INDIAN INSTITUTE OF MANAGEMENT KOZHIKODE

Working Paper

**IIMK/WPS/472/ITS/2021/09**

August 2021

# SAHEB: Securing the Access to Hardware for an Exam Browser

**Lija Chandran TV[1]**
**Mohammed Shahid Abdulla[2]**

[1] Independent Researcher, Information Technology and Systems, Email: lijachandran@gmail.com

[2] Associate Professor, Information Systems Area, Indian Institute of Management, Kozhikode, IIMK Campus PO, Kunnamangalam, Kozhikode, Kerala 673 570, India; Email: shahid@iimk.ac.in, Phone Number (+91) 495 – 2809254

# SAHEB: Securing the Access to Hardware for an Exam Browser

**Lija Chandran TV**
lijachandran@gmail.com
*Independent Researcher, Information Technology and Systems*
**Mohammed Shahid Abdulla**
shahid@iimk.ac.in
*Associate Professor, Information Systems Area, IIM Kozhikode*

## Abstract

We propose here a secure online exam solution to overcome the vulnerabilities of the existing online exam applications. In particular, we profile applications like *Mettl* or *Safe Exam Browser* which sanitize the examinee's computer by placing it into 'kiosk mode' of the MS Windows operating system. At launch, these applications also scan the system for a list of prohibited processes and kill these before the start of the exam. Yet, security claimed by these applications can be compromised with the help of applications or processes not in the prohibited list, including via a simple name-change. It is also possible to have an open source screen-sharing application and an audio chat application which can, in combination, be used to breach the system. We also consider the possibility that some important application, e.g. required for executive student's work activities, is in the prohibited list, terminating which would cause loss of data or business. Our solution format SAHEB addresses these issues. SAHEB, which we evaluate for feasibility in its various features, consists of an open source hypervisor which will modify access to the peripherals of the examinee's computer. A lightweight browser-only Linux distribution will further deploy the exam's material.

## 1. Introduction

With the outbreak of COVID19, there is a massive shift in the work methods around the globe. Most organizations, including educational institutions, started to adopt the remote mode of work. Pandemic restrictions meant that all regular classes had to be conducted online. Similarly, the traditional pen and paper examination (*offline* mode) also had to be replaced by remote mode (*online* and distributed mode) of conducting exams. This was often in the wake of much controversy which included canceling of many nation-wide exams. While most meetings and classes have shifted to the remote mode using the applications like *Zoom*, *Webex* and *Microsoft Teams*, the assessments have also switched to online platforms. Applications like *Mettl*, *Wheebox*, *Safe Exam Browser* and *Moodle* are examples of platforms that can be used to conduct proctored (invigilated, via camera) or non-proctored online examinations. Though these existing platforms do a good job at conducting assessments remotely, these applications are also prone to security breaches. The paraphernalia for a breach, e.g. SW toolchain, can be made available at scale and hence breaches may neither be restricted to small groups, nor entirely detectable.

When applications like *Mettl* and *Safe Exam Browser* (SEB) are launched, these ‘lock down’ the computer[1][2] in order to turn it into a secure workstation. In the Windows operating system this mode of execution is called *kiosk* mode[3], whilst in MacOS this is the *assessment* mode[4]. Putting a computer into the lockdown will result in the forced termination of the currently running applications, and also the blocking of function keys on the keyboard, incl. short-cut keys, besides any access to external drives or the facility to share the screen via an application like TeamViewer, or any screen-share programs. Such a mode creates risk of examinee’s work-related programs also needing to shut down, a situation likely to be common (yet detrimental) to executive participants in class.

There may also be situations where the devices like smartphone, laptop, tablet and desktop are shared: used for both educational and office purposes (e.g. of the parent or sibling), or where students are interns with certain organizations yet registered for a course. Thus there can be critical office or work-related applications or processes running on the computer and terminating these applications while launching the online exam applications might lead to loss of data, or result in undesirable business consequences.

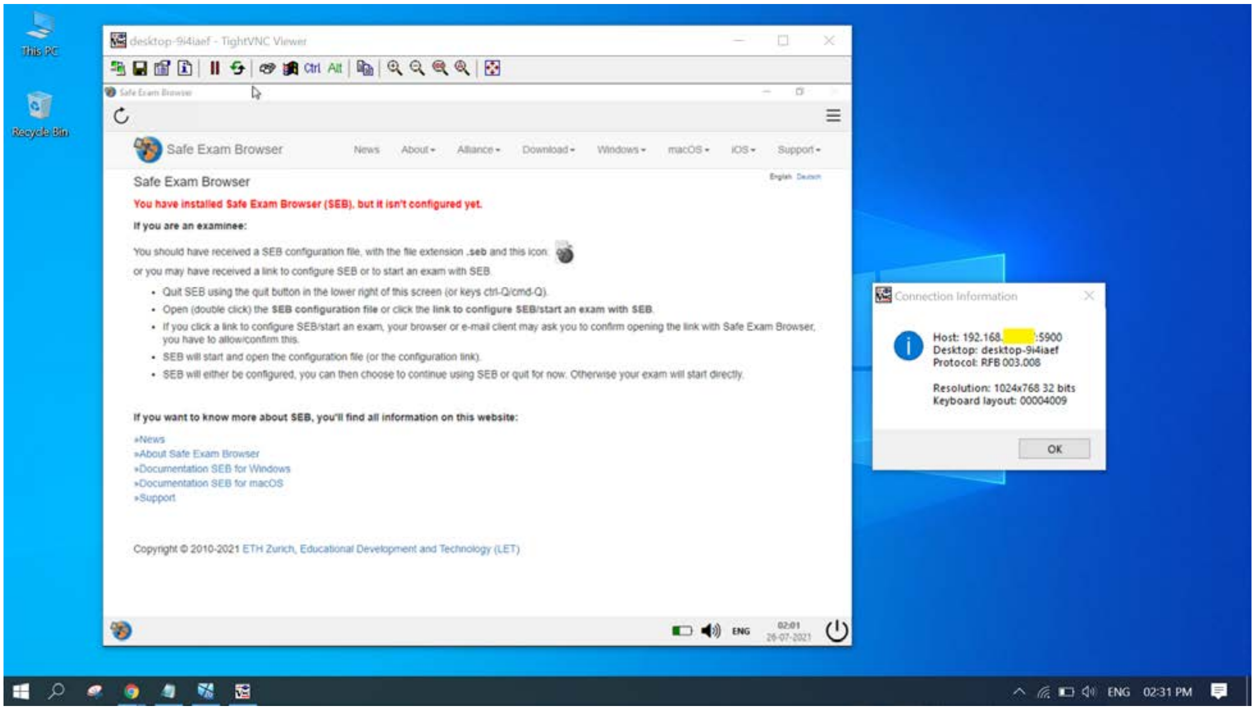
There also exist motivations for students to secure high marks in select courses of their programs, to be considered directly for interviews with hiring organizations, or more recently even for long term visa in developed countries[5]. If cheating on an exam is relatively risk-free with easy-to-understand tools required, students might try to ensure a high score, as the exams are conducted on their personal devices and in their personal premises.

In the next section we describe the breach of the commonly used SEB application, to establish that the examinees can cheat the online exam system easily without being noticed by the proctor - an invigilator who is watching the student via the computer’s camera. Next, we propose a solution which would run semi-independently of the host computer using virtualization: this virtual machine is also granted access - using existing techniques - to features such as prevention of screen capture or use of peripherals. This is so that the assessment can be conducted on a computer that can be considered sanitized.

## 2. Vectors for compromise of SEB

SEB runs in kiosk mode of MS Windows OS, a facility provided in the host computer's OS for such purposes. There is a configurable list of applications which are prohibited while the SEB is running. Many of the instant chat and screen sharing applications are added in this list by default. These applications will be shut down upon the launch of SEB. There is also a provision to white-list any applications. Such applications will then be shown on the taskbar of SEB and can run in the background, with the understanding that certain examinations require SW such as calculators or spreadsheets. Yet, the difficulty is that applications which are in neither list are not checked upon the launch of SEB: with these applications continuing to run in the system. The examinee can choose to run a remote desktop sharing application which is not in the blacklist and not known to the proctors or their organization when configuring SEB, e.g. TightVNC. Once this is done, another person can access the desktop of the examinee remotely, and also mark the correct answer in an MCQ (or type answer in words) once granted access to the system.

Figure 1 below establishes, via a screenshot, that a computer system running SEB can be accessed remotely using TightVNC Viewer.



desktop-9i4iaef - TightVNC Viewer

Safe Exam Browser

You have installed Safe Exam Browser (SEB), but it isn't configured yet.

**If you are an examinee:**

You should have received a SEB configuration file, with the file extension **.seb** and this icon:

or you may have received a link to configure SEB or to start an exam with SEB.

- Quit SEB using the quit button in the lower right of this screen (or keys ctrl-Q/cmd-Q).
- Open (double click) the **SEB configuration file** or click the **link to configure SEB/start an exam with SEB**.
- If you click a link to configure SEB/start an exam, your browser or e-mail client may ask you to confirm opening the link with Safe Exam Browser, you have to allow/confirm this.
- SEB will start and open the configuration file (or the configuration link).
- SEB will either be configured, you can then choose to continue using SEB or quit for now. Otherwise your exam will start directly.

**If you want to know more about SEB, you'll find all information on this website:**

»News
»About Safe Exam Browser
»Documentation SEB for Windows
»Documentation SEB for macOS
»Support

Copyright © 2010-2021 ETH Zurich, Educational Development and Technology (LET)

Connection Information

Host: 192.168.[redacted]:5900
Desktop: desktop-9i4iaef
Protocol: RFB 003.008

Resolution: 1024x768 32 bits
Keyboard layout: 00004009

*Figure: 1*

Other modifications to compromise include: online calling applications like Google Meet which can be opened in an alternative browser (that is not specifically blocked by name). Similarly, it is possible to have computer programs that conduct instant chat in audio mode - with the default being that an incoming message on a particular channel will be read out loudly via the computer's speakers. Thus information may not be per se visible behind the compulsory full-screen that SEB adopts during an exam, but can be gained via audio inputs. It is also possible that a computer's peripherals are modified such that MCQ answers eg. A-D are translated into morse code[6] and the same will be received at the examinee's computer as a sequence of beep sounds or as a sequence of vibrations via a suitably modified peripheral connected to audio-out. Compromising the security of the online exam thus seems feasible with the help of such applications/peripherals.

## 3. Screen-capture prevention in different Operating Systems

This section describes techniques used in different operating systems to prevent screen capture. This screen capture must be necessarily disabled in order to protect the privacy and confidentiality of the question material being displayed on the screen of the personal devices. As described earlier, security of the online examinations can be primarily breached with the help of screen sharing applications. Hence, it is important to keep the contents of online exam applications from being captured by the screen recording applications. A notable caveat here is the use of other devices in candidate's private premises, such as high-resolution pan-tilt-zoom CCTV cameras to capture screen status and to relay this to an answerer outside the room. Such situations are not guarded against.

### 3.1. MS Windows

All desktop applications in the MS Windows OS can impose restrictions on capturing and displaying the content of the application window. This can be achieved by using a programming interface called *SetWindowDisplayAffinity*[7] with the use of two parameters: *WDA_MONITOR* and *WDA_EXCLUDEFROMCAPTURE,* the window content of the application will be displayed only on the current monitor with the window appearing sans content (or not appearing at all) in the screen capture output. In this way any application running locally on the desktop can prevent itself from being captured by screen recording/sharing applications.

An addition to such features is the behaviour of a cloud service from Microsoft called Windows Virtual Desktop. As recently announced[8] , a feature called Screen Capture Protection exists even in virtualized desktops which may be used for remote work by employees. These Virtual Machines stored in the cloud are mirror images of a Windows Desktop, presumably available in a physical sense at the workplace. The screen capture protection feature prevents the entire content - whatever the application - of this remote desktop being captured by screen share applications on the host computer. The portion of the screen where the remote desktop is presently running will be shown as a black screen in the screen-share output. This outcome is achieved by setting a registry key entry[9]. Thus screen-capture by an entire combination of applications can be prevented this way.

### 3.2. Android OS

The apps on Android platform need screen buffer capture permissions to capture the screen content. There have been reports of remote access security vulnerabilities in Android phones where the attackers could take complete control of devices[10]. Silent capture of the screen by apps has been prevented from Android 10 onwards[11], and this is done by not granting the screen reading permissions for the apps without specific user consent. Note the difference from 3.1 where in MS Windows such permissions are granted by default.

## 4. Browser-only Linux Distributions

Our solution SAHEB proposes to deploy a virtual machine with a sanitized lightweight operating system which is customized for the conduct of MCQ-based online exams. This operating system will be shared as an ISO file or OVF (Open Virtualization Format) file, so that it can be deployed as a virtual machine on the candidate’s laptop or desktop. As the Linux operating system is open source, it has in the past been modified and used by several organizations and individuals for specific purposes, including lightweight or mini- and micro- deployments for small hardware. Ubuntu, Centos, Linux Mint, Fedora, OpenSUSE and Arch Linux are examples of Linux distributions. For SAHEB, the Linux OS will be packaged with a single application, i.e. a browser preconfigured with a single URL - that of the MCQ site (e.g. Moodle learning management system). Also, the host OS must be configurable for disabling screen-share permission for any application.

Linux distributions can be modified in such a way that an ISO can be used only for browsing, configurable further to always access only a particular link. The following flavours of Linux OS are examples of browsing-only distributions:

- ➔ *Porteus Kiosk*[12] - Has only a web browser in locked down mode. Users are not allowed to change the settings, download or install other software. After booting up it automatically opens Firefox or Google Chrome browser with configured homepage.

- ➔ *JustBrowsing*[13] - This flavour can be booted from a live CD, USB, hard drive and also available as an OVA (Open Virtualization Archive) file. It has a choice of browsers e.g. Google Chrome, Firefox and a few useful applications bundled with it like calculator, text editor, and a clock.

- ➔ *Webconverger*[14] - Linux based operating system to access a broader range of web applications privately and securely. It runs the Firefox browser in a kiosk mode.

These OS above are suitable for our solution as the ISO files are a few MBs, and unlikely to be a large download even in combination with the hypervisors recommended in Section 5 below, thus these can be shared and downloaded easily among examinees. A further conjecture of ours is that lightweight Linus OS as guest OS in a virtual machine does not add substantially to the computational load on the host computer.

## 5. Open Source Hypervisor on host computer

Hosted or Type-2 hypervisors[15] can be installed on top of the OS (termed the Host OS) in our desktops/laptops. Thus, Virtual machines can be created on our desktops and laptops with the help of these hypervisors, by installing the Linux OS flavours explained in Section 4. Such a setting enables us to have access to another operating system within confines of our laptop or desktop (i.e. examinees resources) without either having to install in entirety or require in advance that the examinee boot into another partition in the system. Note that the latter procedure has the disadvantage of killing any other work-related processes that the examinee may be having running.

Since hypervisors would be required to be set with configuration parameters by the examining organization to A. ensure full-screen size, B. turn off screen share in Host OS etc. - an open-source or configurable hypervisor would be needed. There are many hypervisors available in the market like VMware Fusion, VMware Workstation Player, Oracle VM VirtualBox, QEMU and Parallels Desktop. Of these, Oracle VM VirtualBox and QEMU are examples of open source hosted hypervisors.

In addition, VirtualBox provides APIs for the examining organization to get real-time details of the virtual machine's display e.g. window height, width and screen resolution of the guest OS's screen[16]. For applications of proctoring, there is also an API to attach the webcam of the host machine to the VM[17].

## 6. SAHEB: Proposed Solution

As explained in the previous sections of this paper, breaching the security of extant online exam software is feasible. Our proposal SAHEB tries to resolve all these possible compromise vectors. The solution consists of the following components:

- An open source hypervisor
- A lightweight browser-only Linux OS

The hypervisor is configured in such a way that it will always be running in a full screen mode. This can be implemented by using the API provided by Microsoft to specify the preferred launch window size for the applications[18]. The screen capture prevention mechanism mentioned in section 2.1 would help to prevent capturing the content on the windows of this hypervisor. The hypervisor can also be configured to block access to all the host machine's peripherals to stop information being output via audio, vibration etc.

The combination of configured hypervisor + browser-only Linux OS can be distributed to the examinees in the form of an ISO file. The examinee has to setup for a virtual machine using the hypervisor and then install or live boot from the ISO. An alternative is to provide this as an OVF (Open Virtualization Format) file which can be directly deployed as a virtual machine on the hypervisor. The VM will now be running in a fullscreen mode and the screen size will be polled at regular intervals to make sure that the VM is not minimized or resized. The student will not be able to access any other resources on the VM beyond the cursory applications provided. The browser would also be pre-configured to have one tab with a fixed URL in order to access only the exam portal.

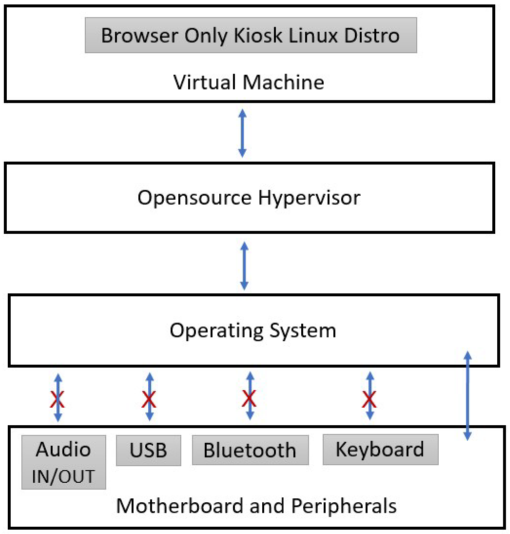


*Figure: 2*

SAHEB also requires the Host OS environment where the hypervisor is running to be sanitized inorder to prevent malpractices during the exam. In order to make sure that the host machine where the hypervisor is being installed is secure, the modes by which external peripherals can be accessed should be disabled by the configuration file of the hypervisor during the exam. Thus, when the hypervisor application is launched on the host OS, the hypervisor will set some parameters as below in host OS inorder to prevent the access to following peripherals/services which can be used for suspect actions:

- ➔ USB - The value for the registry key entry corresponding to the USB storage device should be set to disable[19].
- ➔ Bluetooth - Turn off Bluetooth service using powershell[20]. Applications can be written to use Bluetooth to signal, via vibration or audio, the answer to a question.
- ➔ Audio device - MS Windows provides Core Audio APIs[21] to control the volume level or mute the volume of any audio endpoint devices like speakers, headphones and microphones.
- ➔ Keyboard - The key press event can be captured and alerted to the examinee that the keyboard should not be used[22][23].

Keyboard disable will apply to MCQ exams such that options A-D be selected using a mouse only. SAHEB arranges to ensure that the computer from which the online exam portal is accessed is sanitised. However, the malpractices committed with the following mass-production devices will not be preventable by SAHEB. We expand on our caveat in an earlier section by adding two more scenarios where this could happen:

1. The presence of a camera (pan-tilt-zoom) device in the room which can record the screen of the laptop or desktop.
2. Presence of a camera device on candidate's outfit, e.g. a shirt buttonhole bodycam.
3. Use of SIM-based devices within 2. which in addition can translate incoming voice/text into vibrations for the candidate to 'feel' answers A-D of a given MCQ.

## 7. Conclusion

Even if proctored or invigilated, the online exam solutions available in the market are vulnerable to the malpractices. Examinees can cheat the system with the help of screen sharing applications and live audio call/chat browser -based applications. Our solution SAHEB sets constraints on sharing of screen content for a popular OS, and also sets restrictions on use of input/output audio devices and other peripherals which can be used to access externally provided information to the candidate. The applications running on the examinee's computer need not be terminated since the exam is being dispensed inside a virtual machine.

## References

[1] *About*. (accessed in July 2021). Safe Exam Browser. https://safeexambrowser.org/about_overview_en.html#concept

[2] *The exam browser that eliminates cheating - Mercer | Mettl secure exam browser is here*. (accessed in July 2021). mettl.com. https://mettl.com/safe-and-secure-exam-browser

[3] Evans, A. (2020, November 20). *What is Windows kiosk mode?* Hexnode Blogs. https://www.hexnode.com/blogs/what-is-windows-kiosk-mode/

[4] *Set up iPad and Mac to give tests and assessments*. (2020, September 16). Apple Support. https://support.apple.com/en-us/HT204775

[5] *Student visa*. (accessed in July 2021). The Official Portal of the UAE Government. https://u.ae/en/information-and-services/education/higher-education/student-visa#long-term-visa-for-outstanding-students

[6] *Morse code translator*. (accessed in July 2021). Morse Code World. https://morsecode.world/international/translator.html

[7] *SetWindowDisplayAffinity function (winuser.h)*. (accessed in July 2021). Developer tools, technical documentation and coding examples | Microsoft Docs. https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-setwindowdisplayaffinity

[8] Patnaik, S. (2021, May 24). *Announcing public preview of screen capture protection in Windows virtual desktop*. Microsoft Tech Community. https://techcommunity.microsoft.com/t5/azure-virtual-desktop/announcing-public-preview-of-screen-capture-protection-in/m-p/1991476

[9] Baur, B. M. (2021, June 24). *Enable screen capture protect on Azure virtual desktop #AVD with Microsoft endpoint manager #MEM*. LinkedIn. https://www.linkedin.com/pulse/enable-screen-capture-protect-azure-virtual-desktop-avd-baur

[10] Vaughan-Nichols, S. J. (2015, August 6). *Check point: Certifi-gate-based attacks could take complete control of Android devices*. ZDNet. https://www.zdnet.com/article/certifi-gate-big-android-security-trouble-for-hundreds-of-millions-of-users/

[11] *Restricted screen reading*. (accessed in July 2021). Android Open Source Project. https://source.android.com/devices/tech/config/restricted-screen-reading

[12] Jokiel, T. (accessed in July 2021). Porteus Kiosk - free and open source kiosk software for web terminals. https://porteus-kiosk.org/

[13] *Justbrowsing*. (accessed in July 2021). JustBrowsing LiveCD :: Home. https://justbrowsinglinux.com/

[14] *Opensource kiosk software to get you onto the Web*. (2017, September 10). Webconverger.org. https://webconverger.org/

[15] *Hypervisor*. (2004, December 11). Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Hypervisor

[16] *VirtualBox main API: IDisplay interface reference*. (accessed in July 2021). Oracle VM VirtualBox. https://www.virtualbox.org/sdkref/interface_i_display.html

[17] *VirtualBox main API: IEmulatedUSB interface reference*. (accessed in July 2021). Oracle VM VirtualBox. https://www.virtualbox.org/sdkref/interface_i_emulated_u_s_b.html

[18] Karl-Bridge-Microsoft. (accessed in July 2021). *ApplicationView.PreferredLaunchWindowingMode property (Windows.UI.ViewManagement) - Windows UWP applications*. Developer tools, technical documentation and coding examples | Microsoft Docs. https://docs.microsoft.com/en-

us/uwp/api/windows.ui.viewmanagement.applicationview.preferredlaunchwindowingmode?view=winrt-20348

[19] *How can I prevent users from connecting to a USB storage device?* (accessed in July 2021). Microsoft Support. https://support.microsoft.com/en-us/topic/how-can-i-prevent-users-from-connecting-to-a-usb-storage-device-460ef516-8ac8-07af-e90b-0d9ac55bcd4d

[20] *Turn on/off Bluetooth radio/adapter from cmd/PowerShell in Windows 10*. (accessed in July 2021). Super User. https://superuser.com/questions/1168551/turn-on-off-bluetooth-radio-adapter-from-cmd-powershell-in-windows-10

[21] Drewbatgit. (accessed in July 2021). *EndpointVolume API*. Developer tools, technical documentation and coding examples | Microsoft Docs. https://docs.microsoft.com/en-us/windows/win32/coreaudio/endpointvolume-api

[22] *KeyEventHandler delegate*. (accessed in July 2021). Developer tools, technical documentation and coding examples | Microsoft Docs. https://docs.microsoft.com/en-us/dotnet/api/system.windows.input.keyeventhandler

[23] Sayed, W. (2009, June 18). *Disable print screen key and all keyboard keys in ASP.NET page*. CodeProject. https://www.codeproject.com/Articles/37292/Disable-Print-Screen-Key-and-All-Keyboard-Keys-in

Research Office
Indian Institute of Management Kozhikode
IIMK Campus P. O.,
Kozhikode, Kerala, India,
PIN - 673 570
Phone: +91-495-2809237/ 238
Email: research@iimk.ac.in
Web: https://iimk.ac.in/faculty/publicationmenu.php