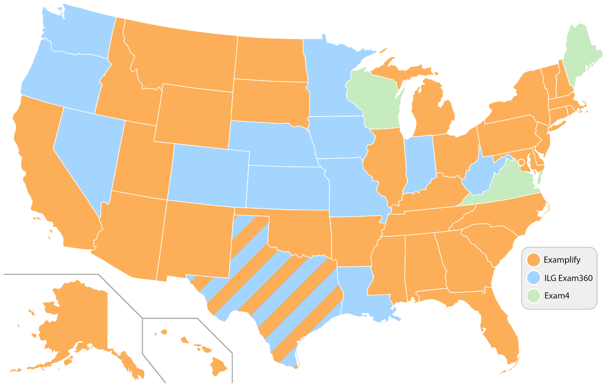

Figure 1: The adoption of remote exam proctoring suites by state bar exam associations across the United States.

Perhaps more importantly, teachers and students have documented their concerns in increasing number since the start of the pandemic. Students have performed small scale experiments testing various proctoring suites [14, 19] and teachers have voiced misgivings [35] about the use of facial recognition.

3 Preliminaries

Computerized exams, coupled with the need for remote testing, have prompted increased reliance on proctoring software, some inclusive of facial recognition classifiers. Subsequently, concerns over privacy, security, racial bias, and fairness have come to the forefront.

3.1 Exam Software

Computerized exam software generally consists of a user interface that displays multiple choice questions with answers and/or text boxes for student answer entry. This is coupled with a series of cheating detection and deterrence tools tailored to meet the assumed threat model. The general assumption: students cheat by searching the internet; opening documents/programs on their computers; or consulting a device, person, or printed material during the exam. To this end, exam software generally block access to the internet and non-approved applications on the user's machine; perform audio recordings to detect verbal communication with another person; and run facial recognition to ensure the appropriate individual takes the entire exam and does so without looking away to consult another information source.

Facial recognition is a contentious aspect of exam software given general concerns with the accuracy of such models across different skin-tones and face structures. As a result, individuals from certain groups may be flagged more frequently for potential cheating—a classic case of 'disparate impact'.

As a highly regulated industry, law pays particular attention to ethical and professional standards. When a law student cheats within school or during a licensing exam, there is a high likelihood that the accrediting organization will prohibit them (potentially permanently) from practice. We infer that these high standards within the profession affect decisions made regarding exam administration and the choice of what remote proctoring software platform to use, if any.

3.2 Legal Education

In modeling the threats to law school and bar exams, we consider the unique structure of testing and nature of the exams themselves. Law school course grades depend primarily on a single, tightly curved final exam typically worth at least 85% of the course grade. Law school grades and thus, exams, are closely tied to job prospects (more-so than many other fields) [6] and bar exam scores are a determining factor in lawyer licensing. These factors, combined with the high-debt burden associated with U.S. legal education, place extreme pressure on students. Strong motivation exists for unscrupulous students to cheat, even by means of paying outside individuals for materials, technical bypasses, or cheating tools. Law exams often take the format of a story riddled with legal issues that a student must identify and analyze. Exam answers consisting of several pages of written text are common so that cheating by copying another student's answer would be painfully obvious. Student cheating by obtaining exam content in advance to pre-write answers or using messenger apps during the exam to consult friends, comprise the likely and more difficult to detect forms of dishonesty.

COVID-19 has caused bar associations and schools to rapidly adopt an at home remote testing model, forcing administrators to look for additional assurances that cheating attempts will be detected.

3.3 Software Usage Survey

To determine the targets for our technical work, we survey software usage across bar associations and law schools. We identify the remote proctoring software used by the top 180 law schools by scraping public facing websites and making private inquiries to law school administrations. Table 1 depicts our survey results. We repeat this process for every state bar association's licensing exam and illustrate the results in Figure 1. Our data set of the individual school and state bar remote proctoring suite adoption choices is available at github.com/WWP22/ProctoringSuiteAdoption. We find that Examplify, ILG Exam360, Exam 4, and Electronic Blue Book (EBB) comprise the four primary exam software suites used by over 93% of law schools and 100% of bar exam associations. We select these for analysis in the remainder of this work.

Exam Proctoring Suite	Schools	Percentage
Examplify	99	55%
Exam4	52	29%
Electronic Blue Book	13	7.2%
Canvas	4	2.2%
MyLaw	3	1.7%
ILG Exam360	1	0.56%
Other	5	4.5%

Table 1: Examplify leads in market share followed by Exam4 and Electronic Blue Book. Two schools did not respond to inquiries and one closed down.

4 Cheating

The uncontrolled setting and hardware of the remote environment makes for a broad threat model. To evaluate the security provided by these exam proctoring suites, we present a reverse engineering methodology and propose three theoretical but realistic adversaries to a remotely proctored law exam. Practical compromises to the security features we identify, along with potential attacks from each of these adversaries, are discussed.

4.1 Methodology

The four exam suites analyzed are not open source, so we reverse engineer the binaries using existing static/dynamic analysis tools. We focus on three questions: (1) Do the exam suites provide the security guarantees they promise? (2) What privacy and security is the user required to give up to achieve these guarantees? and (3) Are the exam integrity checks fair across all examinees?

We isolate suspected critical functions using common reverse engineering methods such as system log inspection and recording of user interface dialogues. We then

Security Feature	**Examplify**	**Exam4**	**EBB**	**ILG Exam360**
Encryption at Rest	AES-256	AES-256	3DES/AES-256	AES-256
Encryption in Transit	HTTPS	HTTP	HTTP	HTTPS
Virtual Machine Detection	Block List	Block List	Block List	Block List
Virtual Device Detection	Block List	N/I	N/I	Block List
Clipboard Management	Integrated	Cleared	Cleared	Integrated
Screenshot Capture	N/I	N/I	N/I	App Window
Process/App Restrictions	Allow List	Block List	Block List	Allow List
Network Access Restrictions	Route Table	Adapter Disable	N/I	Null DNS

Table 2: Summary of security features implemented in the evaluated exam proctoring suites. N/I indicates the feature was not implemented by the exam proctoring suite.

use traditional disassembly and reverse engineering workflows to manually inspect binary areas of interest.

We employed dynamic analysis to identify and describe functionality that was not apparent through the static analysis, disabling anti-debugging functionality when present in the software.

To evaluate the integrity of exams in transit, we use standard network interposition techniques to determine whether the connection to retrieve the exam is over an encryption TLS connection or in plaintext. If the connection is over TLS, we evaluate the program's response to being served: (1) a valid certificate that has an incorrect common name; and (2) a self-signed certificate that is not recognized by any certificate authority but bears the correct common name. We also attempt to force a downgrade by blocking access to the port before the handshake can occur and then watching for further plaintext retries on a separate port.

For readability, we defer exam fairness methodology to Section 5 due to its increased depth.

4.2 Threat Model for Exam Proctoring

We informally model three adversaries likely to interfere with the fair and secure operation of a remotely proctored law exam. We emphasize that, in reality, *many* students may be modelled as a more sophisticated adversary than their background would suggest by hiring help. Attack budgets ranging from a few to several thousands of dollars are feasible, given the hundreds-of-thousands of dollars spent on law school. We define three adversaries:

Law Student An individual able to adjust basic system and file settings, configure simple hardware devices, and use known passwords. No experience in reverse engineering software, programming, modifying binary settings, or extracting encryption keys is assumed.

Law Student with CS Background An individual with significant programming experience, familiarity with basic system administration, and the knowledge to extract keys from the binary. No ability to modify any portion of the binary and no extensive experience reverse engineering software is assumed.

Experienced Reverse Engineer An individual with all the prior capabilities with additional experience analyzing and rebuilding binaries, disassembling software, and applying patches. Familiarity with advanced system administration and the ability to: adjust any setting, configure hardware devices, modify drivers, use custom encryption tools, and extract encryption keys from and modify the control flow of the binary is assumed. While we expect the number of students with this background is low, such individuals may sell their services or cracked versions of software.

4.3 Reverse Engineering Findings

This subsection details our findings regarding how proctoring suites operate. Exam proctoring suites use a few distinct components to ensure exam integrity is maintained: **computer monitoring**, **exam content protection**, and **identity verification and authentication**. We categorize our findings based on these components for the following four proctoring suites: Examplify, Exam 4, Electronic Blue Book, and ILG Exam360.

4.3.1 Computer Monitoring

The computer monitoring components in an exam suite aim to prevent a student from accessing unauthorized resources during the exam and may even restrict the student from beginning the exam if certain parameters are not met. A summary of the security related features used by each suite is provided in Table 2, along with a summary of privacy related features in Table 3. Each computer monitoring component of exam integrity is outlined, followed by each exam suite's implementation and vulnerability to the aforementioned adversaries. We validate the attacks referenced in this section against the proctoring suite in

Privacy Related Feature	**Examplify**	**Exam4**	**EBB**	**ILG Exam360**
Initial Identity Verification	Automated	N/I	N/I	Human
Continuous Identity Check	Automated	N/I	N/I	Human
System Service	Always Running	App Running	App Running	App Running
Device Identifiers	App List/OS/Hardware	N/I	N/I	OS/Hardware

Table 3: Privacy related features implemented in the evaluated exam proctoring suites. Exam4 and Electronic Blue Book do not implement any image based identity verification functionality or collect any notable device identifiers.

a simulated test environment to ensure the attacks will succeed in a real deployment.

Virtual Machine Detection Virtual machine software allows a guest operating system to be run inside the primary environment, bypassing any monitoring the proctoring suite could hope to achieve. To prevent this, most suites feature virtual machine detection to detect and prevent attempts to run the software inside a virtual container. The most common implementation of this is a simple check of the computer's CPU manufacturer to see if the string matches known virtual machine software vendors. An additional check of CPU temperature for constant values or known presets can be run since a virtual CPU does not often pass through the real value from the actual CPU.

All the exam suites we examined implement a virtual machine check by comparing the CPU vendor field against a list of known virtual machine vendors. Examplify extends this by retrieving the CPU temperature and flagging a device if it reports a CPU temperature of 100C, as this is the typical default value virtual machine vendors use. Electronic Blue Book also checks the computer, hard drive, network adapter, and bios vendor information to see if any fields contain the string 'virtual'. If a virtual machine is detected, they log the attempt and prompt the user to run the software outside a virtual machine. Table 4 provides a summary of the popular virtual machine software blocked by each proctoring suite.

The CPU vendor check implemented by all the exam suites can be easily bypassed using common virtual machine software since this field is generally configurable. The CPU temperature check Examplify conducts is defeated by configuring the virtual machine to pass through the CPU temperature of the actual computer's CPU vendor or a random string not on the block list the exam proctoring suites are using. A law student with a CS background, capable of installing a virtual machine and configuring the CPU vendor, could readily complete this attack.

Virtual Device Detection Virtual webcam and microphone devices can allow adversaries to take exams outside an appropriate physical context by replaying a prerecorded file or piping a connection to a separate device. Adversaries could bypass exam suite identity verification by returning video of themselves in another location while someone takes the exam for them or by returning a prerecorded video of them taking the exam. Exam proctoring suites attempt to mitigate this threat by checking the device vendor and bus location against a list of flagged vendors. If one of these blocked vendors is detected, the software will either flag the exam for further review or prevent the student from beginning to take it.

Examplify and ILG Exam360 detect virtual webcams and microphones by retrieving the operating system's device list and comparing it to known virtual device vendors. As Exam4 and Electronic Blue Book do not use the computer's webcam or microphone, they do not implement a check. Table 5 provides an overview of the popular virtual webcam/microphone vendors on the blocked list.

An adversary using a virtual webcam or microphone can easily evade detection by installing a virtual driver not on the known vendor list. More skillfully, an adversary could rename the driver of a blocked virtual device since the signing key of the driver is not checked by the exam proctoring suite. All these attacks are well within the capabilities of a student with a CS background. A student able to create a custom virtual device that masqueraded as a legitimate driver could redistribute this to other students.

Clipboard Management Students copying pre-written text into an exam is a major concern, especially in the field of law, where exam essay answers may require lengthy summaries of law and/or analysis that can be prepared before the exam. Exam proctoring suites attempt to prevent this by either clearing the clipboard before the exam or logging its contents for later analysis. During the exam, the clipboard generally can be used inside the proctoring suite. However, copying from outside apps is prohibited or alternately logged using a similar method.

The exam proctoring suites implement clipboard protection by calling the system clear function before the exam begins. The content is not captured before the clear operation by any of the suites. Examplify and ILG Exam360 implement a custom restricted clipboard for use inside the test environment that limits what can be copied.

Virtual Machine	Examplify	Exam4	EBB	ILG Exam360
VirtualBox	✓	✓	✓	✓
VMWare Workstation	✓	✓	✓	✓
VMWare Fusion	✓			
Parallels	✓	✓		✓
Hyper-V	✓			
QEMU	✓			

Table 4: Popular virtual machine software blocked by the evaluated exam proctoring suites. VirtualBox and VMWare Workstation were blocked by all of the proctoring suites.

These protections do not preclude the use of an external hardware connected clipboard such as a KVM or a built-in keyboard macro that allows note storage. Such hardware devices frequently present themselves as standard human interface devices, which do not require any additional drivers. We do not investigate the Mac version of the exam proctoring suites. However, colloquial evidence suggests that the iCloud clipboard sharing might bypass protections by loading information from the phone's clipboard into the Mac clipboard. Exam proctoring suites could attempt to protect against cutting and pasting clipboard content by fingerprinting the input rate of a student's keystrokes, but we did not find an implementation of this in any of the proctoring suites we analyzed. Purchasing a hardware device capable of maintaining an external clipboard is an attack most law students could perform.

Screenshot Capture Exam proctoring suites may offer screenshots of the student's screen during the exam to allow a proctor to retroactively review the exam session to determine if unauthorized resources were accessed on the computer. These screenshots are normally captured using a highly privileged system level service, which leads to potential privacy issues when an exam is not in progress.

ILG Exam360 is the only exam suite we analyzed that provides screenshot captures of the student's computer during an exam. The screenshot is captured by calling their highly privileged system service using a Javascript call that uploads the screenshot to Exam360.

Bypassing this protection directly would require someone who had extensive reverse engineering experience since integrity checks need to be removed from the Exam360 binary before a modified Javascript module can be loaded. The modified Javascript module could then be coded to either upload a set of pre-taken screenshots or only snapshot a specific area of the screen. A less complex approach to bypassing this protection would be to use a secondary display/device. However, the viability of this approach may be limited by other protections such as requiring test takers to show their work area before the exam.

Process/Application Restrictions Process restrictions are normally used to limit what applications a student can access during an exam. These are generally implemented using a process allow list that contains processes specifically allowed by the exam, in addition to critical processes the operating system needs to maintain the computer's function. A weaker implementation involves using process block lists that prevent certain processes, such as web browser activation, from being started. Both approaches are implemented using the exam proctoring suite's highly privileged system service, which starts a watchdog service that forcibly kills unauthorized processes.

Examplify and ILG Exam360 compare the processes currently running on the system to a list of processes they and the exam itself allow, continuously monitoring for and forcibly killing any processes that are running but not included on the list. Exam4 and Electronic Blue Book have a list of disallowed services that they kill upon exam commencement.

To subvert either restriction, a student with CS background could recompile a piece of open-source software to report a different process name. As an example, the Chromium web browser could be recompiled to report as 'explorer.exe', which is allowed by every testing suite we looked at since it is a critical user interface component for Windows based systems. For suites using block lists, most law students would be capable of finding a process not on the list through trial and error. Proctoring suites generally allow a student to test the exam taking restrictions before beginning the exam, providing a low risk way for students to experiment with circumventing the software.

Network Interception Closed book exams require the software to limit a student's ability to search for information on the internet. Approaches to block internet access that we found included: dropping internet traffic, inserting an active man in the middle to capture traffic, or redirecting the traffic to the vendor's servers. The simplest approach is dropping the traffic using a routing rule.

Virtual Webcam/Microphone	**Examplify**	**Exam4**	**EBB**	**ILG Exam360**
ManyCam	✓	N/I	N/I	✓
YouCam	✓	N/I	N/I	✓
MyCam	✓	N/I	N/I	✓
Logitech Capture		N/I	N/I	
OBS Studio		N/I	N/I	✓

Table 5: Popular virtual webcams and microphones detected or blocked by Examplify or ILG Exam360. Exam4 and Electronic Blue Book do not use the webcam or microphone during exams so no specific security features to target these devices are implemented.

Examplify restricts network traffic by inserting a null route into the default operating system routing table. Exam4 disables the network adapter during the examination. ILG Exam360 inserts a null DNS entry into the network configuration causing domain name resolution failure. Electronic Blue Book does not implement any network restrictions other than blocking access to common websites through their process block list. None of the implementations inspected capture browser traffic or redirect it to a site controlled by the suite vendor.

The network restrictions for both Examplify and Exam4 would likely require an experienced reverse engineer to bypass, as it requires significant modifications to the binary to override. These proctoring suites routinely check the setting throughout the exam, which prevents easier bypasses from working. ILG Exam360's protection can be bypassed by accessing external resources using the IP address or manually setting the DNS server in the browser. This is a relatively simple attack within the capabilities of any of our theoretical attackers.

Summary Most vulnerabilities can be easily exploited by a law student with a computer science background, raising serious concerns regarding the overall security of these proctoring suites. The vulnerabilities and the adversary model capable of exploiting them are summarized in Table 6. Recapping, proctoring suites attempt to maintain exam integrity through the computer monitoring discussed above and through exam content protection and identity verification/authentication, to be discussed in the next two sections.

4.3.2 Exam Content Protection

Exams are often downloaded to student computers before the actual exam begins. Security during transit between proctoring suite vendor servers and students is paramount since traffic can be easily intercepted using off the shelf solutions. Exam protection during the download and while sitting on the student's computer is vital to prevent early or unauthorized access. We detail our findings for each exam suite for both encryption at rest and in transit below.

Encryption In Transit Examplify and ILG Exam360 use transport layer security (TLS) for all their connections. The certificate chain, expiration date, and common name are correctly verified, mitigating active man in the middle attacks. The connection is never downgraded to plaintext HTTP, even if the software is unable to successfully complete the handshake. Examplify includes their own certificate store inside the software to prevent potentially using a modified system certificate store. Exam4 and Electronic Blue Book allow the individual institution to select their transport layer security settings. Electronic Blue Book allows each institution to configure whether to use TLS or not, and the configurations are available publicly. We found several that failed to enable TLS. Exam4 similarly features per-school configuration, but the configurations are not public. However, the school we obtained the Exam4 binary from did not have TLS enabled.

Examplify and ILG Exam360's implementation would require an experienced reverse engineer to modify the binary to remove the certificate verification. This would allow a man in the middle to be used to capture information in transit. Exam4 and Electronic Blue Book could potentially be exploited by a law student with CS experience depending on the configuration the institution used. If TLS was not enabled, a man in the middle proxy could be run without any significant custom configuration.

Encryption At Rest Examplify, ILG Exam360, and Exam4 use AES-256 for encrypting exams at rest on student computers. ILG Exam360 and Exam4 use SHA1 to derive the AES key from a password. Examplify uses 10,000 iterations of the PBKDF2 function to derive the exam password. The exam manifest, which contains the main exam password salt, exam information, allowed resources, and monitoring settings, is encrypted separately using a static key stored in the Examplify binary. Electronic Blue Book allows the institution to choose between 3DES and AES for exam encryption and uses 1,000 iterations of SHA1 by default for password derivation, but the institution can configure the iteration count. The pass-

	Adversary		
Vulnerable Component	Law Student	Law Student with CS Background	Exp. Reverse Engineer
Virtual Machine Detection		✓	✓
Virtual Device Detection		✓	✓
Clipboard Management	✓	✓	✓
Screenshot Capture			✓
Process/App Restrictions		✓	✓
Network Interception	✓	✓	✓

Table 6: Proctoring suite security component vulnerability to different adversary types.

word salt is statically stored in the Electronic Blue Book binary and is set to 'Kosher'.

The encryption of the exams at rest would be vulnerable to exhaustive key search attacks depending on the complexity of the password being used. The implementation used by Examplify and Electronic Blue Book can be searched moderately quickly due to the static salt and low password derivation function iterations being used. We believe these attacks could be easily conducted by a law student with CS experience.

4.3.3 Identity Verification and Authentication

Exam suites all implement some form of user authentication to ensure the test taker matches the individual to be assessed.

Logins Exam4, ILG Exam360, and Electronic Blue Book implement standard single factor logins. Examplify implements a similar single factor login. OAuth is not supported by any of these solutions. As a result, institutions cannot easily add more extensive identity verification measures such as two factor verification.

General Interaction Fingerprinting General interaction fingerprinting analyzes the pattern of a student's key strokes and mouse movements against the class average for anomalies and, if present, flags the exam for human proctor review. This poses the risk of potentially unfairly flagging students with disabilities or those who legitimately deviate from the class average pattern. While none of the suites we analyzed use this, it is used by Proctorio, a common proctoring suite outside our scope.

Facial Recognition As an analog to students showing ID upon entering an exam hall, some exam proctoring suites employ facial recognition for identity verification. The distance of the facial feature vectors of a student's image are compared against those of the student's trusted reference image and if below a certain threshold, the student is considered verified.

ILG Exam360 offers facial recognition, but also employs a remote human verification method before exam initiation. A webcam connects a student with a human proctor who conducts the final verification. In the end, the process resembles that of an exam hall. Examplify's verification implementation relies on an automated facial recognition classifier. Our research quantifies bias introduced by Examplify's process, which we detail in Section 5.3. Exam4 and Electronic Blue Book do not offer facial recognition.

Summary Demonstrating how suite security features can be easily bypassed by a law student with computer science background sheds light on the potential negative impact remote exam proctoring can have on exam integrity. A discussion of the privacy concerns generated by suite security features and their potential for introducing bias into the exam taking process is also necessary to understand the full ramifications of remote proctoring.

4.3.4 Student Privacy Concerns

Two major privacy questions arise when evaluating the impact of remote exam proctoring software : (1) Is the user appropriately informed of the information being collected upon engaging with the remote exam software? and (2) Does the potential for pervasive monitoring after the student is no longer actively taking an exam exist? To this end, we develop an analysis tool to assist other researchers in identifying remote exam software privacy issues.

Informed Consent

An examinee cannot provide meaningful consent to the activities performed by the exam proctoring software if they are not informed of the specific data being collected or the surveillance mechanisms utilized. Examinees are prohibited from reverse engineering the software to discover such information and attempts to glean this information by reading privacy policies, end user license agreements, or similar documents will be met with vague and sometimes conflicting verbiage. As an example, ExamSoft's Examplify privacy policy notes, "in order to secure the exam taker's device, ExamSoft must access and, in some instances, modify device system files." This broad statement provides no substantive limitation on what the Examplify software may do. Other privacy policies contain conflicting statements about the software's activities. For example, the Extegrity Exam4 privacy policy states, "Exam4 does not access any data on the laptop other than the data created by the act of taking an exam" and "Exam4 monitors activity the student engages in while taking an exam and creates an encrypted log report for evidentiary purposes." It is not possible for Exam4 to monitor activity the student engages in if it does not access any data other than that created by the act of taking an exam. These types of statements thwart a student's ability to meaningfully consent. Furthermore, even if an examinee was fully aware of the software privacy implications, meaningful consent is still lacking given the examinee's lack of meaningful alternatives. Faced with accepting and using the software as-is or refraining from taking the exam, most law students would opt for the former, as the latter coincides with an inability to become a licensed lawyer. Such a choice is not a choice. These policies and agreements fail to meet conventional ethical standards for consent [3].

Post Exam Monitoring Examplify installs a highly privileged system service that is constantly running on the computer even if Examplify is not open. Currently running applications on a user's computer are logged to a debugging file that is uploaded periodically once the Examplify application is open. The service also regularly reaches out to the Examplify server to check for and install updates for the service or the binary. Exam4 and ILG Exam360 also implement a system service but stop it when the exam is terminated gracefully. Electronic Blue Book directly hooks into the Windows system service with their binary to provide their monitoring features, guaranteeing no additional background monitoring is being performed once the binary is closed.

4.4 Automating Privacy Impact Analysis

We created an analysis tool based on the RADARE2 and Ghidra frameworks [8] to simplify the reverse engineering process for researchers who want to quickly analyze the privacy impact of other exam proctoring solutions. We release the tool publicly at github.com/WWP22/ProctoringSuiteAnalysisTool. This paper's approach, using traditional tools like IDA, works well for in depth studies, but is not well suited for providing a quick summary of an exam proctoring suite's privacy impact. Our tool requires a user to simply run the Python script on the binary they want to analyze and a high-level overview of the application will be provided.

4.4.1 Design

We design the reverse engineering tool using a methodology similar to one an experienced reverse engineer would likely follow when analyzing a piece of unfamiliar software. We prioritize using techniques we believe generalize well to other pieces of software, such as tracing the control flow of device drivers and system libraries. These would need to be called by any potential program hoping to successfully trigger the device. This improves the ability of our software to be widely applicable to any proctoring suite.

The analysis tool first loads all the shared object files the binary uses and then performs auto analysis using RADARE2's built-in analysis suite. This attempts to locate the segments and functions to generate a control flow graph. From this control flow graph, we extract cross-references to lines in any part of the code, which allows us to more easily establish where certain data elements are being used.

We can detect privacy-sensitive calls such as calls to a microphone, webcam, or video driver by fingerprinting common vendor and system library names. We also attempt to extract information about the security features the exam proctoring suite implements, including whether it detects virtual machines, uses a secure connection to reach the back-end server, and encrypts on-disk content. If on-disk encryption is found, we display the cipher suite being used and attempt to extract the encryption key and initialization vector. We do so by searching for keys of the correct bit length in a user-definable window around any data references found in the encryption function. For a more complete analysis, the tool can be run with the live memory option, which initializes the binary, attaches the GNU Project debugger (GDB), runs to a user-defined breakpoint, and then performs the analysis. This allows for a more complete analysis of libraries and code segments that are stored encrypted at rest or loaded from a remote endpoint. A user can view a summary of the binary's security and privacy properties or opt for a more detailed analysis featuring control flow graphs and decompilations with Ghidra of relevant code segments.

4.4.2 Evaluation

We evaluate the automated analysis tool on the four suites to determine whether it accurately identifies relevant security and privacy information. The tool is evaluated without using the live memory analysis option. We believe this would be the most common configuration of the tool due to the relatively large performance and memory overhead of searching the entire live memory space multiple times on consumer hardware.

We find the tool able to correctly identify camera and microphone usage in all cases aside from a false posi-

					Race Annotation				
Dataset	**Source**	**# of subjects**	**# of faces**	**In-the-wild**	White	Asian	Black	Hispanic	Indian
MORPH-II	Government Data	14K	55K		✓	✓	✓	✓	
Multi-PIE	Study Capture	750K	337		✓	✓	✓		
LFWA+	LFW	6K	13K	✓	✓	✓	✓		✓
VGGFace2	Image Search	9K	3M	✓	✓[†]	✓[†]	✓[†]		✓[†]

Table 7: Summary of datasets used for the facial recognition accuracy section.

tive triggered when analyzing Electronic Bluebook. This false positive is due to the inclusion of a large English dictionary in the binary that incorrectly triggers one of the vendor searches we run. The relevant function control flow graph and decompilation is presented to the user, which would allow the user to trivially identify it as irrelevant and subsequently disregard.

The tool performs similarly well with virtual machine detection, insecure connections, and on-disk encryption, with results mirroring the results we obtained in our manual analysis. The encryption key Examplify uses is successfully extracted and presented to the user along with a few false positives. We write a simple Python script to test the encryption keys and initialization vectors the tool outputs and can successfully decrypt Examplify's libraries in under one minute.

Our tool was designed to generalize well to other software suites, however, we were not able to obtain other exam proctoring software to validate this. We leave this for future work.

Of the features present in the proctoring packages, facial recognition capabilities are particularly amenable to tool-based analysis. A rigorous evaluation of such would require a purpose built tool. This is beyond our scope and therefore left for future work. We do, however, develop a detailed methodology that future research can follow for analyzing algorithmic bias manually.

5 Automated Identity Verification

Traditionally, human verification through simple recognition of a student and identification card checks have been used to ensure exam integrity. While these checks could still be conducted in the remote exam suite setting, reduced staffing costs and an increased capacity for the number of students that can be verified provide large incentives for vendors to move to a fully automated facial recognition system. When facial recognition classifiers are used to determine who can take an exam or whether a student is flagged for cheating, it is critical that the system is accurate and fair.

Using datasets we believe accurately reflect the real-world images these systems would be operating on, we evaluate the overall accuracy of current state-of-the-art facial recognition systems alongside Examplify's facial recognition classifier, 'face-api.js'. The classifiers we select for comparison against 'face-api.js' are based on their accuracy using Labeled Faces in the Wild (LFW) [31], the current dataset used most prominently in the literature for benchmarking the fairness of facial recognition algorithms [21]. This results in a selection containing Facenet [30], VGG-Face [26], OpenFace [1], and ArcFace [12, 18, 13, 11].

Given that the expertise of remote proctoring firms is outside AI/ML and that in the current business environment such firms are unlikely to develop and train their own models (an assumption borne out by our analysis of the leading market products), it is reasonable to select pre-trained, off-the-shelf models for comparison. Finally, we conduct an analysis of the error rates based on the subject's race to assess the fairness of the classifier across subjects from different races. We release our image selections, data processing code, statistical analysis results and classifier performance data at github.com/WWP22/AIVerify.

5.1 Accuracy

Based on Examplify's identity verification system, we separate the facial recognition steps an exam proctoring suite would need to perform into two steps: (1) the initial verification against a student's identification photo to bootstrap their identity and (2) the continuous verification to ensure the student who verified their identity initially continues to take the entire exam.

Dataset Selection We select MORPH-II [29], Multi-PIE [16], LFWA+ [24] and VGGFace2 [7] as our datasets for evaluation of facial recognition classifier performance, using images very similar to ones likely encountered dur-

[†] VMER uses more specific race annotations than the current federal government standards for collecting and presenting data on race. We group these more specific annotations as follows using the federal standards and the NIST FRVT as a guideline: Caucasian Latin→White; African American→Black; East Asian→Asian; and Asian Indian→Indian

	FNMR		FMR	
Classifier	# of subjects	# of comparisons	# of subjects	# of comparisons
MORPH-II	13K	42K	14K	681K
Multi-PIE	264	81K	264	172K
LFWA+	1.7K	7.4K	5.7K	287K
VGGFace2	9.1K	457K	9.1K	457K

Table 8: Summary of the sub-sampled datasets used for the false non match rate (FNMR) and false match rate (FMR) initial verification analysis.

ing real-world use. A full description of the datasets can be found in Table 7.

Metric Selection The false non match rate (FNMR) and the false match rate (FMR) are the primary proxies for demonstrating the effect various facial recognition classifiers have on the user. The FNMR demonstrates the rate at which the facial recognition system would fail to verify a student and the FMR demonstrates the rate at which the system would falsely verify a different person as the student. Ideally, we would extract these directly from the exam proctoring suite, but this was out of our reach as we were unable to obtain the required accounts. Instead, we base our approach off of the method Examplify implements to detect cheating, which relies on the shortest distance (the Euclidean distance) between feature vectors on a reference face and the current face. We use a selection distance of 0.60 to mirror Examplify's cut-off distance for cheating. We also use a Z-test to calculate the p-value between the different feature distance data-points to determine whether we can draw statistically significant conclusions (customarily $p < 0.05$) about classifier performance. These metrics are used in other key facial recognition accuracy studies such as the NIST FRVT [17], allowing our results to be easily comparable.

5.1.1 Initial Verification

Initial identity verification in a remote exam setting relies on comparing a known image of the subject (likely a school ID or driver's license) to a capture of the subject trying to take the exam. This can present background, lighting, and age progression challenges for the facial recognition classifier.

Image Selection We select images from each of the above datasets to create sub-sampled datasets for our FNMR and FMR evaluation. We create our FNMR datasets by selecting the earliest capture of each subject in the dataset as our reference image. We use up to 50 images from subsequent captures of the same subject to compare against. The extended length of time between subsequent captures in the Morph-II, LFWA+, and VGGFace2 datasets make them useful for the initial identity verification comparison since institutions commonly use the student's driver's license as the reference image, which has an average photo refresh time of 5.7 years [37]. For the Multi-PIE dataset we select subjects who attended multiple sessions, using each subject's first attended session as the reference image and up to three of their subsequent session images as comparison images. Use of these Multi-PIE captures provides insight on whether more recent reference images improve the performance of the facial recognition classifier. We create the datasets we use for evaluating the FMR by selecting the earliest subject capture as the reference image and 50 earliest captures of other subjects as comparison images. Table 8 summarizes these sub-sampled datasets.

Classifier Performance High FNMRs occur across all the datasets evaluated for both the SOTA classifiers and 'face-api.js'. When run on datasets that are in-the-wild versus those with more constrained capture conditions, significantly elevated FNMRs occur across all the classifiers except VGG-Face. We see a large difference in the FNMRs between the Morph-II and Multi-PIE datasets for the 'face-api.js' classifier, which suggests that the time between the reference image and the comparison image may not be a major factor in classifier performance. Overall, we see an inverse relationship between the FNMR and the FMR such that an algorithm less likely to fail at identifying a correct student would be more likely to verify an imposter (hired test-taker). We summarize these findings in Table 9.

We select the SOTA algorithm that achieved the best overall FNMR, VGG-Face, and compare the mean feature distances to 'face-api.js'. We note 'face-api.js' produces better average feature distances compared to VGG-Face over Morph-II with $mean_d = 0.257$ versus $mean_d = 0.533$ ($p = 1.44 * 10^{-133}$). We see VGG-Face out perform 'face-api.js' on all of the other datasets with a maximum $mean_d = 0.234$ compared to a maximum $mean_d = 0.474$ ($p = 6.96 * 10^{-84}$). For a goal of minimizing FNMR, VGG-Face's performance suggests that it would likely be the more reliable algorithm.

Analysis The high FNMRs evidenced make any of the evaluated classifiers inappropriate for use in an auto-

	MORPH-II		Multi-PIE		LFWA+		VGGFace2	
Classifier	FNMR	FMR	FNMR	FMR	FNMR	FMR	FNMR	FMR
Facenet	7.5%	14%	9.2%	50%	14%	2.9%	17%	2.0%
VGG-Face	4.1%	78%	1.9%	83%	2.8%	68%	3.4%	42%
OpenFace	5.7%	72%	8.6%	77%	12%	62%	11%	63%
ArcFace	12%	2.2%	11%	48%	24%	0.54%	23%	1.2%
FaceAPI	0.66%	21%	9.2%	19%	22%	0.69%	14%	0.42%

Table 9: FNMR and FMR averages across various datasets using both state of the art algorithms and 'face-api.js'.

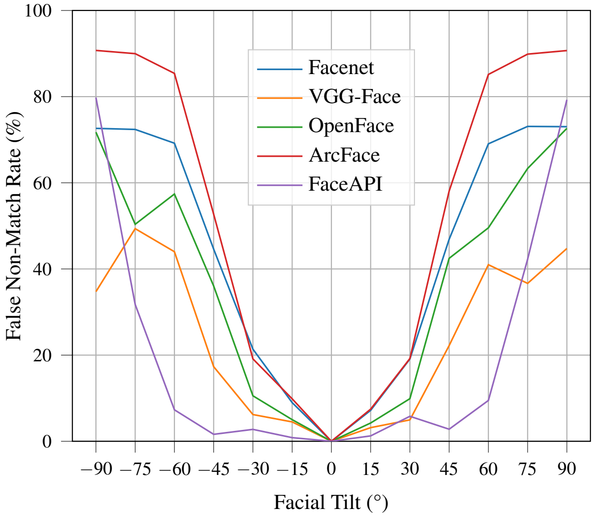

Figure 2: False Non Match Rate (FNMR) of subjects at various facial rotations ranging from -90° to 90°.

mated facial recognition setting due to the extreme cost to a student unable to be verified. Exam proctoring vendors can attempt to reduce the error rate by selecting classifiers with the lowest FNMR, however, these classifiers all present much higher FMRs, suggesting they provide little more than security theater. Using more recent trusted reference images had minimal impact on the FNMR when compared to using images captured over longer time periods. This minimizes an institution's ability to mitigate the high FNMR by collecting more recent reference images.

5.1.2 Continuous Verification

Continuous identity verification works to ensure the initially verified student takes the entire exam and is not replaced by another person. The student's recent reference image, verified during initial verification, is compared to subsequent captures taken silently at random intervals during the exam. These unprompted captures create challenges for the facial recognition classifier since the student's facial rotation, expression, and lighting may all vary from the prompted reference image taken in a controlled fashion.

Image Selection We create subsets from the Multi-PIE dataset for evaluating FNMRs based on varying facial rotations, facial expressions, and lighting. Each dataset contains 151K, 104K, and 131K samples respectively, with 337 subjects each.

Classifier Performance Analyzing classifier performance based on the FNMR for variable facial rotations, we find the SOTA classifiers and 'face-api.js' are roughly equivalent up to a 30 degree rotation but diverge thereafter, with 'face-api.js' maintaining a relatively low FNMR up to 60 degrees. The SOTA classifiers, except VGG-Face, fail to verify most subjects once the rotation reaches 60 degrees, while 'face-api.js' still verifies over a 75-degree rotation. Figure 2 shows the relationship between facial rotation and classifier performance. There is minimal variation from the average FNMR discussed in Section 5.1.1 when the subject changes their facial expression. Classifier performance falls off sharply as the lighting conditions differ more significantly from the reference image. 'Face-api.js' underperforms VGG-Face when evaluating images taken under varying lighting conditions but outperforms all the other SOTA classifiers. Appendix A provides a full description of the FNMR performance on these datasets.

Analyzing classifier performance based on feature distance, we use the SOTA algorithm with the best FNMR, VGG-Face, and compare it to 'face-api.js'. We find VGG-Face achieves better feature distances at +/- 15, 30, 75, and 90 degrees with an average $mean_d = 0.376$ compared to an average $mean_d = 0.475$ ($p = 4.26 * 10^{-05}$). 'Face-api.js' achieves better average feature distances at +/- 45 and 60 degrees with an average $mean_d = 0.451$ compared to an average $mean_d = 0.490$ for VGG-Face ($p = 5.54 * 10^{-20}$). The variable performance between VGG-Face and 'face-api.js' as the facial rotation changes precludes us making a reliable recommendation on which one would perform better in a testing scenario.

Analysis The classifiers exhibit increased FNMRs once tasked with comparing images in challenging lighting or facial rotation conditions. Students taking an exam in

	MORPH-II				LFWA+				VGGFace2			
Classifier	Black	White	Asian	Hispanic	Black	White	Asian	Indian	Black	White	Asian	Indian
Facenet	6.8%	9.7%	15%	10%	15%	15%	9.5%	14%	26%	17%	17%	15%
VGG-Face	3.6%	5.2%	11%	7.4%	1.4%	2.9%	2.7%	0.58%	1.9%	3.6%	3.1%	3.4%
OpenFace	6.0%	4.6%	9.5%	3.8%	13%	12%	10%	18%	18%	10%	14%	7.1%
ArcFace	11%	13%	23%	11%	30%	24%	15%	27%	34%	21%	27%	19%
FaceAPI	0.57%	1.1%	0.0%	0.1%	23%	22%	18%	21%	20%	13%	14%	11%

Table 10: FNMR averages across various datasets using both state of the art algorithms and 'face-api.js'.

good faith over long exam periods have images that reasonably exhibit lighting and facial position variation, especially since captures are taken without notice. The FNMRs suggest that a significant number of students would fail to be verified correctly even in the less extreme conditions and almost all the students would fail to be verified correctly in the more extreme cases. The variance in lighting seen was on the more extreme side but not outside the realm of real-world exam environments, where exams can start in daylight and end after dark.

5.2 Fairness

When forms of identity verification are used as part of examination procedure, expending every effort to minimize bias is critical to promoting exam fairness.

Dataset Selection We select MORPH-II and LFWA+ as they provide labels of a subject's race in the original dataset and VGGFace2 as labels are available through additional studies such as VGG-Face2 Mivia Ethnicity Recognition (VMER) [15]. Multi-PIE is not included in this section due to the limited number of subjects in some of the races we analyze.

Metric Selection The FNMR metric, which potentially flags a student for cheating, is utilized in this section since it reflects the most negative impact on the student.

Classifier Performance On the Morph-II dataset, all classifiers except OpenFace flag 'White' subjects at a slightly higher rate than 'Black' subjects. Across races, 'Asian' subjects are flagged at higher rates in all classifiers except 'face-api.js'.

On the LFWA+ dataset, higher FNMR values for 'Black' subjects versus 'White' subjects occur for all the classifiers except VGG-Face and Facenet. 'Black' subjects have higher FNMRs compared to 'Asian' subjects across all classifiers except VGG-Face. They also have higher FNMRs compared to 'Indian' subjects across all classifiers except OpenFace. Compared to other races, 'Asian' subjects have lower FNMR values across all classifiers run on LFWA+ except VGG-Face and Facenet.

On the VGGFace2 dataset, all classifiers except VGG-Face yield higher FNMR values for subjects labelled as 'Black' versus 'White', 'Asian', or 'Indian'. 'Indian' subjects have the lowest FNMRs across all races using all classifiers except VGG-Face on the VGGFace2 dataset. While lower FNMRs are seen on 'Asian' subjects compared to 'White' subjects with LFWA+, elevated FNMRs occur on the 'Asian' group compared to the 'White' group using the VGGFace2 dataset except with the Facenet and VGG-Face classifiers. A summary of these results can be seen in Table 10.

We compare the mean feature distance within the 'face-api.js' classifier for subjects from each race to determine if we see the performance vary based on the subject's race. Across the Morph-II dataset, we see minority groups with a reduced average feature distance, $mean_d = 0.321$, compared to the 'White' group, $mean_d = 0.387$ ($p = 2.63 * 10^{-05}$). Across the LFWA+ dataset, we see a reduced mean feature distance for the 'Asian' group, $mean_d = 0.494$, compared to the 'White' group, $mean_d = 0.550$ ($p = 2.40 * 10^{-12}$). We cannot draw conclusions about the performance of the other minority groups compared to the 'White' group ($p = 0.815$). Across the VGGFace2 dataset, we see the 'Black' group exhibit a significantly worse feature distance, $mean_d = 0.499$, compared to the feature distance for the 'White' group, $mean_d = 0.475$ ($p = 3.12 * 10^{-130}$). We draw the opposite conclusion when comparing the other minority groups, $mean_d = 0.449$, to the 'White' group ($p = 4.56 * 10^{-36}$). The difference in the performance when looking at mean feature distance versus FNMR can be attributed to the higher variance on the minority groups for VGGFace2 compared to the 'White' group.

Analysis We see variations in the performance of the different classifiers depending on the race of the subject being analyzed. In the large VGGFace2 and LFWA+ datasets, we see higher FNMRs for subjects labelled as 'Black' versus 'White' except with the VGG-Face classifier. Similar disadvantage for subjects tagged as 'Asian' compared to those tagged as 'White' occurs on the VG-

GFace2 dataset except for with the VGG-Face classifier. This demonstrates the propensity for minority groups to be flagged at higher rates than other groups depending on the capture conditions. Additionally, we see major variance in FNMR between subjects of different races based on the dataset, demonstrating the difficulty of correcting for this in the software using the classifier.

5.3 Evaluating Real-World Implementation

Examplify uses the 'face-api.js' classifier for facial recognition, with a pre-trained publicly available model inside the exam proctoring suite. We see in Section 5.1 that 'face-api.js' was quite inaccurate overall with an average FNMR of 11.47% when evaluated on datasets we believe accurately reflect images it would be running on during a live exam. We also noted statistically significant improvements in performance in VGG-Face over 'face-api.js' when comparing the mean feature distance over our unconstrained datasets. This suggests that Examplify needs to perform further evaluation of their classifier choice. In Examplify's implementation, the initial verification step acts as a gatekeeper for the student to be able to take the exam. The high average FNMR suggests that this would disadvantage numerous students attempting to take their exam. We see similarly concerning performance on the continuous verification task run in the background during an exam if enabled by the institution. Realistic variations in the capture conditions such as face rotation or lighting changes could cause the image to fail to be verified, flagging the student's exam for human review. Depending on how this is presented to the proctor for human review, this may unfairly bias the proctor against the student with a presumption of guilt.

We see variations in the performance of the 'face-api.js' classifiers depending on the race of the subject with some minority groups being disadvantaged in LFWA+ and VGGFace2 datasets. This suggests that the automated facial recognition system may be unfairly disadvantaging certain students based on their race by not allowing them to take their exam or by presenting their exam for human proctor review at a higher rate than other non-minority students in certain cases. Given the variability and bias in the facial recognition steps, we believe a human-based verification model is a fairer approach to ensuring exam integrity.

If an automated classifier is to be used, we recommend training models on a dataset that contains a balanced sampling of subjects from different races versus using pretrained default models. We also recommend evaluating the performance of the classifies on datasets that realistically represent the use case of the system. Classifier performance cannot be accurately assessed just using an overall performance metric without looking at that performance metric across subjects from different races to determine whether the classifier is acting fairly.

We were only able to analyze the local facial recognition component of the Examplify proctoring suite since an instructor account could not be obtained. Further verification steps may be conducted on the server-side to minimize the high FNMR seen on the client-side implementation. Without an Examplify instructor account with facial recognition features enabled, we could not verify whether this further validation is being performed.

6 Discussion

Impact on Marginalized Groups Minorities and other marginalized groups are traditionally underrepresented in the legal profession. Exam software with built-in skin-tone biases creates an invisible barrier. This occurs both in the intuitive case where minorities are flagged for cheating at higher rates and where they are flagged at substantially *lower* rates. Opponents of affirmative action have erroneously argued that minorities are not able to cope with the rigor of law school. By using software that flags minority students substantially less, such opponents will continue to cast doubt on the validity and competence of minority students. Thus, substantial bias in either direction can perpetuate systemic racism. This may harm the chances of minority students trying to enter the legal profession.

Law schools, bar associations, and other educational/licensing institutions must investigate and commission research into inherent bias in the exam software they utilize to prevent discrimination in exam administration.

Fundamental Challenges of Remote Examination Remote exam proctoring suites suffer from fundamental limitations in the threat model they can protect against since they run on untrustworthy student hardware versus exam hall hardware. The student has full administrative access and will always be able to bypass security features and monitoring schemes given enough time. Exam proctoring suite vendors can attempt to increase the time and skill level necessary to compromise the exam by adding complexity to the process through obfuscation and active anti-debugging measures.

To create a truly secure remote exam proctoring suite, a vendor needs to establish a trusted environment on the device that restricts the student's ability to extract or modify part of the exam suite. Intel SGX and other similar trusted execution environments could potentially be employed to ensure a secure environment. However, this would involve a major engineering undertaking and introduce additional usability concerns. Furthermore, Intel has recently announced the deprecation of SGX in consumer chips as a result of ongoing security issues with the technology [10].

Risks of Privileged Software All the software we evaluated required privileged system access. Operating systems increasingly restrict such access as it is a common source of malfeasance. Buggy but well-intentioned code given such access substantially broadens the attack surface of the OS and serves as a glaring target. Experts urge against granting such access even to third-party antivirus software [2]! Compounding the problem, students are likely to be unaware that privileged system services from the proctoring packages do not uninstall automatically and persist after the exam is over [4], putting them at long term risk.

Privacy, Surveillance, and Ethical Concerns Attempting to meet their design goals, platforms engage in sweeping surveillance. Keylogging, screen captures, drive access, process monitoring, and A/V feeds give their software access to personal data stored on the device. Outside this context, these features appear only in malware, highlighting the unusual capabilities of these software suites. Some binaries also include anti-debugging mechanisms in their code, further limiting the ability of student advocates to assess the security and safety of the software.

The context in which students are required to install proctoring software mitigates their ability to meaningfully consent to the substantial impositions on their privacy and security. A veneer of platform legitimacy is conveyed by the institutional backing of the school or testing company. Trust in institutions, Balash, ET AL. [4] found, substantially reduces student willingness to object to remote proctoring. Even if they did, students are not provided with a reasonable alternative.

Recommendations Our strongest recommendation is that where allowable, educators design assessments that minimize the possible advantage to be gained by cheating. Project work, open book essays, or other assessment modes featuring unrestricted resource access yield fewer opportunities to gain unfair advantage.

Where re-imagining assessment is not possible, students should be offered a *meaningful* chance to opt-out of digital testing on their own hardware and should be given the choice of either using provided hardware or paper-copy and live proctoring. Furthermore, the substantial effort schools expend to help students install proctoring software should be matched with equal efforts to help them uninstall it, while advising them of the risks of retaining it.

We recommend against the use of facial recognition systems. Where infeasible, secondary human review should be conducted by multiple diverse individuals to reduce human biases as well. Lastly, if current classifiers are going to see continued use, candid conversation regarding generalized differences in facial features between racial groups needs to be addressed by programmers through dialogue with racially diverse focus groups and accounted for in calibration settings to reduce the incidence of false identification.

Acknowledgements

We thank Paul Ohm (Georgetown) and the University of Pennsylvania Carey Law School staff for their assistance; Simon Coghlan for consultation on the ethics of our dataset usage; and Jelena Mirkovic for the thoughtful feedback.

References

[1] Amos, B., Bartosz, L., and Satyanarayanan, M. OpenFace: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[2] Anthony, S. It might be time to stop using antivirus. Ars Technica, 2017.

[3] Appelbaum, P. S., Lidz, C. W., and Meisel, A. Informed consent: legal theory and clinical practice, 1987.

[4] Balash, D. G., Kim, D., Shaibekova, D., Fainchtein, R. A., Sherr, M., and Aviv, A. J. Examining the examiners: students privacy and security perceptions of online proctoring services. *USENIX Symposium on Usable Privacy and Security*, 2021.

[5] Barrett, L. Rejecting test surveillance in higher education. *SSRN Electronic Journal*, Jan. 2021.

[6] Be prepared: law school doesnt even resemble your college experience.

[7] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. Vggface2: a dataset for recognising faces across pose and age, 2018.

[8] Cheng, E. Binary Analysis and Symbolic Execution with angr. PhD thesis, PhD thesis, The MITRE Corporation, 2016.

[9] Cohney, S., Teixeira, R., Kohlbrenner, A., Narayanan, A., Kshirsagar, M., Shvartzshnaider, Y., and Sanfilippo, M. Virtual classrooms and real harms. *USENIX Symposium on Usable Privacy and Security*, 2021.

[10] Corporation, I. 12th generation intelő core processors datasheet.

[11] Deng, J., Guo, J., Niannan, X., and Zafeiriou, S. Arcface: additive angular margin loss for deep face recognition. In *Cvpr*, 2019.

[12] Deng, J., Guo, J., Yuxiang, Z., Yu, J., Kotsia, I., and Zafeiriou, S. Retinaface: single-stage dense face localisation in the wild. In *arxiv*, 2019.

[13] Deng, J., Roussos, A., Chrysos, G., Ververas, E., Kotsia, I., Shen, J., and Zafeiriou, S. The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking. *Ijcv*, 2018.

[14] Feathers, T. Proctorio is using racist algorithms to detect faces. *Vice*, 2021.

[15] Greco, A., Percannella, G., Vento, M., and Vigilante, V. Benchmarking deep network architectures for ethnicity recognition using a new large face dataset. *Machine Vision and Applications*, 2020.

[16] Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. Multi-pie. *Image and vision computing*, 28(5):807–813, 2010.

[17] Grother, P. J., Grother, P. J., and Ngan, M. *Face recognition vendor test (frvt)*. US Department of Commerce, National Institute of Standards AND Technology, 2014.

[18] Guo, J., Deng, J., Xue, N., and Zafeiriou, S. Stacked dense u-nets with dual transformers for robust face alignment. In *Bmvc*, 2018.

[19] Johnson, E. Hey @proctorio @artfulhacker how do you explain this? *Twitter*, 2020.

[20] Karim, M. N., Kaminsky, S. E., and Behrend, T. S. Cheating, reactions, and performance in remotely proctored testing: an exploratory experimental study. *Journal of Business and Psychology*, 29(4):555–572, 2014.

[21] Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4873–4882, 2016.

[22] Langenfeld, T. Internet-based proctored assessment: security and fairness issues. *Educational Measurement: Issues and Practice*, 39(3):24–27, 2020.

[23] Leslie, D. Understanding bias in facial recognition technologies. *arXiv preprint arXiv:2010.07023*, 2020.

[24] Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[25] Nagpal, S., Singh, M., Singh, R., and Vatsa, M. Deep learning for face recognition: pride or prejudiced? *arXiv preprint arXiv:1904.01219*, 2019.

[26] Parkhi, O., Vedaldi, A., and Zisserman, A. Deep face recognition. university of oxford, 2015.

[27] Raman, R., Sairam, B., Veena, G., Vachharajani, H., and Nedungadi, P. Adoption of online proctored examinations by university students during covid-19. *Education and Information Technologies*:1–20, 2021.

[28] Reed, A. Online bar exams come with face scans, bias concerns. Bloomberg Law, 2020.

[29] Ricanek, K., and Tesafaye, T. Morph: a longitudinal image database of normal adult age-progression. In, volume 2006, 341–345, May 2006.

[30] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: a unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823, 2015.

[31] Serengil, S. I., and Ozpinar, A. Lightface: a hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 23–27. Ieee, 2020.

[32] Singer, N. Online cheating charges upend dartmouth medical school. NYTimes, 2021.

[33] Singh, R., Agarwal, A., Singh, M., Nagpal, S., and Vatsa, M. On the robustness of face recognition algorithms against attacks and bias. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34 of number 09, 13583–13589, 2020.

[34] Slusky, L. Cybersecurity of online proctoring systems. *Journal of International Technology and Information Management*, 29(1):56–83, 2020.

[35] Swauger, S. *MIT Technology Review*, 2020.

[36] Teclehaimanot, B., You, J., Franz, D. R., Xiao, M., and Hochberg, S. A. Ensuring academic integrity in online courses: a case analysis in three testing environments. *The Quarterly Review of Distance Education*, 12(1):47–52, 2018.

[37] Tefft, B. C. Driver license renewal policies and fatal crash involvement rates of older drivers, united states, 1986–2011. *Injury epidemiology*, 1(1):1–11, 2014.

[38] Turani, A. A., Alkhateeb, J. H., and Alsewari, A. A. Students online exam proctoring: a case study using 360 degree security cameras. In *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, 1–5. Ieee, 2020.

[39] Wu, W., Protopapas, P., Yang, Z., and Michalatos, P. Gender classification and bias mitigation in facial images. In *12th ACM Conference on Web Science*, 106–114, 2020.

A Continuous Verification Performance

The average FNMRs of the five facial recognition classifiers across sub-sampled datasets we created using the Multi-PIE dataset are presented to illustrate classifier performance in a continuous verification setting.

A.1 Average FNMR

	Multi-PIE Dataset			
Classifier	Session	Facial Rotation	Lighting	Facial Expression
Facenet	9.2%	48%	20%	9.4%
VGG-Face	1.9%	26%	2.3%	2.1%
OpenFace	8.6%	39%	11%	7.7%
ArcFace	11%	58%	14%	11%
FaceAPI	9.2%	22%	8.7%	9.6%

This table summarizes the average FNMRs of the five classifiers when tested on ‘Session’, ‘Facial Rotation’, ‘Lighting’, and ‘Facial Expression’ sub-sampled datasets. Reduced average FNMRs occur with the ‘Session’, ‘Lighting’, and ‘Facial Expression’ datasets compared to the ‘Facial Rotation’ dataset. This is likely because the average number of facial features being occluded by any of these variations is less than when the subject’s face is rotated away from being centered with the camera.

A.2 Facial Rotation

	Rotation											
Classifier	-90	-75	-60	-45	-30	-15	15	30	45	60	75	90
Facenet	73%	72%	69%	45%	21%	9.0%	7.2%	19%	47%	69%	73%	73%
VGG-Face	35%	49%	44%	17%	6.2%	4.5%	3.2%	5.0%	22%	41%	37%	45%
OpenFace	72%	50%	57%	36%	11%	5.0%	4.2%	9.9%	42%	50%	63%	73%
ArcFace	91%	90%	85%	53%	19%	9.9%	7.4%	19%	58%	85%	90%	91%
FaceAPI	80%	32%	7.3%	1.6%	2.8%	0.84%	1.2%	5.8%	2.8%	9.5%	42%	79%

This table depicts the average FNMRs of the five classifiers when tested on facial rotations of +/- 15, 30, 45, 60, 75, and 90 degrees. The average FNMR increases as the facial rotation increases, most likely due to more facial features being hidden in the image.

A.3 Session & Lighting

	Session Gap			Lighting			
Classifier	1	2	3	00	04	12	16
Facenet	10%	9.2%	8.0%	34%	15%	17%	13%
VGG-Face	2.1%	1.8%	1.8%	1.1%	2.7%	2.4%	3.0%
OpenFace	9.3%	8.4%	7.7%	13%	12%	12%	8.4%
ArcFace	12%	10%	9.2%	15%	15%	15%	13%
FaceAPI	10%	8.9%	8.3%	17%	7.2%	8.2%	2.9%

This table depicts the average FNMRs of the five classifiers across images taken in different capture sessions and lighting conditions. The session gap refers to the number of sessions between the reference image and the comparison image. The expectation of the average FNMR increasing as the session gap increased was not evident with any of the classifiers. This suggests that the time between captures is less important than other factors. The average FNMR for Facenet, OpenFace, and ‘face-api.js’ degraded under one of the worst lighting conditions in the Multi-PIE dataset, ‘00’, which corresponds to a 90 degree lighting angle. Similar degradation when testing against the mirrored condition, ‘16’, did not occur suggesting that performance in bad lighting conditions is very variable across most of the classifiers.