Proposal No. 830929
Call H2020-SU-ICT-03-2018

Project start: 1 February 2019
Project duration: 42 months

Cyber Security for Europe

D7.4

Common virtual lab with open-source tools for research and development

Document Identification	
Due date	29 April 2022
Submission date	29 April 2022
Revision	1.0

Related WP	WP7	Dissemination Level	Public
Lead Participant	BRNO	Lead Authors	Valdemar Švábenský (BRNO) Jan Vykopal (BRNO) Attila Farkas (BRNO)
Contributing Beneficiaries	—	Related Deliverables	D7.2

Abstract

The previously published deliverable D7.2 presented Cyber Sandbox Creator (CSC): an open-source tool for creating portable and lightweight virtual labs. The associated report defined the user requirements, described the technical architecture, and proposed use cases for cybersecurity education, testing, and certification. This document, which is an accompanying report to the deliverable D7.4, builds upon D7.2 by describing two main contributions: (1) open-source tools and materials for supporting research and development using CSC and (2) demonstrating practical use cases of CSC.

This document is issued within the CyberSec4Europe project. This project has received funding from the European Union's Horizon 2020 Programme under grant agreement no. 830929.

Executive Summary

Cyber Sandbox Creator (CSC) is an open-source tool for building lightweight virtual laboratories for cybersecurity education, testing, and certification. Its first version was released in February 2020 as open-source software available at https://gitlab.ics.muni.cz/muni-kypo-csc/cyber-sandbox-creator. Since then, CSC has been used in practice numerous times and has been continuously improved to address the needs of a broad range of users.

CSC has been used in seven institutions in five European countries to support education, testing, or certification. Moreover, its capabilities have been demonstrated in two research publications.

This accompanying report (deliverable D7.4 is actually the virtual lab with open-source tools for research and development) introduces open-source tools, materials, and use cases deployed and demonstrated in the lab. The tools and materials include:

- the latest version of CSC (3.0.0),
- customizable images of operating systems usable with CSC,
- a transparent toolset for logging command-line actions inside CSC-generated environments,
- an exemplary educational game for practicing cybersecurity skills,
- dataset of command-line logs from authentic educational contexts (includes contributions also by third parties outside the CyberSec4Europe pilot),
- best practices for designing new cybersecurity training content, and
- a ready-made lab environment for testing and certification of hardware devices.

The use cases include:

- preparing a lab environment for Flagship 2 Challenge, a massive online exercise delivered by the members of the CyberSec4Europe pilot,
- testing sandboxes before their cloud deployment in the KYPO Cyber Range Platform,
- developing a locally virtualized testing environment for specialized purposes,
- verifying the capabilities of a realistic manufacturing execution system,
- teaching university classes, and
- auditing the capabilities of hardware devices in an isolated, reproducible environment.

The target audience for this document are users who are familiar with the basics of CSC and consider deploying it for their own use cases. This document will provide an inspiration and open-source examples that others can adopt or adapt.

Document information

Contributors

Name	Partner
Petr Švenda	BRNO
KYPO Cyber Range Platform authors	CONCORDIA

Reviewers

Name	Partner
Jozef Vyskoč	VaF
Vasileios Gkioulos	NTNU
Vashek Matyáš	BRNO

History

Version	Date	Authors	Comment
0.1	2022-01-31	Valdemar Švábenský, Jan Vykopal, Attila Farkas	First version submitted after the internal reviews
0.2	2022-04-27	Valdemar Švábenský, Jan Vykopal, Attila Farkas	Incorporated the comments from peer reviewers
1.0	2022-04-29	Ahad Niknia	Final check, preparation and submission

Table of Contents

List of Acronyms

B	**BRNO**	Masaryk University, Brno, Czech Republic
C	**CS4E**	CyberSec4Europe, Cyber Security for Europe
	CSC	Cyber Sandbox Creator
	CTF	Capture the Flag
J	**JAMK**	Jyväskylän ammattikorkeakoulu (University of Applied Sciences, Finland)
K	**KYPO CRP**	KYPO Cyber Range Platform
O	**OS**	Operating system
R	**RGCE**	Realistic Global Cyber Environment
Y	**YAML**	YAML Ain't Markup Language (a human-readable data serialization language)

Glossary of Terms

C **Capture the Flag game, CTF game**

A training activity in which participants solve technical cybersecurity tasks. Completing each task results in finding ("capturing") a text string called flag.

Cyber range

A sophisticated virtual network infrastructure that allows hosting multiple sandboxes for multiple users at the same time.

S **Sandbox**

An isolated virtual environment that allows cybersecurity experimentation (including the execution of cyber attacks) without negative consequences on the underlying IT infrastructure.

(Cyber) Sandbox Creator

A tool developed as a part of the CS4E project that allows the user to create arbitrarily defined sandboxes.

Sandbox Manager (renamed from *Sandbox Runner* in D7.2)

A tool developed as a part of the CS4E project that allows the user to run the sandboxes created with the Cyber Sandbox Creator.

T **Training**

A sequence of tasks completed in a virtual environment under the guidance of a human or automated tutor. The learner's goal is to develop and practice cybersecurity skills. The training can have various formats, most commonly a cybersecurity game, in particular Capture the Flag game.

V **Virtual lab**

Depending on the context, this can mean (a) Cyber Sandbox Creator and Sandbox Manager, or (b) the environment (sandbox) generated by Cyber Sandbox Creator and deployed using the Sandbox Manager.

1 Introduction

The deliverable document D7.2 [D7.2] argued for the importance of a tool for creating lightweight virtual lab environments. We developed such a tool, called the *Cyber Sandbox Creator* (CSC), and explained its technical background and implementation in the deliverable D7.2.

The purpose of this accompanying document (deliverable D7.4 is the virtual lab with open-source tools for research and development) is to report on the practical usage of CSC in cybersecurity testing, education, and certification. We present numerous open-source tools, materials, and their associated use cases, as well as explain how they support the goals of the CyberSec4Europe pilot.

1.1 Role of This Document in Relation to the Deliverable D7.4

This document is the accompanying part to the deliverable *"D7.4 Common virtual lab with open-source tools for research and development"* due in Month 39. It extends the internal milestone document *"Open-source tools, data, and use-cases deployed and demonstrated in the lab"* submitted in Month 33. The key part of the deliverable D7.4 is the associated GitLab repository of CSC [CSC] and the set of open-source tools and materials referenced from this text.

1.2 Target Audience and Assumed Prerequisite Knowledge

The target audience for this document are users with IT background who are familiar with the basics of CSC and consider deploying it for their own use cases. This text builds upon and extends the deliverable D7.2, so it assumes the knowledge of the previously defined terminology. For those who are unfamiliar with CSC and for the general public, we strongly recommend starting with reading the D7.2 document first, in order to understand the motivation and use cases for CSC. To avoid redundancy, the in-depth explanation of CSC's inner workings is not repeated here.

1.3 Document Outline

This document is structured into four sections. Section 2 presents open-source tools and materials that are either generated by CSC or otherwise relate to it. Section 3 reports specific use cases of CSC from the areas of testing, education, and certification. Finally, Section 4 summarizes the achievements throughout the project, explains the difference from the previous deliverable D7.2, and proposes future work.

2 Open-source Tools and Materials Usable with CSC to Support Research and Development

We created multiple tools and materials to be used with CSC to support cybersecurity testing, education, and certification. For the purposes of this document, we define a *tool* as a software application and its associated documentation. *Material* is any other practical output relevant to the goals of the CyberSec4Europe pilot.

This section reviews tools and materials that satisfy all the following conditions:

1. They are
    a. either usable with CSC to support the goals of CyberSec4Europe or
    b. a product of CSC deployment demonstrating the CyberSec4Europe goals in practice.
2. They were created by the developers or users of CSC.
3. They are publicly available under an open-source license.

All the tools and materials are in the English language. Other language mutations are currently not under consideration, because the tools and materials depend on many external components that are in English as well. Therefore, translating the tools and materials would be impractical.

2.1 CSC Core

By "Core" we mean the minimal viable product that can generate locally virtualized lab environments. (Extensions directly usable with CSC are described in separate subsections.) The core tool enables customized deployment of various environments based on the user's needs. Moreover, the tool:

- is fully open-source in GitLab [CSC],
- contains thorough documentation and wiki pages, and
- is being gradually covered by integration tests.

2.1.1 Updates from the Previous Deliverable D7.2

Since the previous major release 1.0.1 (November 2020) released as a part of the D7.2, various stakeholders have been using CSC (see Section 3 for the description of specific use cases). After gathering feedback from the stakeholders, we defined the plan for further extensions of CSC. As a result, the development of CSC has remained use-case driven, supporting the needs for testing, education, and certification practices.

CSC was extended in several updated releases, with the latest being 3.0.0 (January 2022). Each release is thoroughly documented in the CSC's GitLab repository [CSC]; the most important updates include:

- CSC-generated environments support accessing USB devices in virtual machines with a graphical user interface. (This enables the certification use case.)
- The sandbox definitions are compatible with the KYPO Cyber Range Platform (KYPO CRP).
- A Vagrant wrapper (Sandbox Manager) was added to simplify the basic operations with a sandbox instance, hide overwhelming technical output from users by default, and give them more informative error messages.
- Sandbox Creator and Manager are accessible from any location on the hosting computer.
- Installation of CSC is possible via *pip* directly from PyPI (Python Package Index). An API was added to make Sandbox Creator and Manager accessible directly from Python.

2.1.2 Comparison with Related State-of-the-art Tools

The current version of CSC differs from other state-of-the-art approaches [Labtainers, CyRIS, Lability] in the following:

- Compatibility with all major operating systems (Windows, Linux, macOS) for hosting the lab.
- Thorough documentation for each user role.
- Relatively simple interface and usage.
- Can be easily used for purposes other than cybersecurity education.

More in-depth comparison is available in [ThesisFarkas].

2.2 Images of Operating Systems (Base Boxes)

The essence of CSC is creating virtual lab environments. This process requires using high-quality images of operating systems for Vagrant [Vagrant] and VirtualBox [VirtualBox]. These images are also called *base boxes* (or simply *boxes*). Although many public base boxes are offered on Vagrant Cloud [VagrantCloud], they have the following limitations:

- They might not be properly tested, documented, or fully functional.
- Even if they are functional, they can change any time without a guarantee that they will remain operational for CSC use cases.
- The contents of the boxes are unknown; they might contain potential security vulnerabilities or even malicious software.

Therefore, we have been creating own base boxes, which brings two key contributions:

1. The boxes are guaranteed to work with CSC since they are under our control.
2. The source files of these images are available [GitLabImages], allowing anyone to generate their modified versions. This enables us, for example, to create almost identical images usable in the KYPO CRP [KYPO]. As a result, CSC is better interoperable with other existing platforms.

Our images are recognizable by the prefix „munikypo“ before their name. The following operating systems are included:

- Linux images:
  - Kali (2020 version is recommended, 2019 is now deprecated),
  - Debian 10,
  - Ubuntu 18.04 (work-in-progress for the 3.1.0 release),
  - CentOS 7.9 (work-in-progress for the 3.1.0 release).
- Windows images:
  - Windows 10,
  - Windows Server 2019.

Since the images are relatively minimal, they are highly flexible. Users of CSC can provision them based on their needs for various use cases in testing, education, and certification. Therefore, a single image can be used to build many different unique lab environments. In total, the boxes were downloaded more than 2200 times (the most popular being Kali Linux).

Finally, we refer readers interested in technical details of base boxes to another document [ThesisVydra], which also includes a practical guide for image developers. It describes the workflow for creating new images and experience with using them.

2.3 Toolset for Logging Linux Shell Commands

The logging toolset is an extension to the CSC Core. It enables the option of transparent monitoring of shell commands inside the lab environment (sandbox) generated by CSC. This feature is important for teachers and educational researchers to understand the students' learning processes, assess them, and provide personalized feedback.

Technically, the toolset is implemented as a set of open-source Ansible [Ansible] roles that can be included in the provisioning definition [FIEtoolset]. After deployment, the toolset records the commands and metadata, such as timestamps, from Linux virtual machines. These log records are then forwarded to a central storage for further processing.

The logs do not store any personally identifiable information to preserve the privacy of the training participants. Students are identified only by an arbitrary numerical ID that is not associated to their identity. Moreover, the transfer of the logs from the trainee to the log server is encrypted and authenticated.

The extended version of the toolset has been published in a conference article [FIEtoolset]. This toolset is compatible with KYPO CRP as well, so the same Ansible roles can be used both with CSC and KYPO CRP.

2.4 Sample Cybersecurity Game for Education

The CSC Core along with the above-mentioned tools and materials (base boxes, logging toolset) form a "CSC suite" that enables creating various lab environments. An example is a serious cybersecurity game: a sandbox and a set of tasks that can be solved in the sandbox to exercise one's cybersecurity skills.

We share a ready-made educational game called Junior Hacker Training [GitLabJunior], which demonstrates the capabilities of the CSC suite. The lab environment for the game was generated by CSC (see Section 2.1) using our base boxes (see Section 2.2). Then, the intermediate sandbox definition was enhanced by the logging toolset (see Section 2.3). Using CSC, educators and students can deploy the game's lab environment locally on their standard personal computers with minimal setup.

What is more, the lab environment definition is compatible with KYPO CRP. As a result, a single game can be deployed either locally using CSC or in cloud using KYPO CRP (see Figure 1). More details are explained in a recent conference article [FIEscalable].

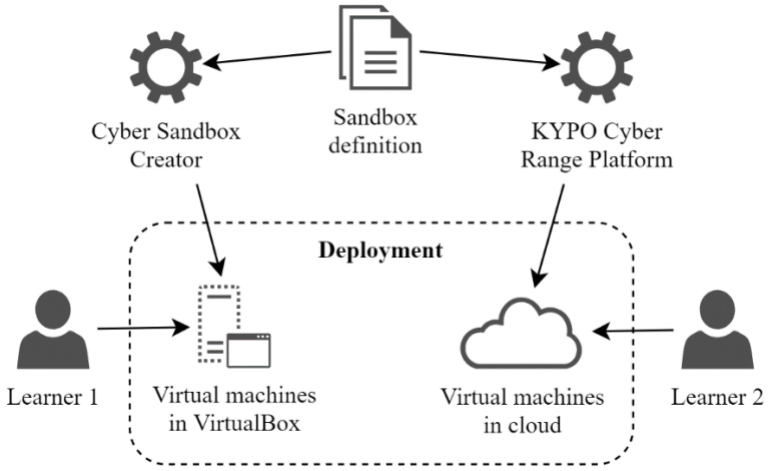

Figure 1: The sandbox definitions for CSC and KYPO CRP are compatible [ThesisSvabensky].

When using a CSC-generated environment for teaching, such as for this example game, an additional shared web portal is needed for features such as presenting the task assignments, collecting answers from students, or scoring. In the deliverable D7.2 [D7.2, Section 4.5.2], we argued for selecting CTFd [Chung17] as a suitable open-source platform for these purposes. A recent comparison of these platforms [Karagiannis20] confirms CTFd as a fitting candidate. We have demonstrated the usage of CSC and CTFd in a large remotely taught class with 207 students. The results were published in a conference paper [SIGCSEapg], and its supplementary materials include instructions for CTFd deployment.

2.5 Sample Dataset Resulting from Cybersecurity Games

As another open-source material, we share the command-line logs collected from authentic teaching contexts: cybersecurity games and exercises deployed in CSC or KYPO CRP. In alignment with open data initiatives in science, this dataset supports efforts in cybersecurity education research. The data are freely available on Zenodo as a part of a journal publication [Dataset], which also describes the data format and value. The possible use cases for the data include but are not limited to training and testing machine learning models (for example, classifiers for skill assessment), evaluating data mining methods, correlating actions from multiple sandboxes, prototyping student models, or detecting security threats.

2.6 Guidelines for Developing New Cybersecurity Games

Educators interested in creating cybersecurity training content may benefit from our documented best practices [GitLabGuidelines]. The material explains how to design a cybersecurity game from both educational and technical perspective; the latter may involve using CSC. The guidelines were used in practice to create the game previously described in Section 2.4.

The guidelines are suitable for beginner game designers. They focus on creating games that teach technical skills, especially in offensive security, but some guidelines are also applicable to defensive challenges. The expected game infrastructure consists of networked virtual machines generated using CSC and the associated tools and materials.

2.7 Example Lab Environment for Certification

We prepared an open-source repository with the definition of a lab environment for certification [GitLabCertification]. Users can instantiate the environment locally and test the properties of cryptographic smartcards using a test suite called SCRUTINY [ThesisNagy]. This practical output includes a guide explaining how to use the test suite in two main scenarios:

1. The user verifies a smartcard profile supplied by the vendor (or community) to determine whether the smartcard is the same as the one that was certified. In other words, it will replicate the steps conducted during certification and report compliance or individual differences.
2. The user generates a smartcard profile and compares other devices to it.

To ensure the trustworthiness of the certification tools, the lab environment definition and the associated test suite are fully open-source.

2.8 Summary of Open-Source Artifacts

The table below lists all the above-mentioned tools and materials. It also explains which CSC-related user roles [D7.2, Section 2.1] can benefit the most from these artifacts.

Artifact	**What it is**	**Who can use it**	**Why it is useful**
CSC (core)	A tool for generating lightweight, locally virtualized lab environments	Educator, Trainee, Researcher, Developer, Specialist, Auditor	Enables anyone to create a custom environment that suits their needs
Base boxes	A set of CSC-compatible images of virtual machines and their customizable source files	Educator, Trainee, Researcher, Developer, Specialist, Auditor	Allows creating virtual environments using verified and functional building blocks
Logging toolset	Ansible roles for capturing command-line logs from the virtual lab environments	Educator, Researcher, Developer, Specialist	Collects information about user actions inside the lab in a transparent way
Cybersecurity game	A set of hands-on tasks and a ready-made lab environment for completing them	Educator, Trainee	Enables practicing one's cybersecurity skills
Dataset	Semi-structured command-line logs and metadata collected from authentic teaching contexts	Educator, Researcher	Captures how students solved cybersecurity assignments, enabling to conduct research in educational data analytics or cybersecurity modeling
Guidelines	Best practices for creating new cybersecurity games	Educator	Simplifies and structures the process of developing new training content
Certification lab	A ready-made lab environment and tools for testing hardware devices	Researcher, Developer, Specialist, Auditor	Provides unified and reproducible conditions for experiments and audits

3 Use Cases Demonstrated in the Lab Environment

This section documents practical use cases of the lab environments generated by CSC. The use cases are grouped into three categories: testing, education, and certification [D7.2, Section 6].

3.1 Developing a Testing Environment

CSC allows creating custom virtual environments for research and software development. To create such an environment, the user needs to complete the following steps:

1. *Write a topology definition.* The topology definition is a text file in YAML format that defines the properties of the virtual machines (such as the OS image that will be used) and their networking. The users of CSC reported that this definition is easy to write (usually 50–70 lines of human-readable format) by modifying the existing examples [CSC] or even from scratch.
2. *Run Cyber Sandbox Creator.* CSC generates a definition file for Vagrant (a tool that that builds a locally virtualized environment) [Vagrant] as well as the base Ansible provisioning [Ansible] that initializes and interconnects the hosts. If the topology definition contains any syntax errors, they are reported now, and the user must fix them before proceeding.
3. *Write user provisioning definition.* In this step, host configuration is specified, such as the content of file system, which is tailored to the specific use case. At this stage, users may simplify their job by using publicly available definitions from the Ansible Galaxy repository [AnsibleGalaxy]. Again, possible syntax errors are reported only when the sandbox is being built.

Use cases related to testing are described below.

3.1.1 Local Development of Sandboxes for CyberSec4Europe Flagship 2 Challenge

We cooperated with JAMK (Jyväskylä University of Applied Sciences in Finland) to help prepare the technical infrastructure for the Flagship 2 Challenge of the CyberSec4Europe pilot (D6.5, due in Month 36). The goal was to develop a sandbox of two interconnected Linux machines with various analytical software tools. The participants of the Flagship 2 Challenge instantiated this sandbox locally and used it to independently analyze the given forensic data and malware samples. Then, they reported their findings to their respective teams, which worked in parallel using JAMK's RGCE cyber range [RGCE].

We have been iteratively developing and testing this sandbox using CSC. The sandbox has been tested for Windows, Linux, and Mac host operating systems and is publicly available [GitLabFlagship]. It also includes step-by-step setup instructions that will allow even first-time users to work with it.

This effort practically demonstrated cyber range federation [D7.1]: both *technical* in terms of sharing cyber range resources, and *operational* in terms of sharing content for the exercise. The added benefit was that it was possible to save resources of RGCE, since thanks to CSC, a part of the exercise was hosted locally on the computers of participants.

3.1.2 Local Development of Sandboxes for KYPO Cyber Range Platform

Using CSC, we locally developed sandboxes that were subsequently deployed to KYPO CRP. The goal was to create several different networked lab environments consisting of interconnected Linux or Windows machines. The lab environments were intentionally small (from 2 to 5 machines) so that they could be instantiated both locally on standard personal computers and in cloud.

This effort demonstrated inter-pilot cooperation with CONCORDIA, another of the four EU pilots [CONCORDIA], as well as operational federation of cyber ranges [D7.1] since the same content can be shared between multiple platforms. More specifically, sandbox definitions used by CSC and KYPO CRP can be stored in and instantiated from a single Git repository, which we have demonstrated on the Junior Hacker game (see Section 2.4). The repositories of sandbox definitions can be shared so that other users can benefit from them, as the example of the publicly available game shows.

Another benefit of using CSC for this use case is that local development is quicker and more direct; it enables catching and correcting mistakes easily before instantiating the sandbox in cloud. Approximately 30 students of the course *Seminar of Simulation of Cyber Attacks* at Masaryk University used this approach to create sandboxes for their serious games [ITiCSEenhancing] in Spring 2020 and Fall 2020 semesters.

3.1.3 Local Development of Other Sandboxes

This use case is similar to the previous one, but the goal here was not to develop sandboxes for KYPO CRP. Instead, we list examples of how the sandboxes were instantiated directly using CSC for various purposes.

Usage at our home institution (BRNO) includes the following:

- Cybersecurity education researchers prepared exercise sandboxes for monitoring of user actions to collect data for research.
- A student has been creating intentionally vulnerable lab environments for demonstrating cyber attacks and defense capabilities [ThesisBelajova].
- A student created small lab environments to locally test the execution and behavior of certain Linux commands, especially in a network [ThesisHemalova].
- A student created a prototype demonstration of adaptive cybersecurity training [ThesisPavelu].
- A student developed cybersecurity training to support the hiring process at a company [ThesisKrajnikova]. The development employed CSC to build virtual machines, which will run Docker containers [Docker] for the training.

Usage outside of our home institution includes the following:

- Students of Slovak University of Technology have been developing sandboxes for cybersecurity exercises and testing since October 2020. The work will be finished in July 2022.
- Student of a Brno University of Technology demonstrated the capabilities of CSC for generating cybersecurity testing environments [ThesisZidovsky].
- Students of a high school in Brno tested CSC and used it to develop a networked environment that was later used within a cybersecurity game.
- A private company in Germany will use CSC to develop sandboxes for commercial cybersecurity testing and training.

In the vast majority of cases, the users were able to work with CSC independently, and further assistance from our side was not needed. The only notable exception was when our partners from Slovak University of Technology discovered a bug with using the Windows 10 image for Vagrant, which is planned to be fixed in the next upcoming release. This discovery demonstrates the importance of testing with external users.

3.1.4 Manufacturing Execution Systems Application

CSC was also used in the industry. The project KYPO4INDUSTRY (K4I) and UNIS company [UNIS] benefited from the CSC while creating a virtual testing environment. This sandbox consisted of many manufacturing devices like terminals, supply lines, printers, and a database. UNIS used the generated sandbox to test their product, an industrial manufacturing controlling software, without purchasing hundreds of costly hardware components [ThesisFarkas].

The starting point for this industry collaboration was the need to simulate hundreds of devices – not only desktop computers, but also devices such as printers. Locally hosted virtual machines were insufficient for this purpose due to the lack of resources on the hosts. Therefore, we used CSC-generated virtual machines, each of which contained different types of devices inside Docker images. In addition, this collaboration lead to the first usage of Windows images (even those deployed on a Windows host).

3.2 Using the Lab Environment in Educational Activities

Educators used CSC to prepare the virtual environment for various hands-on exercises. With minimal setup, students then instantiated the environment locally on their hosts and solved practical assignments in it.

3.2.1 CyberSec4Europe Flagship 2 Challenge

The Flagship 2 Challenge took place in January 2022. We contributed to it in two main aspects:

- First, we collaborated with JAMK to generate an isolated sandboxed environment for the exercise. Subsequently, 61 participants who enrolled in the *analyst* role used the environment to examine various data samples as part of the exercise. The sandbox consisted of two interconnected virtual machines: Kali and Hive [GitLabFlagship]. The content of the sandbox has been iteratively developed and adjusted based on the requirements set by JAMK. The final sandbox has been tested on all three major operating systems: Windows, Linux, and Mac to accommodate a wide range of participants. Its readme also includes detailed instructions to simplify its deployment.
- Second, before and during the exercise, we provided technical support to participants who used the sandbox. All the discovered issues and their solutions were subsequently documented in the CSC wiki, so that future users can benefit from the lessons learned. The most important takeaways include the discovery that some antivirus programs block the download of Vagrant boxes, and that for an average sandbox, the users need at least 20 GB of free disk space, which was problematic for some participants.

3.2.2 University Classes

In November 2020 and May 2021, more than 250 students of the *Information security and cryptography course* at Masaryk University completed a training in a sandbox locally generated with CSC. Since each student instantiated the sandbox on their own computer, it was possible to teach the classes on a massive scale. The sandbox was complemented by the task presentation via CTFd, as indicated in Section 2.4.

Our teaching experience was published in a conference paper [FIEscalable]. Overall, the learners valued the realistic nature of the virtual environment that enabled them to exercise theoretical concepts and tools. The instructors valued time-efficiency when using CSC to prepare and deploy the hands-on activities.

Next, 18 students of The Hague University of Applied Sciences completed the publicly available training [GitLabJunior] in a sandbox locally generated with CSC. Their shell commands were collected using the logging toolset [FIEtoolset] and published as part of the dataset [Dataset]. We consider this remote international collaboration a success, since the students were able to instantiate the CSC-generated environment without any issues and reported learning about new tools. However, we also discovered that students tried to obtain the answers (flags) to the game tasks using the easiest way, so some groups worked together by sharing their screen. Other students searched for the answers in the sandbox repository or made expert guesses. To mitigate this issue, we examined the generation of personalized tasks within the sandbox to prevent this behavior [SIGCSEapg].

3.3 Using the Lab Environment for Certification Activities

Since the version 2.1.0 (August 2021), CSC-generated lab environments support the connection of USB devices inside virtual machines with a graphical user interface. We demonstrated this feature in the use case for certification of cryptographic smart cards.

First, we generated a definition of a lab environment for certification (see Section 2.7). This definition is open-source so that other analysts may reuse it or adapt it for their purposes. The lab environment consists of a single Linux virtual machine that contains preinstalled Python scripts for smartcard certification [ThesisNagy].

After connecting a smartcard reader to the host computer via a USB, smartcards can be examined in the lab environment. The Python test suite for certification, called SCRUTINY, generates a JSON certificate and an HTML report, which details the features of the smartcard (see Figure 2). The whole testing process is automated and instant so that no additional special skills are needed. The results interpretation is straightforward thanks to the structured report that is generated.

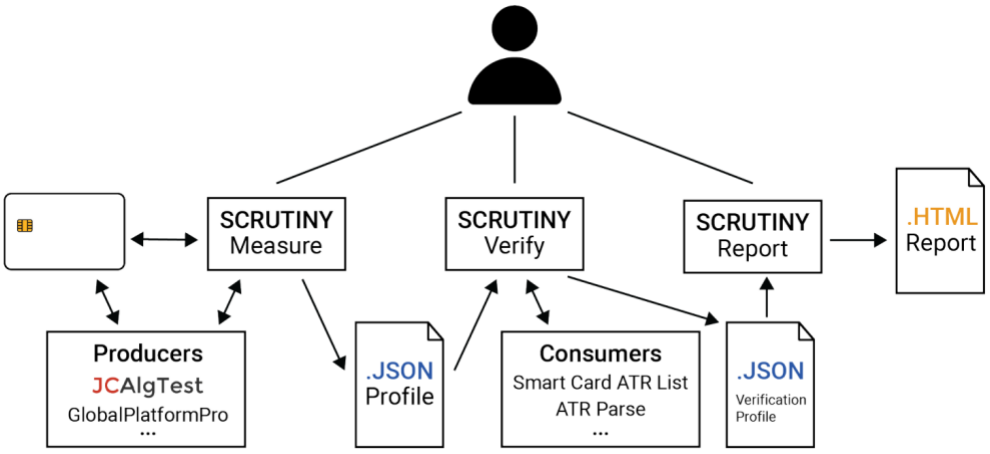

Figure 2: SCRUTINY working pipeline for smartcard certification [ThesisNagy]

There are two main workflows for the certification use case.

- A Developer/Specialist from a company that issues a smartcard *S* can upload its JSON certificate to a public URL as a declaration of the features supported by *S*.
- An Auditor/Researcher who wants to examine the smartcard *S* will instantiate the same environment locally and connect a USB smartcard reader. Then, the smartcard *S* can be verified under replicable conditions with the pre-installed test suite, so that the users can compare the features of *S* with the issuer-declared certificate. This is important for the future reproducibility of certification activities by anyone who might want to verify the compliance of the certified hardware.

4 Conclusion

This report presented the open-source tools, materials, and use cases of Cyber Sandbox Creator (CSC), a software application for creating lightweight virtual lab environments. This document is accompanying the D7.4 deliverable, whose core is the GitLab repository of the application [CSC] and all the tools/materials linked above.

4.1 Current Status

CSC has supported efforts in education, testing, or certification in at least seven institutions across the European Union (in Czechia, Finland, Germany, Netherlands, and Slovakia). The institutions include CyberSec4Europe partners, as well as those outside the project. Moreover, research use cases featuring CSC have been published in two conference articles [FIEscalable, FIEtoolset].

Thanks to the various practical use cases, we were able to collect rich feedback from users. As a result, CSC received several significant updates compared to the prototype previously documented in the deliverable D7.2. These include, for example: support for USB devices, installation from PyPI, Python API, and enhanced compatibility with the KYPO Cyber Range Platform.

Overall, we succeeded in achieving the following previously-defined work plans [D7.2, Section 7.2]:

- Developing images of virtual hosts, including the support for MS Windows boxes.
- Creating a role and example for the command log server configuration.
- Selecting a suitable CTF portal and providing its example configuration.
- Testing the federation of CSC with JAMK RGCE cyber range [RGCE] in a cybersecurity exercise.

4.2 Plans for Further Development

The only remaining plan we set in D7.2 and did not yet achieve is the estimation of the required system resources for sandbox deployment. We are working on this feature for the final release 3.1.0 before the CyberSec4Europe project ends in July 2022, along with the following:

- Increasing the robustness of the lab environments by adding more validation.
- Testing and improving the stability of the lab environments on various host operating systems.
- CSC output verbosity control.

Other possibilities for future work, which, however, we will not deal with, include:

- Graphical user interface for using the Cyber Sandbox Creator or Manager.
- Graphical user interface for creating topology definitions and assigning Ansible roles to the hosts.
- Support of complex network topologies.

References

[Ansible] Red Hat. Ansible. Online at https://www.ansible.com/ [Accessed on January 20, 2022].

[AnsibleGalaxy] Red Hat. Ansible Galaxy. Online at https://galaxy.ansible.com/ [Accessed on January 20, 2022].

[Chung17] Kevin Chung. Live Lesson: Lowering the Barriers to Capture The Flag Administration and Participation. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*. 2017. https://www.usenix.org/conference/ase17/workshop-program/presentation/chung

[CONCORDIA] CONCORDIA. Cyber security cOmpeteNce fOr Research anD Innovation. Online at https://www.concordia-h2020.eu [Accessed on January 20, 2022]

[CSC] CyberSec4Europe. Cyber Sandbox Creator. Online at https://gitlab.ics.muni.cz/muni-kypo-csc/cyber-sandbox-creator [Accessed on January 20, 2022].

[Cyris] JAIST. CyRIS: Cyber Range Instantiation System. Online at https://github.com/crond-jaist/cyris [Accessed on January 20, 2022].

[D7.1] CyberSec4Europe. Report on existing cyber ranges, requirements. Online at https://cybersec4europe.eu/wp-content/uploads/2020/09/D7.1-Report-on-existing-cyber-ranges-and-requirement-specification-for-federated-cyber-ranges-v1.0_submitted.pdf [Accessed on January 20, 2022].

[D7.2] CyberSec4Europe. Virtual lab for open-source tools education and research. Online at https://cybersec4europe.eu/wp-content/uploads/2021/01/D7.2-virtual_lab-v0.2_submitted.pdf [Accessed on January 20, 2022].

[Dataset] Valdemar Švábenský, Jan Vykopal, Pavel Seda, and Pavel Čeleda. Dataset of Shell Commands Used by Participants of Hands-on Cybersecurity Training. In Elsevier Data in Brief, 2021. https://www.sciencedirect.com/science/article/pii/S2352340921006806

[Docker] Docker. Online at https://www.docker.com/ [Accessed on January 20, 2022].

[FIEscalable] Jan Vykopal, Pavel Čeleda, Pavel Šeda, Valdemar Švábenský, and Daniel Tovarňák. Scalable Learning Environments for Teaching Cybersecurity Hands-on. In 2021 IEEE Frontiers in Education Conference (FIE). New York, NY, USA: IEEE, 2021. https://ieeexplore.ieee.org/document/9637180

[FIEtoolset] Valdemar Švábenský, Jan Vykopal, Daniel Tovarňák, and Pavel Čeleda. Toolset for Collecting Shell Commands and Its Application in Hands-on Cybersecurity Training. In 2021 IEEE Frontiers in Education Conference (FIE). New York, NY, USA: IEEE, 2021. https://ieeexplore.ieee.org/document/9637052

[GitLabCertification] Masaryk University. SCRUTINY sandbox. Online at https://gitlab.fi.muni.cz/cybersec/cs4e/scrutiny-sandbox [Accessed on January 20, 2022].

[GitLabCTFd] Masaryk University. CTFd deploy. Online at https://gitlab.fi.muni.cz/cybersec/cs4e/ctfd-deploy [Accessed on January 20, 2022].

[GitLabFlagship] Masaryk University. Flagship 2 sandbox. Online at https://gitlab.fi.muni.cz/cybersec/cs4e/flagship2-sandbox [Accessed on January 20, 2022].

[GitLabGuidelines] Masaryk University. Guidelines for designing cybersecurity serious games. Online at https://gitlab.ics.muni.cz/muni-kypo-trainings/guidelines [Accessed on January 20, 2022].

[GitLabImages] Masaryk University. MUNI-KYPO-IMAGES. Online at https://gitlab.ics.muni.cz/muni-kypo-images [Accessed on January 20, 2022].

[GitLabJunior] Masaryk University. Junior Hacker Training. Online at https://gitlab.ics.muni.cz/muni-kypo-trainings/games/junior-hacker [Accessed on January 20, 2022].

[ITiCSEenhancing] Valdemar Švábenský, Jan Vykopal, Milan Cermak, and Martin Laštovička. "Enhancing cybersecurity skills by creating serious games." In Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, pp. 194-199. 2018. https://dl.acm.org/doi/10.1145/3197091.3197123

[Karagiannis20] Stylianos Karagiannis, Elpidoforos Maragkos-Belmpas, and Emmanouil Magkos: An Analysis and Evaluation of Open Source Capture the Flag Platforms as Cybersecurity e-Learning Tools. In IFIP World Conference on Information Security Education, pp. 61-77. Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-59291-2_5

[KYPO] Masaryk University. KYPO Cyber Range Platform. Online at https://crp.kypo.muni.cz/ [Accessed on January 20, 2022].

[Lability] Lability. Online at https://github.com/VirtualEngine/Lability [Accessed on January 20, 2022].

[Labtainers] Cynthia E. Irvine, Michael F. Thompson, Michael McCarrin, and Jean Khosalim. Live Lesson: Labtainers: A Docker-based Framework for Cybersecurity Labs. In *2017 USENIX Workshop on Advances in Security Education (ASE 17)*. 2017. https://www.usenix.org/conference/ase17/workshop-program/presentation/irvine

[RGCE] JAMK. Realistic Global Cyber Environment. Online at https://jyvsectec.fi/cyber-range/overview/ [Accessed on January 20, 2022]

[SIGCSEapg] Jan Vykopal, Valdemar Švábenský, Pavel Seda, and Pavel Čeleda. Preventing Cheating in Hands-on Lab Assignments. In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education (SIGCSE '22). New York, NY, USA: ACM, 2022. https://www.muni.cz/en/research/publications/1816366

[ThesisBelajova] Daniela Belajová. Persistent Software Vulnerabilities for Hands-on Cybersecurity Training. Brno, 2022. Available from: https://is.muni.cz/th/b4vd5/. Master's thesis. Masaryk University, Faculty of Informatics. Thesis supervisor: Pavel Čeleda.

[ThesisFarkas] Attila Farkas. Portable Virtual Cybersecurity Labs. Brno, 2021. Available from: https://is.muni.cz/th/oljh6/. Master's thesis. Masaryk University, Faculty of Informatics. Thesis supervisor: Jan Vykopal.

[ThesisHemalova] Lýdie Hemalová. Analysis of errors made by participants of cybersecurity training. Brno, 2021. Available from: https://is.muni.cz/th/mdf07/. Master's thesis. Masaryk University, Faculty of Informatics. Thesis supervisor: Valdemar Švábenský.

[ThesisKrajnikova] Eva Krajníková. Cybersecurity Games for Assessment of Skills. Brno, 2022. Available from: https://is.muni.cz/th/jap60/. Bachelor's thesis. Masaryk University, Faculty of Informatics. Thesis supervisor: Jan Vykopal.

[ThesisNagy] Imrich Nagy. Building open profiles of certified cryptographic devices. Brno, 2021. Available from: https://is.muni.cz/th/g7q67/. Master's thesis. Masaryk University, Faculty of Informatics. Thesis supervisor: Petr Švenda.

[ThesisPavelu] Lin Pavelů. Adaptive cybersecurity games. Brno, 2021. Available from: https://is.muni.cz/th/mnrr8/. Bachelor's thesis. Masaryk University, Faculty of Informatics. Thesis supervisor: Jan Vykopal.

[ThesisSvabensky] Valdemar Švábenský. Automated Feedback for Cybersecurity Training. Brno, 2022. Available from: https://is.muni.cz/th/dg3b4/. Doctoral thesis. Masaryk University, Faculty of Informatics. Thesis supervisor: Pavel Čeleda.

[ThesisVydra] Zdeněk Vydra. Automated Preparation of Virtual Images. Brno, 2021. Available from: https://is.muni.cz/th/pkevc/. Bachelor's thesis. Masaryk University, Faculty of Informatics. Thesis supervisor: Jan Vykopal.

[ThesisZidovsky] Patrik Židovský. Cyber Ranges. Brno, 2021. Available from: https://dspace.vutbr.cz/xmlui/handle/11012/190256. Bachelor's thesis. Brno University of Technology. Thesis supervisor: Jan Hajný.

[UNIS] UNIS. Supplier of complex investment units. Online at https://www.unis.cz/en [Accessed on January 20, 2022].

[Vagrant] HashiCorp. Vagrant. Online at https://www.vagrantup.com/ [Accessed on January 20, 2022].

[VagrantCloud] HashiCorp. Vagrant Cloud. Online at https://app.vagrantup.com/boxes/search [Accessed on January 20, 2022].

[VirtualBox] Oracle. VirtualBox. Online at https://www.virtualbox.org/ [Accessed on January 20, 2022].