Exploiting Virtualisation for Interactive Virtual Teaching

Fabien Duchene, Thomas Given-Wilson, Axel Legay, Sebastien Strebelle
Université catholique de Louvain, Louvain-la-Neuve, Belgium
{firstname.lastname}@uclouvain.be

ABSTRACT

Cyber security is a rapidly evolving, highly complex, and strategically important area of computer science. Teaching cyber security poses many challenges compared with other topics, often requiring more complex tools and direct support and supervision during instruction. This was made significantly more complex with the advent of COVID-19 and having to rapidly shift to entirely virtual classroom environment. This paper overviews the key challenges and solutions practiced during an unexpected and rapid move to virtual teaching of cyber security.

1. THE CHALLENGES OF LEARNING SECURITY

With to the ever-increasing reliance on the Internet, online services, and the advent of the *Internet of Things* (IoT), computer networks security and information technology security in general is becoming a vital part of computer systems and thus of the computer science students' curriculum. One major challenge with cyber security is that it isn't a self-contained subject that can be learned like a new programming language. From hardware fault injections [10, 8] to *cross-site scripting* (XSS) [6] learning about security requires a deep understanding of many components of computers science. Usually, the theoretical part is taught during an interaction-intensive session with the professor. The interaction aspect comes from the fact that to understand a simple buffer overflow [7], the professor needs to discuss several points that are not directly related to security like the memory management in different languages [4]. Similarly in networking security, while teaching a SYN flood attack [5], the professor needs to make sure that TCP [9] is already well understood by the students. The same challenge arises for practical sessions, but with an added layer of complexity: the system configuration. To allow the student to reproduce and deeply understand the security flaws they are exploiting, they often need to run the exact same configurations as their tutor. This is because discovered security vulnerabilities are often quickly fixed, requiring the student to run a very specific version of multiple pieces of software and sometimes vital parts of the operating system (e.g. libraries like glibc).

These requirements of high interactivity and very specific system configurations can be managed inside the normal university environment. Face-to-face lectures provide the time and interaction for quick feedback and discourse to ensure comprehension from students, and quick adaptation on diversions to cover any gaps in student learning. For the , the system environment can be managed by deploying the infrastructure the practicals and making it available on the university's systems. This ensures all students have access to the same environment resolves some of the challenge. The interactivity and rapid feedback are addressed by having tutors observing and discussing with students and being able to interact with the system the student is working on.

To tackle these challenges, we started to virtualise the infrastructure dedicated to the . In this paper, we first describe our virtual architecture and discuss how this helped us through the COVID-19 pandemic. We also more broadly discuss how the interactive aspects had to be adapted during the remainder of the academic year.

2. VIRTUALISING THE INFRASTRUCTURE

The challenge of having a very specific system configuration that is able to be used for a large variety of cyber security lessons and for a large number of students is non-trivial. It is not feasible to have a single system for use by all students since they could conflict with their efforts, or damage the configuration. Similarly, having many (even very slightly) different systems (such as the same OS and tools installed on various lab computers) makes it unlikely that the lessons and cyber attacks being taught will work effectively and large amounts of time will be wasted on understanding and adapting to a new configuration.

The approach we used to address this challenge was to provide a virtual machine (VM) that has the entire environment virtualised, and the main configuration options already configured for the students. An overview of this virtualised environment is shown in Figure 1. The virtualised environment is built on a VM based on Kali Linux [2], a penetration testing oriented distribution to which we added some tools and some specific versions of the system libraries. The students can download the image and run it with VirtualBox [3], a free x86 and AMD64 virtualisation tool.

To allow the students to work easily with the different software required we use docker [1] inside the VM. Docker is a technology that allow running a packaged application with all its dependencies. The application and its dependencies are called a *container*. A further benefit of is that installing different versions of the same software on the same machine is sometimes non-trivial, and asking the students to uninstall and then install a specific version for a specific lab would be sub-optimal.

With this architecture, the students only need to type one command to run the specific container needed for the

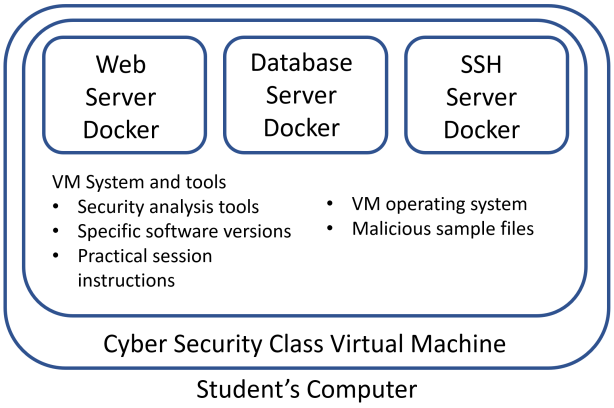

Figure 1: Overview of the virtual environment created for teaching cyber security.

. Another reason of this choice is that if a student end up breaking the software they are working on (e.g. erasing the whole directory containing the web server files), they just need to type one command to delete the container and then start from a fresh new version in a matter of seconds.

The use of the virtualised environment ensured all students had the same configurations and should be able to perform the exploits correctly to both learn about and exploit systems. However, there are still subtle details that vary at different moments in time or according to how the students used their virtualised environment, and so face-to-face interaction and the ability for tutors to interact with the student's system was a key aspect of the normal teaching activities.

The Advent of COVID-19

The advent of COVID-19 required an extremely rapid response due the rapidly evolving situation. On Thursday 12th of March the university had to shift all teaching to online only classes for the following Monday (the lectures for this subject were 0830 on Monday mornings). This required a major shift in the teaching of cyber security and an evolution of the approaches used to ensure that students could continue to study and learn with minimal impact due to the COVID-19 pandemic. To effect this required a concerted effort from all areas of the university; teaching, administration, etc. to make all the tools and capabilities available to ensure that online instruction could continue with minimal disruption.

The virtualisation of the system configuration here turned out to be a major benefit when shifting to online teaching since the virtual environment was made available to students and they could run this on their own computers (including laptops at the university before the COVID-19 pandemic). Of course such a virtualised environment is not small, and so a few students needed to prepare well in advance to download the virtual environment and be able to run it on their own system.

Generally the use of the virtualised environment here were extremely helpful, as even when changes to the environment needed to be made (as the course evolved and lessons changed), these could still be shared online and made available to the students. Although this had some overhead for the students to download the virtualised environment in advance, this was not significantly more complex than providing the virtualised environment on university systems.

3. INTERACTIVE ASPECTS

The more challenging aspect of the adaptation to the COVID-19 pandemic was to maintain the interactive aspects of the teaching while going completely online. This had to be adapted in two different ways, one for the lectures, and another for the labs. Although further adaptations and advances were also made to ensure good support and education outcomes for the students.

For all of the online teaching the university provided Microsoft *Teams* accounts and access for all staff, and "Teams" for each class that had all the students and staff from that class together. This allowed staff to be able to have "meetings" where they could share videos or presentations and other members of the same Teams team could join and observe or interact. However, there were some challenges with both adapting on short notice, using new software, and overload on the Teams servers.

For the lectures the adaptation was tested with a direct streaming of the lectures (internally with teaching staff only) and found to be difficult to manage due to unreliability (at the time) of the Teams environment. The decision was made to pre-record the lectures in small components that covered small parts of each lecture's content. These were then uploaded and made available to the students in advance of the lecture time. During the normal lecture time the professor and some of the tutors were online. The lecture was still planned with specific timing, each video being advised to be watched, and then a time for feedback and questions with the professor conducting. In between these times (e.g. when the students where watching the pre-recorded lecture videos), the professor and tutors would respond to questions written in the Teams team chat. The professor and tutors also adapted and discussed to respond to student queries during the lecture time. This system lead to a 400% increase in the number of questions asked to the professor, as measured by questions counted originally in the lectures and after going online as questions by Teams and email.

For the labs the adaptation was different since there was no common screen or progress during labs before the COVID-19 pandemic. Instead, after the class moved online the students studied on their own virtualised environments with tutors (and sometime the professor) available for the duration of the tutorial time. Typically the students and tutors would converse via written messages during the tutorial time, and share parts of the lessons in text. Additionally the tutors would join smaller group calls with (groups of) students to discuss their progress or assist when the students needed more detailed responses than could easily be done by text. Further, at times when several students had similar problems, the tutors would run their own instance of the virtual environment and share this via Teams with the students, so they could watch the tutor step through the exploits and hacking with ongoing commentary. During this time there was also the opportunity to ask via voice or chat for details on each part of the demonstration.

In addition to feedback directly from teaching staff, the students also provided support and assistance to each other via the shared environment on Teams. It was common for

students to answer each others' simple questions, or share their own discoveries and exploits as part of the lesson. This proved a nice benefit of the shared online teaching environment, where already having all students and teaching staff able to share text allowed for easier copying of key information between students as part of their learning.

To further support the students with their laboratory understanding, the teaching staff developed detailed solutions (or walk-throughs) of each laboratory. These detailed not only the solution, but also how to achieve the solution and the process behind reaching the endpoint of the laboratory (such as how to adjust an exploit based on what was observed).

Due to students no longer being at the university campus (and all university buildings being closed), there was no longer any opportunity for office hours or other visiting and discussion with the professor and tutors. To compensate for this our approach was to have a shared discussion area in Teams where students could ask questions at any time and we would ensure this was checked several times a day during working days (and occasionally on weekends). Responses to questions or requests for assistance here were generally in the form of text chat, but some voice calls and shared screens (including with the virtualised environment shared for common usage) were also performed. This compensated to a large degree, and was also appreciated by the students who could ask questions at their convenience and receive a prompt response.

Further to this, we also provided a schedule where students could arrange a consultation with a tutor in advance to discuss questions about the teaching or the assignments. These consultations were done in the form of a voice chat directly with students, often using shared screens. Some students seemed to fear asking a question in a public place or to disturb a tutor. Providing dedicated one-to-one time with students helped them have a safe environment to ask questions and improve their learning experience.

Another complication was the scheduling and plans for the final exam for the subject. This had originally been planned as an oral exam. Unfortunately this would have been impossible due to restrictions on face-to-face meetings, and an online oral exam posed several complications that had not been tested. The choice was offered to the students to take an adjusted mark based on their assignment during the semester as their final mark, with an exam offered only for students who wished to keep the original balance of marks. (In practice only small number of students chose this option, the rest preferring their adjusted assignment marks.)

The students also acted themselves to adapt to the new environment and the inability to easily discuss lab work in tutorials. Some students created a shared git repository where they uploaded their own versions of the laboratory solutions and various students were able to upload or download solutions. This allowed the students to also learn from each other in addition to the resources provided by the teaching staff. The teaching staff also had access to this and checked for mistakes or misunderstandings that might have been shared. This collection of student-developed solutions and examples provided and excellent addition to the normal lab walk-throughs provided through online teaching.

Some students also created their own syllabus which they added to as the lecture slides and videos came out. They were able to continuously collaborate on their understanding and improve their presentation of the class subject matter.

4. OBSERVATIONS AND FEEDBACK

In addition to the in (virtual) class changes to student behaviour, some larger outcomes were also observed. These were both in the assessment outcomes, and in the course feedback.

The main change in the assessment outcomes related to the option for having the assignments alone account for the entire subject assessment. Many students took this opportunity to develop more comprehensive solutions to the assignments. This was observed both in their engagement related to the assignments (questions, discussions, and feedback) and in the quality of the assignments themselves. This was reflected in a significant improvement over the previous year. This is particularly significant since cyber security is a highly practical subject to learn and practice, and so the increased effort, attention, and results of the assignments was considered a benefit from a learning perspective.

To improve our teaching methods, we submitted a 13 questions survey regarding the lectures to the students at the end of the semester. 19 students out of 60 completed the survey. Figure 2 the results of the question where the students where asked if the format of the lectures was suitable to help them understanding the subject. 21.05% of the student answered with a simple "yes", 52.63% said "yes" and stressed the importance of the interactive discussions with the professor via , and stressed on the importance of discussions with the professors on the forums, no student would have preferred a live-streaming solution, 5.26% said that they would have preferred the teaching to happen in an auditorium and 5.26% said none of the above.

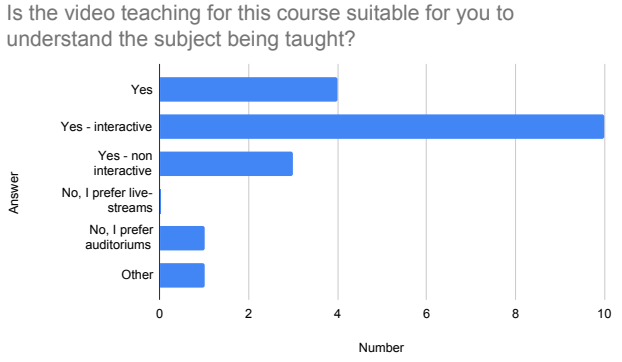

Figure 2: Results of the survey asking the students if video teaching was suitable.

For this question, students were able to add comments to help us understand their choice. While some said that "nothing could replace face-to-face teaching" another said "I love the video system. It allows me to watch the videos at my pace in the best possible conditions. The only downside would be the challenge of resisting the urge to watch the videos later but the interactive-discussion being scheduled [during the lecture time] solved the problem for me".

In Figure 3, we asked the students about the pace of the lectures. 94,74% said that the pace allowed them to under-

stand the subject and 5.26% said that the pace was too slow. None said that the pace was too fast.

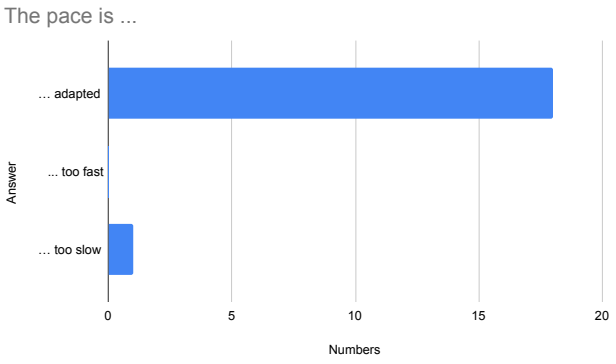

Figure 3: Results of the survey asking the students if the pace of the lectures was suitable.

5. CONCLUSIONS/LESSONS LEARNED

Teaching computer science and in particular cyber security is an ever evolving process that needs to be flexible and quickly adapt to the rapidly changing landscape. Having already started to use a virtualised environment for teaching of cyber security proved extremely helpful when teaching had to be moved away from the university environment to an entirely virtual one. Various other adaptations were made with respect to how lectures were delivered and how to interact with students, including adding more opportunities and methods for interaction using online only teaching environments. In addition to adaptations from teaching staff to ensure the continuity and required interactivity, the students also adapted to support themselves and each other during this pandemic. The resulting assessment outcomes and student feedback reflected that the adaptations were overall successful and despite the changes, the students' learning outcomes did not appear to diminish.

6. REFERENCES

[1] Docker - project website. `https://www.docker.com/`, 2020. [Online; accessed 8 July 2020].

[2] Kali linux, an advanced penetration testing linux distribution used for penetration testing, ethical hacking and network security assessments. `https://www.kali.org/`, 2020. [Online; accessed 8 July 2020].

[3] Oracle vm virtualbox. `https://www.virtualbox.org/`, 2020. [Online; accessed 8 July 2020].

[4] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, Q. Zhang, and H. Hinton. Stackguard: Automatic adaptive detection and prevention of buffer-overflow attacks. In *USENIX security symposium*, volume 98, pages 63–78. San Antonio, TX, 1998.

[5] W. Eddy. TCP SYN Flooding Attacks and Common Mitigations. RFC 4987, Aug. 2007.

[6] S. Fogie, J. Grossman, R. Hansen, A. Rager, and P. D. Petkov. *XSS Attacks: Cross Site Scripting Exploits and Defense*. Syngress Publishing, 2007.

[7] J. C. Foster, V. Osipov, and N. Bhalla. *Buffer Overflow Attacks*. Syngress Publishing, 2005.

[8] T. Given-Wilson, N. Jafri, and A. Legay. The state of fault injection vulnerability detection. In M. F. Atig, S. Bensalem, S. Bliudze, and B. Monsuez, editors, *Verification and Evaluation of Computer and Communication Systems - 12th International Conference, VECoS 2018, Grenoble, France, September 26-28, 2018, Proceedings*, volume 11181 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 2018.

[9] J. Postel. Transmission Control Protocol. RFC 793 (Internet Standard), Sept. 1981. Updated by RFCs 1122, 3168, 6093, 6528.

[10] H. Ziade, R. Ayoubi, and R. Velazco. A survey on fault injection techniques. *Int. Arab J. Inf. Technol.*, 1:171–186, 01 2004.