TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Olesia Yaremenko 177240IVCM

SKILLS EVALUATION OF PARTICIPANTS OF CYBERSECURITY EXERCISES ON THE EXAMPLE OF A VIRTUAL HANDS-ON FORENSIC LAB

Master's thesis

Supervisor: Sten Mäses
MSc

Tallinn 2019

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Olesia Yaremenko 177240IVCM

KÜBERTURBEHARJUTUSTES OSALEJATE OSKUSTE HINDAMINE ARVUTIKRIMINALISTIKA-TEEMALISE VIRTUAALLABORI NÄITEL

Magistritöö

Juhendaja: Sten Mäses
MSc

Tallinn 2019

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Olesia Yaremenko

13.05.2019

Abstract

The aim of this thesis is to research possibilities for the use of user activity logging and analysis in cybersecurity exercises as an extensive source of data for detailed evaluation of skills. For this, a virtual lab with user activity monitoring elements was designed, implemented, and evaluated based on the ADDIE model.

The user activity log files gathered as a result of the virtual lab being completed, were analyzed to define if it is possible to make any conclusions based on the data in the logs, and whether the information can be used to create a profile of a user's skills. The research performed for the thesis shows that skills evaluation of participants of cybersecurity exercises is indeed possible through analysis of user activity data, with the condition that activity monitoring tools are precisely tuned to collect data according to the specifics of the tasks, lab objectives, and expectations as to what it is necessary to measure or monitor, when the exercise is designed.

This thesis is written in English and is 67 pages long, including 6 chapters, 37 figures and 9 tables.

Acknowledgements

I would like to express my gratitude to my supervisor Sten Mäses for his advice and guidance in this thesis work. I would also like to thank all those who taught me, inspired and contributed to my development throughout the years – there are too many names to count.

Last but not least, I would like to acknowledge the invaluable support and encouragement of my family and friends throughout not only this work but my whole life.

This would not have been possible if not for you.

List of abbreviations and terms

CIA triad	Confidentiality, integrity and availability
CPU	Central processing unit
FTP	File Transfer Protocol
i-Tee	Intelligent Training Exercise Environment
IP	Internet Protocol
NICE	National Initiative for Cybersecurity Education
NIST	National Institute of Standards and Technology
SQL	Structured Query Language
SSH	Secure Shell
TalTech	Tallinn University of Technology

Table of Contents

List of Figures

List of Tables

1 Introduction

Development of virtualized simulation environments allowed cybersecurity exercises and competitions to become an important part of cybersecurity education and training. However, in most cases, both participants and organizers pay attention just to the scores that are based on the fact of completing or not completing the tasks of the exercise. The potential of additional analysis of how exactly the tasks are approached and how the problems are solved remains mostly undiscovered.

This research demonstrates some of the possible usages of more extensive logging and analysis of information about the activity of participants of the exercises, that can provide insights about their skills and be a valuable source of data for more detailed evaluation. As an example of a cybersecurity exercise, a virtual hands-on forensic lab is used in the research.

1.1 Problem statement

The growing value of cyber exercises makes it important to gather as much useful information about the participants of the exercises as possible – while not being overly intrusive.

Currently, during an evaluation of participants of cybersecurity-related exercises, competitions, exams, conclusions about their skills are made based on the grades they are assigned if they complete the provided tasks or not. The methods used to perform a task are rarely tracked, and the fact that the same problem could sometimes be solved in multiple ways is ignored.

However, information about a participant's approach to a problem could be valuable for building a profile that would be able to reflect the skills, abilities, and knowledge of the participant.

So is it possible to track differences in approach to a task? If yes, then for what skills, how, and in what conditions? Is it possible to build a more detailed profile of the skills in comparison to what we can say about one's skills based on the information we usually gather now?

In spring of 2019, a course "Foundations of Cyber Security" targeted for computer science bachelor students took place in Tallinn University of Technology (TalTech). The course included virtual labs that were built on the i-Tee[1] distance laboratory system and covered various cybersecurity-related topics. For this research, an exercise lab focused on learning about confidentiality, integrity and availability (CIA triad) with the help of various forensic tools was implemented. The lab setup included elements that allowed to log user activity and prompted the users to forward these activity logs to a remote server for analysis before finishing the lab session.

The goal was to explore potential options for user activity logging and analyze what kind of information, if any, can be gathered to evaluate user's skills and detect issues the user could come across while working on the lab. The objective of the research was to design and implement a cyber security exercise that would allow user activity logging.

1.2 Motivation

Tracking differences in users' approaches to tasks and their solutions could be useful and used for multiple purposes. It could help to build a more detailed, clear profile of the person's skills, and the skillset evaluation is crucial as it could be used for such purposes as:

- evaluation of candidates for cybersecurity job positions;
- improvement of student selection during an admission process;
- creation of more efficient cybersecurity teams, in which skills of members would complement and balance each other;

[1] https://github.com/magavdraakon/i-tee

- self-assessment for those interested in evaluating their cybersecurity skills.

Skill-based evaluation can be introduced as a part of monitoring and advising system for education, on-the-fly guidance and help, understanding how well a participant is doing, etc.

1.3 Contribution

Research about evaluation methods and measurements of cybersecurity-related competencies and skills based on the analysis of logs and data about user activity in virtualized simulation environments is quite limited. In most cases, conclusions about the skills of participants are based on the fact of completion of the exercise – which, in the author's opinion, is a very limited and generalized way of skill evaluation.

As there has been little research done in regards to individual skills evaluation in cybersecurity that is based on environment-integrated measurements and analysis of logs and data about user activity in virtualized simulation environments, this is the gap that the author attempts to address.

The following subsections provide more details regarding the research contribution of the author.

1.3.1 Research design

The research follows the ADDIE instructional design model to identify instruction outcomes, define the scope and effective way of evaluation of the results of the virtual lab.

1.3.2 Research contribution

The contribution involves implementing a virtual lab with user activity logging functionality. The activity logs gathered in the process were afterwards analyzed to (1) evaluate certain skills of a user and identify possible issues the user could come across while completing the tasks, and (2) define which activity monitoring tools were most useful and what were their shortcomings.

1.3.3 Design evaluation

The lab was evaluated in two iterations. After collecting the results of the first group of students, their results were analyzed to define if it is possible to implement any changes in the lab that would allow to collect more detailed activity logs. However, the results showed that there were no improvements to be made in the available time – the main issue was that some participants in the first group did not submit the activity logs as instructed due to inattentiveness or technical issues, which resulted in incomplete or missing logs.

All the activity logs were afterwards analyzed to define which tools and commands the participants were using, what tasks they needed assistance with, etc.

1.4 Limitations

There are certain limitations and assumptions to the research. One of the limitations that is necessary to take into consideration is time restrictions. It was not possible to base the research on any already existing datasets and activity logs, as there has not been any openly available previous research done that could generate such data with enough information to make it possible for the author to perform analysis without making a lot of assumptions. Design and implementation of a new virtual lab took approximately three months. Also, the virtual lab could only be given to the students of the previously mentioned course, significantly limiting the target group. Due to this, it was not possible to have multiple iterations with several control groups, involving different activity logging tools and techniques.

The specifics of the target group also put limitations onto the content of the lab – the content of the tasks needed to approximately correspond to the level of knowledge one can expect from a bachelor's level student.

There are also certain limitations connected to the lab environment. It was not possible to use virtual machines with Windows operating system or any proprietary scripts, which limited possible scenarios and tools that could be used in the lab development.

The forwarding of activity logs to a remote server was not fully automated and required additional actions from the users, so if the user forgot to perform the necessary action or in case of technical issues with the environment, the activity logs could be lost or incomplete. There is always a possibility that the logging tools may fail or be not as precise as necessary, which may make it difficult or even impossible to make any conclusions. Also, evaluation of certain skills (for example, non-technical skills, especially ones that can be demonstrated in a free form and may be open to multiple interpretations depending on the specifics of a case, such as skills related to cybersecurity law) may be impossible or very difficult to implement in virtualized environments, and for them, alternative evaluation methods have to be used. However, such skills are out of the scope of the research.

There is no universal, generally agreed-on framework to cover all cybersecurity-related skills. So, this study covers only a limited set of skills selected by the author. However, it is assumed that the same principles can be applicable to a wider set of skills.

1.5 Thesis outline

This thesis is divided into six chapters. Chapter one covers the overview of the thesis including the problem statement, proposed solution and the author contribution in this research. Chapter two presents literature background necessary for the research. In Chapter three, the capabilities of the used user activity monitoring tools are discussed. Chapter four contains the research design methodology including the analysis, design, development, implementation and evaluation phases of the lab. Chapter five covers suggestions for future work. The last chapter contains the summary of the performed research.

2 Related literature

The existing literature on the topic of skills evaluation of participants of cybersecurity exercises, especially through analysis of logs of their activities, is limited. The existing research in the field revolves around measuring or improving learning effectiveness of cybersecurity exercises (mostly large-scale team exercises and group learning, but not individual) [1]–[4] or around designing and implementing virtualized learning environments [5] or cybersecurity competitions [6].

M. Granåsen and D. Andersson [1] concentrate on measuring team effectiveness and working with big datasets from different sources, while totally omitting individual learning evaluation. Research of K. Maennel [2] is mostly dedicated to metrics connected to learning effectiveness of teams as well, but there are several references to sub-team and individual metrics. A. Malviya et al. [4] cover in detail the topic of situational awareness in teams and its relation to performance and the aspect of data collection in cybersecurity competitions that involves both gathering data from various devices and asking the participants various questions. Overall, even though the research mentioned above is targeted towards teams, some general concepts and ideas (for example, about gathering data from various devices and tools) can also be applicable for the research targeted towards individual users.

Due to the limited amount of papers related to this particular topic, the author also looked into literature in the related fields of study. In the process of review of the papers related to profiling of cyber attackers, some of the ideas given in the papers [7], [8] (in particular, about data collection and measurements) also provided the author with some inspiration and are at least partially applicable to the topic the author plans to address in this research.

In particular, J. Brynielsson et al. [7] in their research talked about designing a cyber defence exercise that can be used to obtain data for construction of attacker personas as a way of profiling the mentioned attacker. S. Kapetanakis et al. [8] suggested to profile cyber attackers via use of case-based reasoning instead of focusing on characteristics of an attack, building an attacker's profile according to the information about various attributes and patterns of the attacks received from real vulnerable systems.

Research papers about obtaining better metrics in virtualized environments [9] and evaluating skills based on computer simulations [10] have been most closely related to the author's topic of choice and provided valuable insights that will be mentioned in the next chapter.

Work of Jonathan McClain et al. [11] discusses factors impacting human performance in forensic analysis and data collection with various measures on questionnaires and human-machine transactions. The authors conclude that main difference between forensic analysts of different levels of experience lies in their use of various software applications.

In the paper dedicated to log analysis in cybersecurity training exercises by Robert G. Abbott et al. [12], infrastructure and techniques for performance data collection and mining data logs for relevant performance variables with the use of specialized software tool able to capture human-machine transactions were discussed. It was possible to identify blocks of activity of users and several tools that the participants were using.

Both of these works [11], [12], however, use Windows operating system non-virtualized environment as a basis for their research. These papers also analyze the usage of corresponding software tools by the participants. There is no information about similar research being done on the basis of a Linux-based operating system and in a virtualized environment. Another thing to consider is that the research was targeting analysts with some level of experience, and part of the tasks that the participants were requested to complete was not individual, but required teamwork.

For this research, the tasks of the hands-on forensic lab were developed on the basis of selected abilities, skills, and knowledge areas that a person working as a cyber defence forensics analyst[1] is expected to have, according to National Initiative for

[1] https://www.nist.gov/itl/applied-cybersecurity/nice/resources/nice-cybersecurity-workforce-framework

Cybersecurity Education (NICE) Cybersecurity Workforce Framework [13] of National Institute of Standards and Technology (NIST). One of the purposes of this framework is to help to identify and develop cybersecurity talent by describing and categorizing cybersecurity-related work [14].

3 Background

In this chapter, the tools that are used for user activity monitoring in the lab are discussed. The topic of metrics and various things to consider are also covered.

3.1 Metrics

Generally, the measurements necessary to determine skill levels and competencies depend on the specific skills, and skill level could be defined by comparing (1) how close a participant's results are to the objectives defined by the organizer of the cyber exercise, and (2) how the results of participants differ among each other while taking the best result as an absolute.

In cases when a task may be completed, for example, with the use of various tools, such differences should be recorded and analyzed, as these differences in participants' approach to the problem may provide information about them having certain skills and competencies (such as knowledge of specific programming language, tool, process, etc.), or ability to think in a creative or unusual way.

All this information may be helpful to provide a clearer understanding of a participant's skills.

After the exercise is completed, log analysis should be performed.

To gather the necessary data for the analysis, logging of such information about the environment and user activity as running processes, network traffic, performed commands, search queries, etc., has to be performed. The measurements have to be both qualitative and quantitative. Metrics can be based on (1) direct input of a participant, (2) time, (3) tools of choice, (4) automated scripts, and (5) string similarity [9, Ch.4].

As exercise scenarios and expectations of organizers about skill levels vary from case to case, the necessary measurements would also differ depending on the objectives of the exercise. However, it should be possible to establish general rules of user activity monitoring for the evaluation of cybersecurity-related skills as a part of the research.

3.2 Other considerations

As was mentioned earlier, skillset – as well as the level of skills of the participants – covered by an exercise would vary from case to case.

Research shows that currently there is no universally approved, standardized framework to be used to map cybersecurity-related skills – various frameworks cover various areas of knowledge and disciplines to a different extent and are also not detailed enough to go beyond general descriptions of expected abilities, competencies, and topics [15]. However, for the purpose of skills evaluation, the more information about skills and competencies that are expected of participants is available, the better. It will help to get a better understanding of what should be tracked and measured, and the results of the log analysis would be more specific, if necessary. At the same time, grouping the skills into broader skillsets allows to provide a more generalized overview of the participant's areas of knowledge [9]. So, depending on the purpose of the skill profile and broadness of the area of knowledge, skills in which are being evaluated, it may be useful to vary the scale of detail in the profile. If the exercise covers a very specific, narrow field, then higher granularity of skills profile may be more beneficial than a generalized overview.

Generally, steps of Stenmap process [10, Ch. 4.2] should be applicable for connecting tasks in a cyber exercise to the skillsets and establishing what measurements are necessary. Qualitative measurements during and after the exercise should consist of behavioral performance assessment of the participants, while quantitative measurements should consist of behavioral performance analysis, tracking and logging, network monitoring during the exercise with the following analysis afterward [16].

For the purpose of skills evaluation in virtualized environments, measurements have to be integrated into the system, as questionnaires will be distracting and most likely too long for the participants to fill in, as they will have to cover many details while still not being the most efficient way to evaluate the participant's cybersecurity-related skills. Also, results of the questionnaires may be quite subjective if the questions are asked incorrectly or in case the participant does not have a clear understanding of their own competencies. So, questionnaires may only be used as a way to validate the results of the study, but not be a part of the final method of evaluation.

3.3 User activity monitoring

In this section, various user activity monitoring tools and commands, their specifics and limitations are covered.

3.3.1 Keylogging

Although multiple options are available for keylogging on Linux systems, logkeys[1] keylogger seems to be the most stable and easy-to-install [17], [18]. Even though some features that could be useful for more precise user activity monitoring (such as logging mouse click events and clipboard content) have not been implemented, the keylogger logs all the keystrokes and contains timestamps.

By default, the log file `/var/log/logkeys.log` is used. It is necessary to indicate the correct language keymap when starting the keylogger, as otherwise the output in the log would contain wrong characters.

3.3.2 Bash history

Bash history file contains the list of last commands performed by a user in Bash shell. So, it can provide useful information about user activity, making it possible to see which command-line tools a user used. Analysis of the content of the file can also help to determine which tools or commands the student is familiar or struggling with.

For the virtual lab, it was decided to gather data from the Bash history file of the student account in the desktop virtual machine. The location of the file is `/home/student/.bash_history`, and it is expected that any console commands a student might perform to investigate on their own virtual desktop or to run console-based tools would be visible in it.

[1] https://github.com/kernc/logkeys

However, it is important to note that while running, Bash maintains the list in memory, and the history file is updated with the commands on exit. So, it is important to exit Bash (close all open console windows) before attempting to gather the data from the file. Otherwise, the list of the commands will be out of date.

3.3.3 Accounting Utilities

To monitor user activity on a Linux system, it is possible to use `acct` accounting utilities [19], [20].

Certain commands associated with these utilities may make it possible to gather information about the commands and applications that were recently executed. Command `ac -p` outputs statistics about connect time for each user on the system; command `lastcomm student` lists last executed commands for the student account; command `sa -m` summarizes the information about previously executed commands by printing out the number of processes and the number of CPU minutes for each user; command `sa -u` prints out userid and the name of the command for each command in the accounting file [19].

3.3.4 Linux Auditing system

The Linux Auditing system contains `auditd` daemon, that is responsible for writing system audit records to the disc, while `aureport` utility allows to review the summary of audit logs. The summary report includes information about the number of changes in the system, logins, authentications, processes, etc.

There are multiple options to gather information about specific events, but as it is not known what information in the audit logs may be useful for skills evaluation, only general summary report data will be gathered in the developed virtual lab; the purpose would be to see if any interesting conclusions can be made only on the basis of numbers in these reports and not other utilities used for Linux auditing [21].

3.3.5 Legal and ethical considerations

There are several potential constraints that should be considered in regards to the content of the developed virtual lab and user activity monitoring.

It is possible that some of the tools and techniques used in the lab (for example, usage of network monitoring software or password-cracking tools) may be used with malicious purposes outside of the study environment. It is assumed that any knowledge the students may obtain while working on the lab would not be used unethically.

Also, lab users should be able to decide whether they want to submit their activity logs or not – this aspect was addressed during lab development, and the log files are not submitted automatically. In case a lab user does not consent to submit the log, no information about this user's activity would be retained besides the data about flag submissions in the database. It is important to note that the tasks of the developed lab do not require the students to provide such confidential data as personal account credentials, contact details, etc. However, in general, if for task completion a user would require to type in any passwords or other confidential information, appropriate measures should be taken to enforce the safety of such data if it is logged, or mechanisms that would remove the information from logs or obfuscate it have to be implemented.

4 Methodology

When creating a virtual lab for teaching, a specific methodology should be used for development to ensure the high quality of the learning experience. The background of the audience, prerequisites and learning objectives have to be clearly defined to ensure that all the goals are met. Instructional design is used to develop learning experiences and environments that make it possible for the students to acquire certain skills and knowledge [22].

For this research, the widespread ADDIE instructional design model was chosen. Its phases provide a guideline for the creation of effective learning tools [23], [24].

The following subsections of the chapter cover the preparation and phases of the ADDIE model.

4.1 Preparation

During the preparation phase it was decided to develop the lab on the i-Tee [25] platform with the use of the following programming languages: PHP, Ruby, Bash, and SQL. The source code of the lab was published in the GitLab repository of TalTech[1].

4.1.1 Metrics and user activity logging

The lab was designed as a flag-based lab, which requires the students to submit answers in a certain format to complete the task. The information about flag submission, including the type of the flag and time of the submission was gathered in a SQL database.

[1] https://gitlab.cs.ttu.ee/vosa/cybersec/ITI0103/forensic

At the same time, user activity monitoring tools including, but not limited to, a keylogger, were running in the background. The gathered information about the activity could be sent by the user to a remote server by a press of a button on the lab page.

For this work, multiple sources of information about user activity were used to later determine the capabilities and limitations of different activity monitoring and logging tools.

4.2 Analysis phase

The analysis phase is covered in this section, where objectives, the learning environment and the target group analysis including pre-requirements are defined [14], [15].

After the discussion with the lecturer of the course, the following facts regarding target audience, learning environment and objectives of the lab were defined:

1. The virtual lab should be developed in the i-Tee system.
2. Free and open-source software should be used.
3. The developer should have the technical knowledge necessary to resolve issues with the system, as it is not actively supported.
4. The environment is using a virtualized Linux operating system, so the knowledge of Linux administration is needed.
5. A remote server for activity log gathering needs to be set up.
6. Target audience consists of bachelor students of TalTech that are studying computer science.
7. The students are to learn and test their knowledge about basic cybersecurity topics in a practical way.
8. While all the students have basic knowledge about some cybersecurity topics, some of them may be more advanced.
9. The tasks of the lab should be suitable for all users in their complexity.
10. The flags in the lab should be randomized when possible, to avoid cheating.

4.3 Design phase

In this section, tasks and content are planned, the focus is on the design of a prototype [23], [24].

It was decided that the lab tasks will focus on familiarizing students with the principles of confidentiality, integrity, and availability in information security [26] by making them investigate on the details of simulated attacks that targeted these three principles, mostly with the use of various forensic tools and commands.

Confidentiality principle revolves around establishing appropriate levels of access to information and managing systems to enforce the rules regarding it, accordingly, by setting up correct file permissions, strong passwords, system control lists, etc. [27].

Key to the integrity of the data is to protect it from being modified or deleted by unauthorized parties and ensure that the changes may be reversible if needed – especially if the data is sensitive [27].

Last, but not least, systems must work properly and be available to provide and protect information when needed [27].

There are three modules in the lab, each of them dedicated to one of the principles of CIA triad. To cover the Confidentiality principle, one of the attack scenarios simulated a brute-force attack on a server over SSH in the first phase. During the second phase of the attack, one of the files on the compromised server was modified by the attacker that successfully brute-forced the credentials on the system. This phase covered Integrity principle of the CIA triad. For an example related to Availability principle, second attack simulated various types of flood attacks onto the server.

These attack scenarios were chosen as they could be set up in a way that would require students to type in a significant part of the commands needed to complete the tasks, making it possible to capture a significant amount of information about their activity. Additionally, for the same purpose, the students were requested to perform searches related to the tasks in the browsers inside the virtual environment.

For the Confidentiality module, the students are expected to submit the username of the successfully brute-forced account, that can be found out by accessing the server and reviewing the information about successful SSH login attempts in the authentication log file of the machine.

For the Integrity module, it is necessary to locate the file that was modified by the attacker, discover that it is a password-protected compressed file archive, and crack the password on the archive to extract the file containing a unique flag in the form of a hash.

Availability module task requires students to monitor and analyze the network traffic on the server machine to match seven attacker Internet Protocol (IP) addresses to the types of attacks, based on such information as request types, protocols used, etc. The simulated attack types are unique for each attacker IP, making it possible to submit up to seven different correct flags for the module.

Based on these tasks, the lab would cover certain aspects of the following skill areas:

- Linux log analysis
- Basic file analysis
- Password cracking
- Network traffic capture
- Network traffic analysis

Depending on the flag submissions and information from user activity logs, it should be possible to identify whether the student shows awareness or knowledge about:

- Linux authentication log structure
- Password cracking methods (brute-force, wordlist-based, etc.)
- Basic file properties (type, extension, encoding, etc.)
- Types of flood attacks

In this case, awareness would mean that the student only knows something regarding a topic and researched on it, but is not able to solve the task covering this topic. Knowledge would mean that the user was able to successfully apply the information acquired from performed research to use the necessary tools or commands to resolve the task related to the topic. There is no strict list of the tools and commands that are expected to be used to complete the lab, as part of the research is to track differences in approach to the tasks – including usage of different tools by the students to achieve the same result. However, some suggestions for the commands and tools that may be used are provided in the general lab solution guideline.

More specific mapping of modules and tasks they consist of to some of the knowledge, tasks, skills, and abilities of a cyber defence forensics analyst according to the requirements given in NIST NICE Framework[1] is shown in Table 1, Table 2, and Figure 1. In cases where the requirements mentioned multiple operating systems, only Linux environment should be taken into account for this research. The mapping is used for evaluation of the participants after analysis of collected activity logs.

Table 1: NIST NICE Framework requirements for a forensics analyst[2], non-specific to lab modules

Forensic lab (requirements, non-specific to lab modules)
A0043: Ability to conduct forensic analyses in and for both Windows and Unix/Linux environments.
K0077: Knowledge of server and client operating systems.
K0119: Knowledge of hacking methodologies.
K0224: Knowledge of system administration concepts for operating systems such as but not limited to Unix/Linux, IOS, Android, and Windows operating systems.
T0027: Conduct analysis of log files, evidence, and other information to determine best methods for identifying the perpetrator(s) of a network intrusion.
T0286: Perform file system forensic analysis.
S0065: Skill in identifying and extracting data of forensic interest in diverse media (i.e., media forensics).

[1] https://niccs.us-cert.gov/workforce-development/cyber-security-workforce-framework/workroles?name_selective=All&fwid=IN-FOR-002

[2] https://niccs.us-cert.gov/workforce-development/cyber-security-workforce-framework/workroles?name_selective=All&fwid=IN-FOR-002

Table 2: NIST NICE Framework requirements for a forensics analyst[3], module-specific

Confidentiality module	**Integrity module**	**Availability module**
K0132: Knowledge of which system files (e.g., log files, registry files, configuration files) contain relevant information and where to find those system files.	K0187: Knowledge of file type abuse by adversaries for anomalous behavior.	K0255: Knowledge of network architecture concepts including topology, protocols, and components.
K0179: Knowledge of network security architecture concepts including topology, protocols, components, and principles (e.g., application of defense-in-depth).	T0398: Perform file and registry monitoring on the running system after identifying intrusion via dynamic analysis.	K0301: Knowledge of packet-level analysis using appropriate tools (e.g., Wireshark, tcpdump).
T0036: Confirm what is known about an intrusion and discover new information, if possible, after identifying intrusion via dynamic analysis.	T0167: Perform file signature analysis.	T0240: Capture and analyze network traffic associated with malicious activities using network monitoring tools.
S0067: Skill in identifying, modifying, and manipulating applicable system components within Windows, Unix, or Linux (e.g., passwords, user accounts, files).	S0092: Skill in identifying obfuscation techniques.	S0156: Skill in performing packet-level analysis.

[3] https://niccs.us-cert.gov/workforce-development/cyber-security-workforce-framework/workroles?name_selective=All&fwid=IN-FOR-002

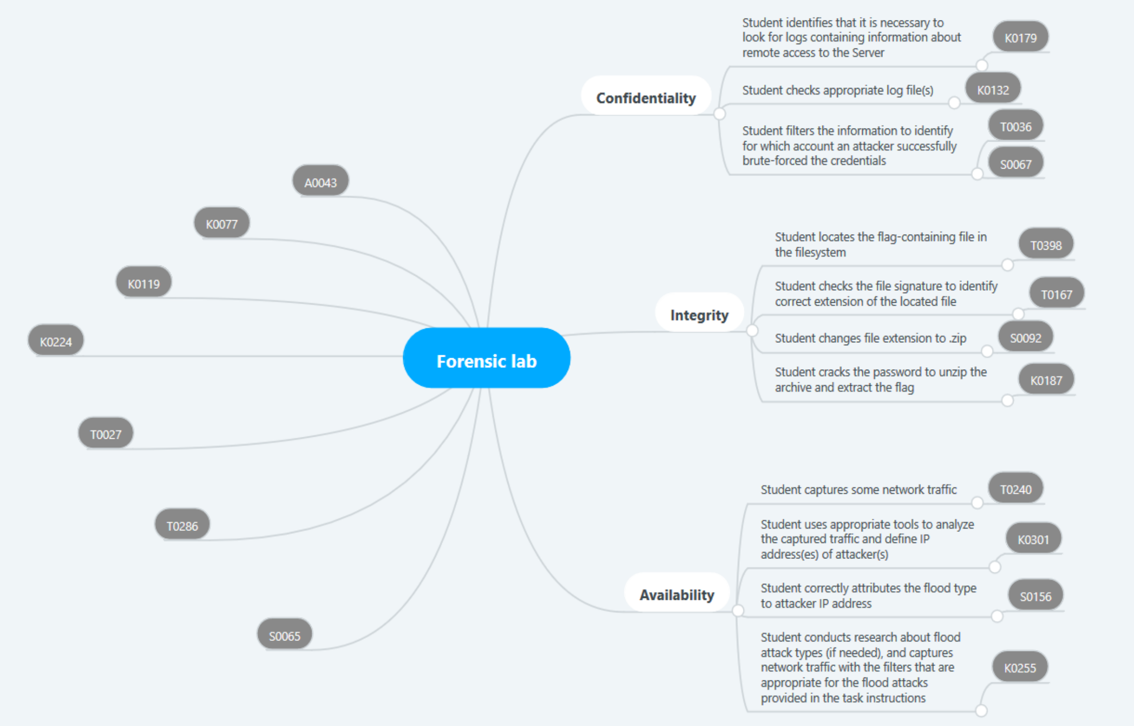

Forensic lab
A0043
K0077
K0119
K0224
T0027
T0286
S0065
Confidentiality
Student identifies that it is necessary to look for logs containing information about remote access to the Server
K0179
Student checks appropriate log file(s)
K0132
Student filters the information to identify for which account an attacker successfully brute-forced the credentials
T0036
S0067
Integrity
Student locates the flag-containing file in the filesystem
T0398
Student checks the file signature to identify correct extension of the located file
T0167
Student changes file extension to .zip
S0092
Student cracks the password to unzip the archive and extract the flag
K0187
Availability
Student captures some network traffic
T0240
Student uses appropriate tools to analyze the captured traffic and define IP address(es) of attacker(s)
K0301
Student correctly attributes the flood type to attacker IP address
S0156
Student conducts research about flood attack types (if needed), and captures network traffic with the filters that are appropriate for the flood attacks provided in the task instructions
K0255

Figure 1: NIST NICE Framework requirements for a forensics analyst, mapped against activities that are expected from lab participants

4.4 Development phase

This section covers the actual creation of lab content, debugging and revision [23], [24].

As mentioned earlier, the complete source code of the lab can be found on the GitLab repository of TalTech[1].

In the following subsections specific aspects of lab development, that can be important for the analysis of student activity will be discussed.

4.4.1 Confidentiality module

For the brute-force attack over SSH, it was necessary to create a vulnerable account with a weak password on the server (see Figure 2 for the commands used in the lab script to create the account), then simulate network scan and brute-force attack from the attacker IP to the discovered server IP (see Figure 3 for the commands).

[1] https://gitlab.cs.ttu.ee/vosa/cybersec/ITI0103/forensic

```
sudo adduser user --gecos "" --disabled-password
echo "user:melissa" | sudo chpasswd
```

Figure 2: Commands used for vulnerable account creation on the server

```
apt install nmap -y
nmap -sP 192.168.8.0/24
nmap 192.168.8.6
nmap -sV -p 22 --script ssh-brute 192.168.8.6
```

Figure 3: Commands used for simulation of brute-force attack over SSH

As can be seen from Figure 3, nmap[1] tool was used to simulate the first phase of the attack scenario. This setup also allows students to see live traffic coming from the attacker if certain commands are run on the server machine while the attack script is running.

4.4.2 Integrity module

To simulate the second phase of the attack scenario, the sshpass[2] tool was installed and used to access the server and pass the commands to create a password-protected compressed file archive with a flag that is unique for each user, inside, and change the extension of the file (refer to Figure 5 for the commands used).

To make the task more complicated, the password for the archive is selected from a wordlist[3], that is downloaded onto the machine during lab setup (see Figure 4), and then removed after the archive creation.

```
wget -O /home/user/.w.txt
https://github.com/brannondorsey/naive-hashcat/releases/download/data/
rockyou.txt
touch .onew.txt
shuf -n 1 /home/user/.w.txt | sudo tee /home/user/.onew.txt
```

Figure 4: Commands used to select a password from a wordlist

```
apt install sshpass -y
sshpass -p melissa ssh -o "StrictHostKeyChecking=no" user@192.168.8.6 "cd
/home/user; touch secretcode.txt; echo -n $(sudo dmidecode -s bios-release-
date)integrity | md5sum | cut -c -32 | tee secretcode.txt; zip -P \$
(cat .onew.txt | tr -d "\n") secretcode.zip secretcode.txt; rm
secretcode.txt; rm .onew.txt; rm .w.txt; cp secretcode.zip
your_secret_code_is_here.txt; rm secretcode.zip"
```

Figure 5: Commands used to create the task with the flag for Integrity module

The flag is an MD5 hash of the student's unique username, concatenated with the word "integrity", which makes it individual for each user performing the lab.

4.4.3 Availability module

To simulate the flood attacks in the second scenario, hping3[1] tool was used. To avoid creation of multiple attacker machines and use less resources, the attacker IP addresses were spoofed, making it possible to generate the traffic showing seven different attacker IP addresses performing various types of flood attacks, instead of only one IP address. To avoid actual denial of service on the attacked server, the speed of the attacks was significantly slowed down. The commands used for the simulation of the attack are shown in Figure 6.

```
apt install nmap -y
nmap -sP 192.168.8.0/24
nmap 192.168.8.6
apt install hping3 -y
sudo hping3 -i 1 -1 --spoof 192.168.8.91 192.168.8.6 & sudo hping3 -i 1 -1 -C
3 -K 3 --spoof 192.168.8.92  192.168.8.6 & sudo hping3 -i 1 -S -d 160 --spoof
192.168.8.93 -p 80 192.168.8.6 & sudo hping3 -i 1 -d 140 --spoof 192.168.8.94
-p 80 -A 192.168.8.6 & sudo hping3 -i 1 -d 160 --spoof 192.168.8.95 -p 80 -R
192.168.8.6 & sudo hping3 -i 1 -d 170 --spoof 192.168.8.96 -p 80 -F -S -R -P
-A -U -X -Y 192.168.8.6 & sudo hping3 -i 1 --spoof 192.168.8.97 --udp --sign
150 -p 80 192.168.8.6
```

Figure 6: Commands used for simulation of the attack for Availability module

4.4.4 User activity logging

To log information about user activity, such tools as acct[2] tool, auditd[3] daemon, and logkeys[4] keylogger were installed on the desktop virtual machine, from which the students access the server to investigate on the attacks. It is also the same machine through which the students submit the flags and perform lab-related browsing of the Internet.

Figure 7 shows the commands used to install activity logging tools and configure keylogger.

[1] https://linux.die.net/man/8/hping3

[2] https://www.gnu.org/software/acct/#TOCintroduction

[3] https://linux.die.net/man/8/auditd

[4] https://github.com/kernc/logkeys

```
apt-get install acct
sudo apt install auditd -y
wget http://launchpadlibrarian.net/165012984/logkeys_0.1.1a+git5ef6b0dcb9e3-
2_amd64.deb
sudo dpkg -i logkeys_0.1.1a+git5ef6b0dcb9e3-2_amd64.deb
rm logkeys_0.1.1a+git5ef6b0dcb9e3-2_amd64.deb
wget https://raw.githubusercontent.com/kernc/logkeys/master/keymaps/en_GB.map
sudo cp en_GB.map /etc/en_GB.map
rm en_GB.map
sudo logkeys --start -m /etc/en_GB.map
```

Figure 7: Commands used to install activity logging tools and configure keylogger

4.4.5 User activity log gathering and submission

To gather the logs about user activity from the tools mentioned above, a separate folder is created on the virtual desktop, in which files containing output of log gathering tools are gathered once a student decided to submit the activity logs. After all the log files are created, they are transferred to a remote server. Each filename contains timestamp indicating the date and time of its creation, the username of the student submitting the logs, and abbreviation of the command used to obtain information about user activity.

As can be seen from Figure 8, besides the acct tool, auditd daemon, and logkeys keylogger, another source of information is the bash history file of the student account on the virtual desktop machine.

No automatic log submission has been implemented, as for it the credentials for the FTP server user have to be saved on the virtual desktop machine, making it possible for a student to find the credentials and use them to access the server and make changes to files, delete them or see how other participants were solving the tasks.

```
sudo mkdir /var/klog
sudo cat /var/log/logkeys.log | sudo tee /var/klog/$(sudo dmidecode -s bios-
release-date)final_klog$(date +%Y.%m.%d.%H.%M.%S)UTC.txt
sudo cat /home/student/.bash_history | sudo tee /var/klog/$(sudo dmidecode -s
bios-release-date)final_tbh_log$(date +%Y.%m.%d.%H.%M.%S)UTC.txt
ac -p | sudo tee /var/klog/$(sudo dmidecode -s bios-release-date)final_ac-
p_log$(date +%Y.%m.%d.%H.%M.%S)UTC.txt
lastcomm student | sudo tee /var/klog/$(sudo dmidecode -s bios-release-
date)final_lastcomm_student_log$(date +%Y.%m.%d.%H.%M.%S)UTC.txt
sa -m | sudo tee /var/klog/$(sudo dmidecode -s bios-release-date)final_sa-
m_log$(date +%Y.%m.%d.%H.%M.%S)UTC.txt
sa -u | grep "student" | sudo tee /var/klog/$(sudo dmidecode -s bios-release-
date)final_sa-u_student_log$(date +%Y.%m.%d.%H.%M.%S)UTC.txt
sudo aureport | sudo tee /var/klog/$(sudo dmidecode -s bios-release-
date)final_aureport_log$(date +%Y.%m.%d.%H.%M.%S)UTC.txt
sudo find /var/klog/ -type f -exec curl -u
remote_server_username:remote_server_password -T {}
ftp://172.16.0.17/final/{} --ftp-create-dirs \;
```

Figure 8: User activity log gathering and submission commands

As an alternative, any other remote FTP server can be configured and provided in the script as a backup option in addition to the one mentioned in Figure 8.

4.5 Implementation phase

According to the ADDIE model, this phase deals with training of facilitators and students, but due to the specifics of the learning environment, in this case the section covers network infrastructure of the virtual lab [23].

The simplified lab network diagram can be seen in Figure 9. Virtual machine with SQL server contains a database for gathering information about flag submissions. FTP server is used for user activity log gathering.

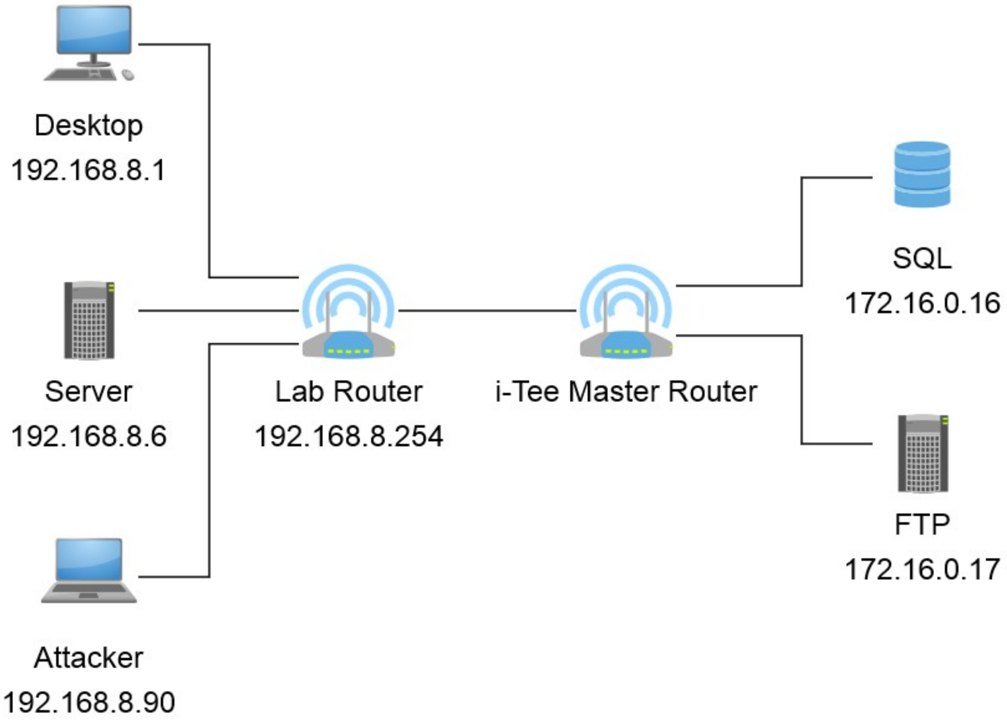
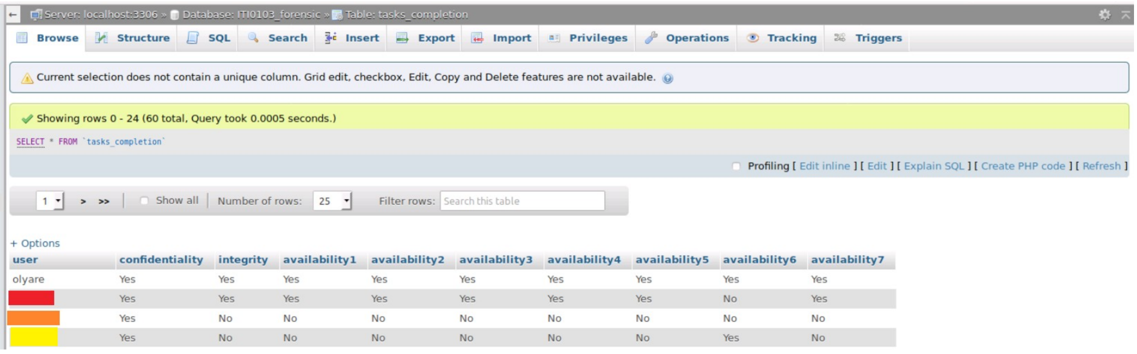


Figure 9: Lab network diagram

The primary virtual machine students get access to is Desktop. From there they are instructed to connect to Server using provided credentials. All attacks are coming from the Attacker machine. Lab Router is necessary for lab configuration and network access.

Information about flag submission by the user is sent to a database at the IP address 172.16.0.16, and user activity log files are sent to the server at ftp://172.16.0.17 address.

Figure 10 gives an example of contents of 'tasks_completion' database table and its structure. 'Yes' values mean that the corresponding flag submitted by the user was correct. In case no correct flags were submitted by a student, no record is created in this table for the student's username.



Server: localhost:3306 » Database: ITI0103_forensic » Table: tasks_completion

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 24 (60 total, Query took 0.0005 seconds.)

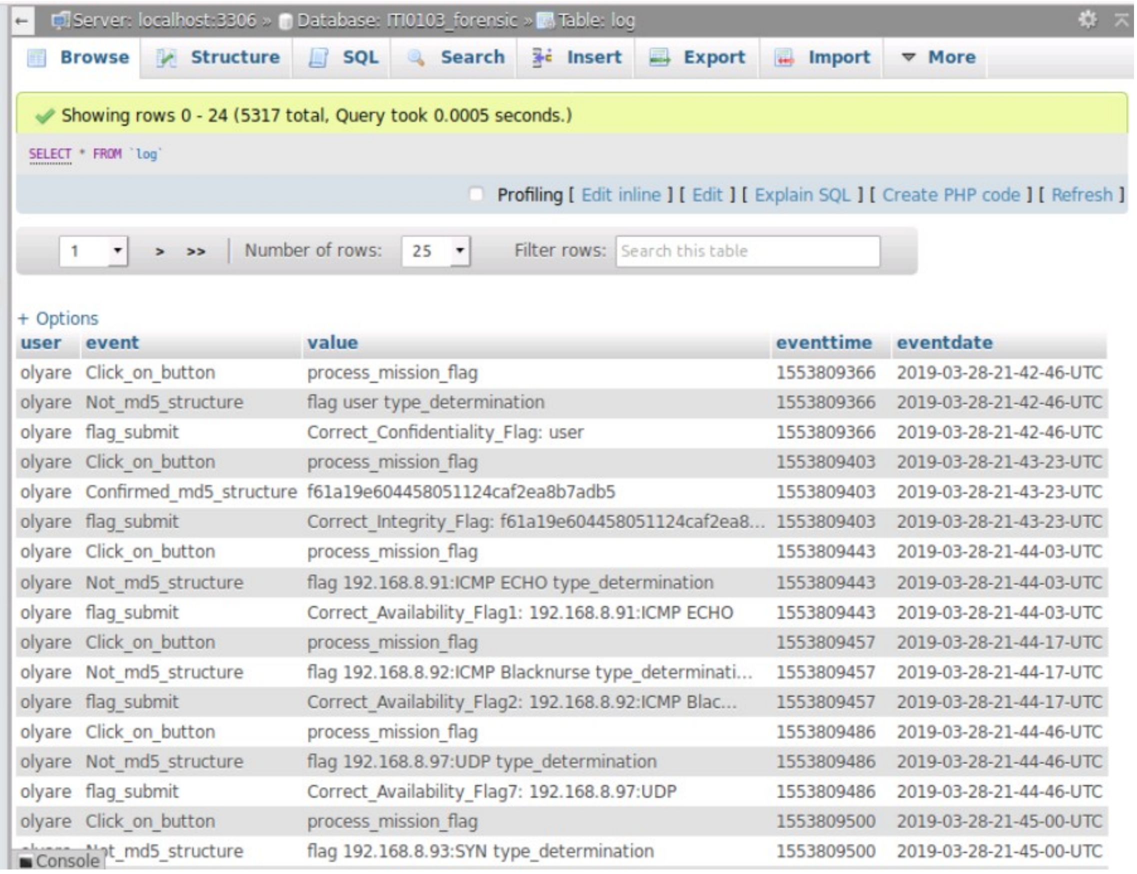
```
SELECT * FROM `tasks_completion`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1 > >> □ Show all
Number of rows: 25
Filter rows: Search this table
+ Options

user	confidentiality	integrity	availability1	availability2	availability3	availability4	availability5	availability6	availability7
olyare	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
[redacted]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
[redacted]	Yes	No	No	No	No	No	No	No	No
[redacted]	Yes	No	No	No	No	No	No	Yes	No

Figure 10: Example of ‘tasks_completion’ database table

Figure 11 demonstrates the structure and some of the possible contents of ‘log’ database table. In this table, information about any flag submission attempts is saved, along with the time when the flag was submitted.



Server: localhost:3306 » Database: ITI0103_forensic » Table: log

Browse
Structure
SQL
Search
Insert
Export
Import
More

✓ Showing rows 0 - 24 (5317 total, Query took 0.0005 seconds.)

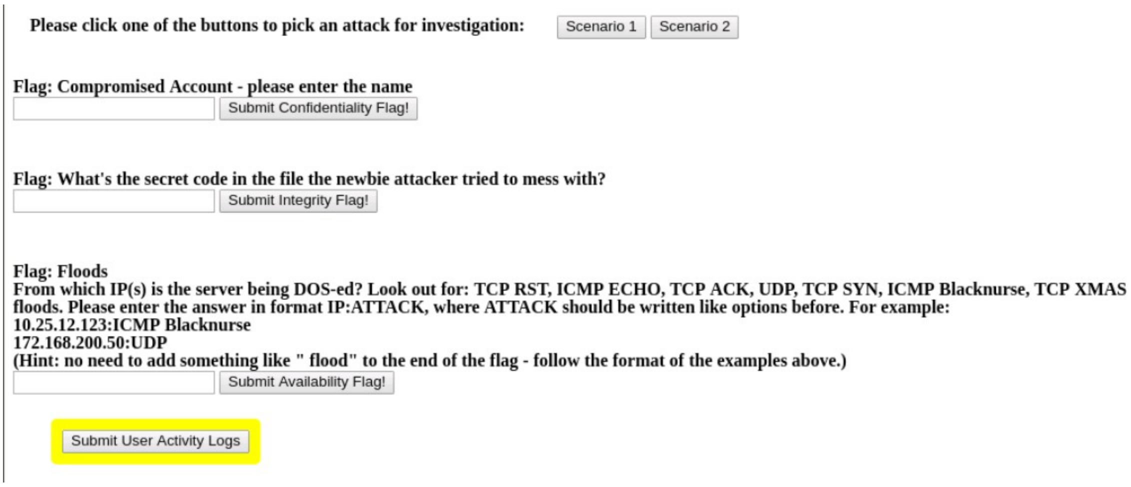
```
SELECT * FROM `log`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1 > >>
Number of rows: 25
Filter rows: Search this table
+ Options

user	event	value	eventtime	eventdate
olyare	Click_on_button	process_mission_flag	1553809366	2019-03-28-21-42-46-UTC
olyare	Not_md5_structure	flag user type_determination	1553809366	2019-03-28-21-42-46-UTC
olyare	flag_submit	Correct_Confidentiality_Flag: user	1553809366	2019-03-28-21-42-46-UTC
olyare	Click_on_button	process_mission_flag	1553809403	2019-03-28-21-43-23-UTC
olyare	Confirmed_md5_structure	f61a19e604458051124caf2ea8b7adb5	1553809403	2019-03-28-21-43-23-UTC
olyare	flag_submit	Correct_Integrity_Flag: f61a19e604458051124caf2ea8...	1553809403	2019-03-28-21-43-23-UTC
olyare	Click_on_button	process_mission_flag	1553809443	2019-03-28-21-44-03-UTC
olyare	Not_md5_structure	flag 192.168.8.91:ICMP ECHO type_determination	1553809443	2019-03-28-21-44-03-UTC
olyare	flag_submit	Correct_Availability_Flag1: 192.168.8.91:ICMP ECHO	1553809443	2019-03-28-21-44-03-UTC
olyare	Click_on_button	process_mission_flag	1553809457	2019-03-28-21-44-17-UTC
olyare	Not_md5_structure	flag 192.168.8.92:ICMP Blacknurse type_determinati...	1553809457	2019-03-28-21-44-17-UTC
olyare	flag_submit	Correct_Availability_Flag2: 192.168.8.92:ICMP Blac...	1553809457	2019-03-28-21-44-17-UTC
olyare	Click_on_button	process_mission_flag	1553809486	2019-03-28-21-44-46-UTC
olyare	Not_md5_structure	flag 192.168.8.97:UDP type_determination	1553809486	2019-03-28-21-44-46-UTC
olyare	flag_submit	Correct_Availability_Flag7: 192.168.8.97:UDP	1553809486	2019-03-28-21-44-46-UTC
olyare	Click_on_button	process_mission_flag	1553809500	2019-03-28-21-45-00-UTC
olyare	Not_md5_structure	flag 192.168.8.93:SYN type_determination	1553809500	2019-03-28-21-45-00-UTC

Figure 11: Example of ‘log’ database table

Figure 12 shows a section of lab web interface, where pressing button ‘Scenario 1’ starts the attack that covers Confidentiality and Integrity modules of the lab, and button ‘Scenario 2’ – Availability module.



Please click one of the buttons to pick an attack for investigation:

Flag: Compromised Account - please enter the name

Flag: What's the secret code in the file the newbie attacker tried to mess with?

Flag: Floods
From which IP(s) is the server being DOS-ed? Look out for: TCP RST, ICMP ECHO, TCP ACK, UDP, TCP SYN, ICMP Blacknurse, TCP XMAS floods. Please enter the answer in format IP:ATTACK, where ATTACK should be written like options before. For example:
10.25.12.123:ICMP Blacknurse
172.168.200.50:UDP
(Hint: no need to add something like " flood" to the end of the flag - follow the format of the examples above.)

Scenario 1
Scenario 2
Submit Confidentiality Flag!
Submit Integrity Flag!
Submit Availability Flag!
Submit User Activity Logs

Figure 12: Flag and activity log submission web interface (part)

After any flag is submitted and saved to the database, the interface informs the user if the flag was correct. Before finishing the lab session, the user is requested to press the ‘Submit User Activity Logs’ button to send activity log files to the FTP server to be analyzed later on.

More details about the program code related to the web interface and communication between the machines can be found in the lab repository.

4.6 Evaluation phase

This phase provides general guideline for completing the tasks of the lab and covers analysis and assessment of the gathered results.

4.6.1 General lab solution guideline

In this subsection, some suggestions for general process of solving the lab tasks are provided. It is expected that the analysis of the activity logs will demonstrate deviations from the guideline provided below, making it possible to track differences in the approach to the tasks of the lab.

A. Scenario 1

The first attack scenario covers Confidentiality and Integrity modules of the lab. Once the scenario has been started, the server can be accessed from the lab desktop machine over SSH with the credentials provided in the instructions. To detect the username of the successfully brute-forced account, a correct log file (`/var/log/auth.log`) should be analyzed. After some time after the attack started, the line containing text "Accepted password for user from 192.168.8.90" will appear in the file. Later on, second line with the same text will appear, which would mean that the attacker successfully brute-forced the credentials for the account with the username `user`. This is the flag for the Confidentiality module.

Figure 13 provides sample commands for Confidentiality module solution.

```
ssh student@192.168.8.6
cat /var/log/auth.log | grep -a "Accepted"
```

Figure 13: Sample commands for Confidentiality module solution

After this, student can navigate the file system of the server to find a file that was modified by the attacker. In this case, the file has been given a name that would make it simpler to identify it as the one that has to be analyzed to solve the task. As an alternative to blindly navigating the file system in search of the file, one may try to track all the files that has been recently changed on the system, but the list may be too long, making it difficult to find the correct file.

Attempts to simply output the content of the file to see the flag would not be successful, as the file extension is incorrect, and what looks like a text file is actually a password-protected archive. Students have to find out the correct file type either by using an appropriate tool or by guessing the extension. Once the correct file type is discovered, the password on the archive should be cracked.

If the correct password is found, it is possible to extract the file with the flag for Integrity module from the archive. As mentioned before, this flag is unique for each username.

For example, for a student with username `olyare` the flag is `f61a19e604458051124caf2ea8b7adb5`, which is an MD5 hash of the string `olyareintegrity`.

Figure 14 provides sample commands for Integrity module solution.

```
cd /home/user
ls
find /home/ -mtime -1 -ls
cat your_secret_code_is_here.txt
file your_secret_code_is_here.txt
unzip your_secret_code_is_here.txt
sudo apt install fcrackzip
sudo scp /home/student/Downloads/rockyou.txt
student@192.168.8.6:/home/student
fcrackzip -v -D -u -p /home/student/rockyou.txt your_secret_code_is_here.txt
sudo unzip your_secret_code_is_here.txt
cat secretcode.txt
```

Figure 14: Sample commands for Integrity module solution

B. Scenario 2

For the second attack scenario that covers Availability module of the lab, various tools and filters can be used to capture and analyze the network traffic on the server. This guideline provides several console (Terminal) commands as examples, but the traffic can also be captured to a file, transferred from server machine to desktop, and analyzed there with graphical tools.

The students are instructed to look for the following types of flood attacks in the traffic: TCP RST, ICMP ECHO, TCP ACK, UDP, TCP SYN, ICMP Blacknurse, TCP XMAS. They are also requested to provide the answers in the format IP:ATTACK, for example: `10.25.12.123:ICMP Blacknurse` and `172.168.200.50:UDP`. Correct flags for this lab module are the following:

- `192.168.8.91:ICMP ECHO`
- `192.168.8.92:ICMP Blacknurse`
- `192.168.8.93:TCP SYN`

- `192.168.8.94:TCP ACK`
- `192.168.8.95:TCP RST`
- `192.168.8.96:TCP XMAS`
- `192.168.8.97:UDP`

Figure 15 provides sample commands for Availability module solution.

```
sudo tcpdump icmp
sudo tcpdump udp
sudo tcpdump tcp
sudo tcpdump host 192.168.8.93
sudo tcpdump host 192.168.8.94
sudo tcpdump host 192.168.8.95
sudo tcpdump host 192.168.8.96
sudo tcpdump host not 192.168.8.1
sudo tcpdump host not 192.168.8.1 -w tcpdump.pcap
sudo scp /home/student/tcpdump.pcap student@192.168.8.1:/home/student/
```

Figure 15: Sample commands for Availability module solution

It is also highly possible that the students would conduct additional research on the Internet to find out key features of various flood attack types to recognize them in the network traffic.

4.6.2 General statistics regarding task completion and log submission levels

As mentioned earlier, two groups of students (in total, 64 participants) worked on the lab. First group consisted of 31 students, second group – of 33 students.

Table 3, Table 4, and Figure 16 provide flag submission statistics for both groups.

Table 3: Group 1 flag submission statistics

	Correct flag submitted	**No correct flag submitted**	**Not attempted to submit any flags**
Confidentiality flag	21	1	9
Integrity flag	2	20	
Availability flag 1	4	18	
Availability flag 2	5	17	
Availability flag 3	4	18	
Availability flag 4	4	18	
Availability flag 5	6	16	
Availability flag 6	5	17	
Availability flag 7	4	18	

Table 4: Group 2 flag submission statistics

	Correct flag submitted	**No correct flag submitted**	**Not attempted to submit any flags**
Confidentiality flag	24	3	6
Integrity flag	10	17	
Availability flag 1	11	16	
Availability flag 2	8	19	
Availability flag 3	9	18	
Availability flag 4	9	18	
Availability flag 5	12	15	
Availability flag 6	9	18	
Availability flag 7	8	19	

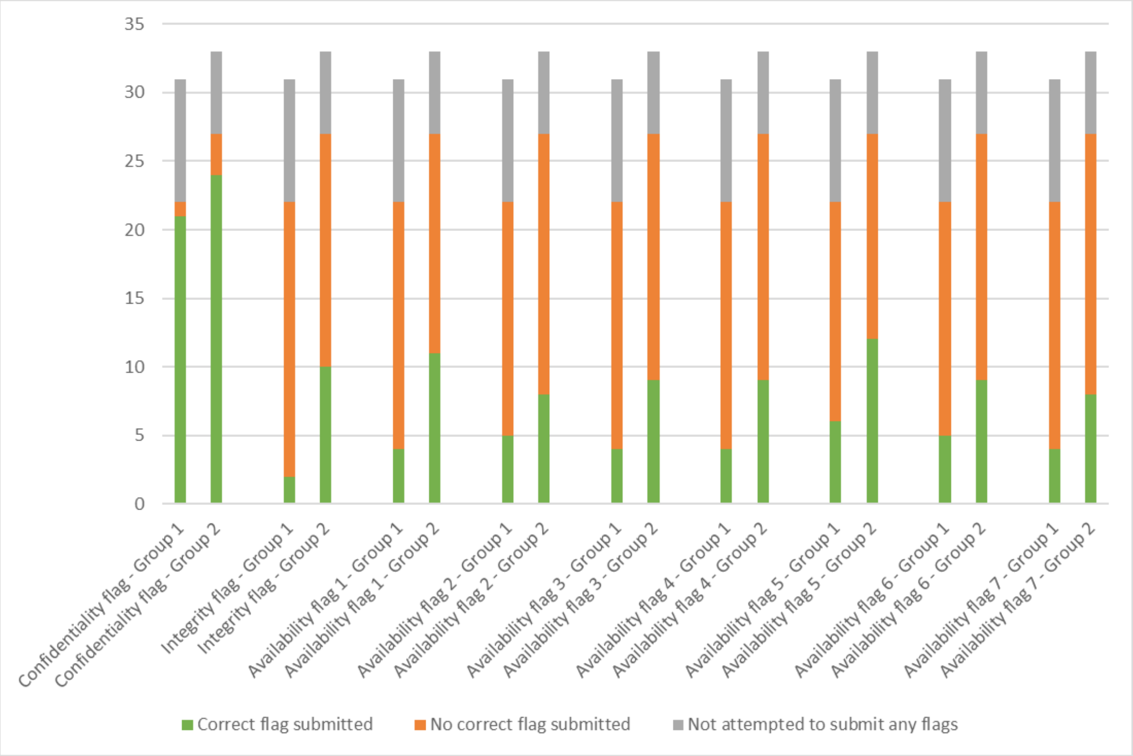


Figure 16: Flag submission statistics

As can be seen, the second group submitted a higher amount of correct flags. Same applies to the activity log submission statistics (see Table 5 and Figure 17).

Table 5: Activity log submission statistics

	Submitted	**Not submitted**
Group 1	17	14
Group 2	26	7
Total	43	21

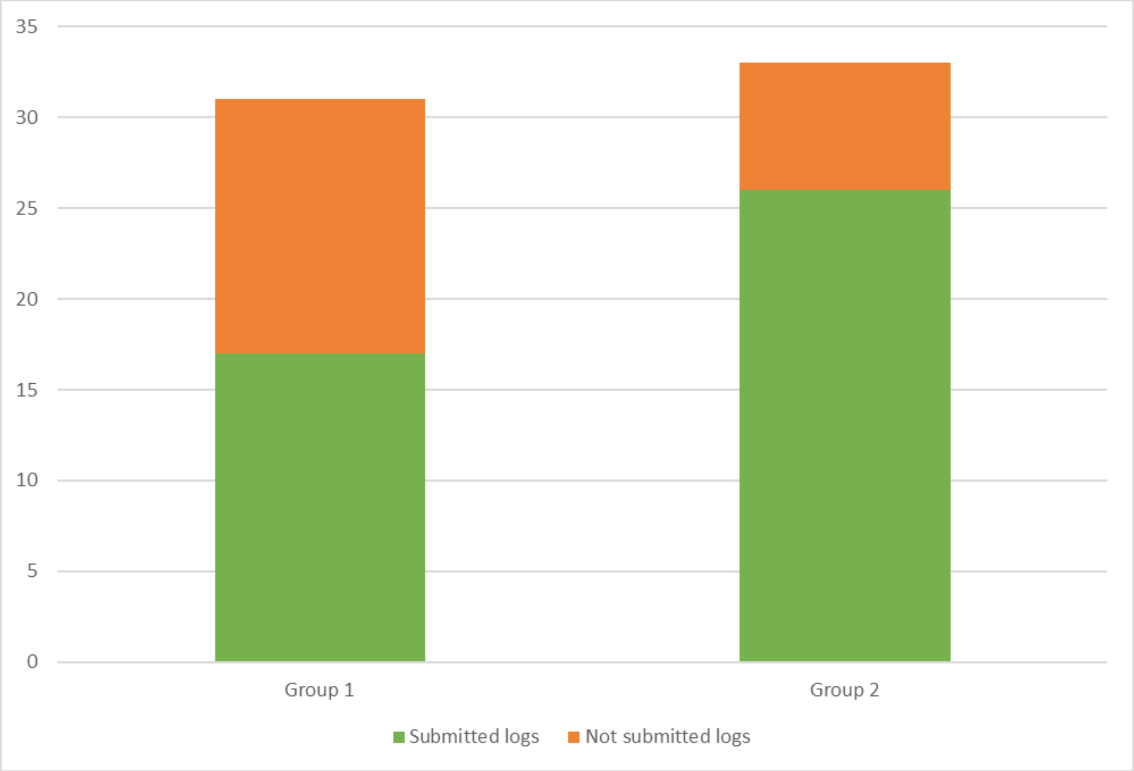


Figure 17: Activity log submission statistics per group

The log files submitted by Group 2 were overall more complete and contained more information about the users' activities. Many students in Group 1 decided to not submit their activity logs, forgot to do it, or might have been unable to submit the files due to technical issues with the virtual environment.

The students that did not attempt to submit any flags (9 from Group 1, and 6 from Group 2), however, submitted their activity log files.

Flag submission attempts were assigned a category: "Valid" or "Invalid". Examples of "Invalid" flag submissions are submissions that were empty (when a participant clicked on submit button without inserting any value into the submission field), contained one or more empty spaces, groups of flag submissions that contained strings of the same characters of varying length. "Valid" attempts category contains "Correct" and "Incorrect" flag submissions.

To avoid the results of further analysis being skewed, it was decided to check data for outliers. As can be seen in Figure 18, there is one extreme outlier in the dataset.

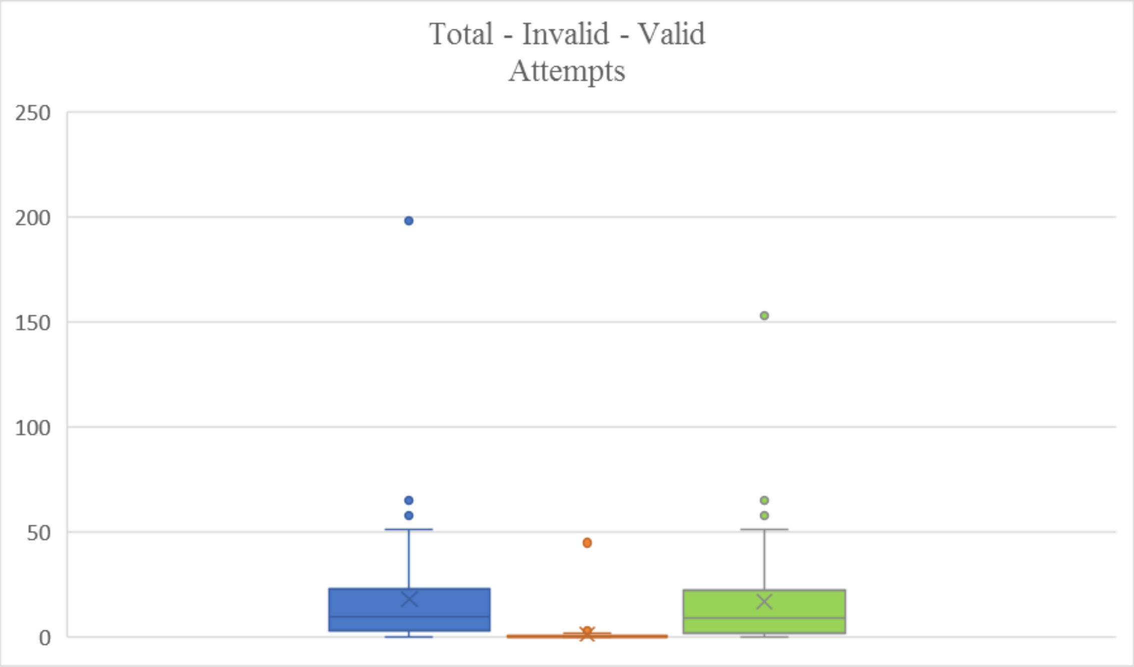


Figure 18: Detecting outliers using boxplot

Analysis of the activity logs and flag submissions of the outliers showed that only extreme outlier with “Total” – “Invalid” – “Valid” attempt counts 198, 45, and 153, accordingly, has to be removed. High attempt counts for other outliers were a result of normal activity. Removing the outlier from the dataset reduces the total amount of participants to be analyzed to 63. Table 6 provides statistical summary of “Total”, “Invalid”, and “Valid” attempts.

Table 6: Statistical summary of Total, Invalid, Valid attempts

	Total	**Invalid**	**Valid**
Mean	15.254	0.556	14.698
Median	9	0	9
Mode	1	0	1
Minimum	0	0	0
Maximum	65	4	65
Count	63	63	63

Table 7 presents statistics for “Correct” and “Incorrect” subcategories of “Valid” flag submission attempts.

Table 7: Statistical summary of Correct and Incorrect attempts

	Correct	**Incorrect**
Mean	2.397	12.302
Median	1	7
Mode	1	0
Minimum	0	0
Maximum	9	62
Count	63	63

Distribution of the total amount of flag submission attempts is shown in Figure 19. 19% of all flag submission attempts were valid.

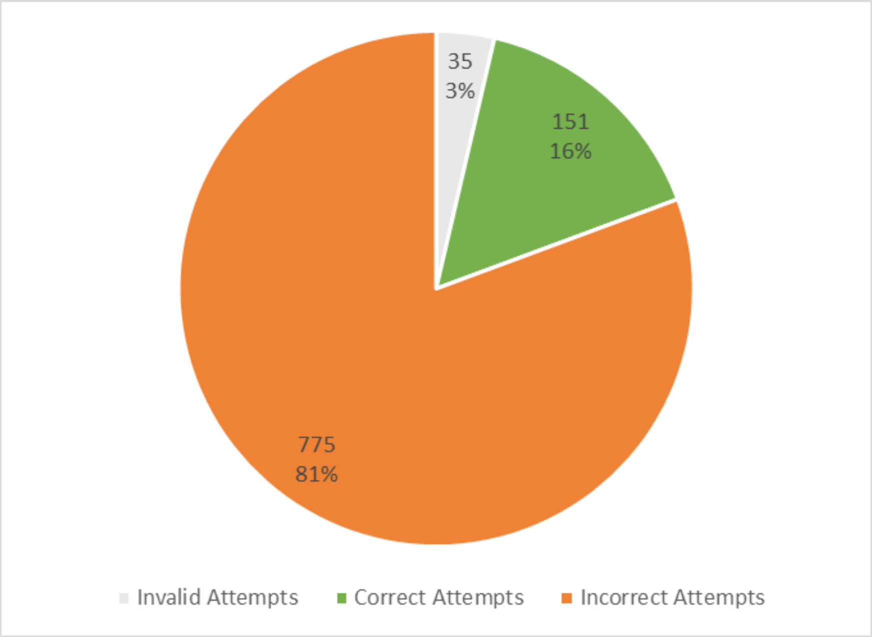


Figure 19: Total flag submission attempts distribution summary

Distribution of valid flag submission attempts is shown in Figure 20.

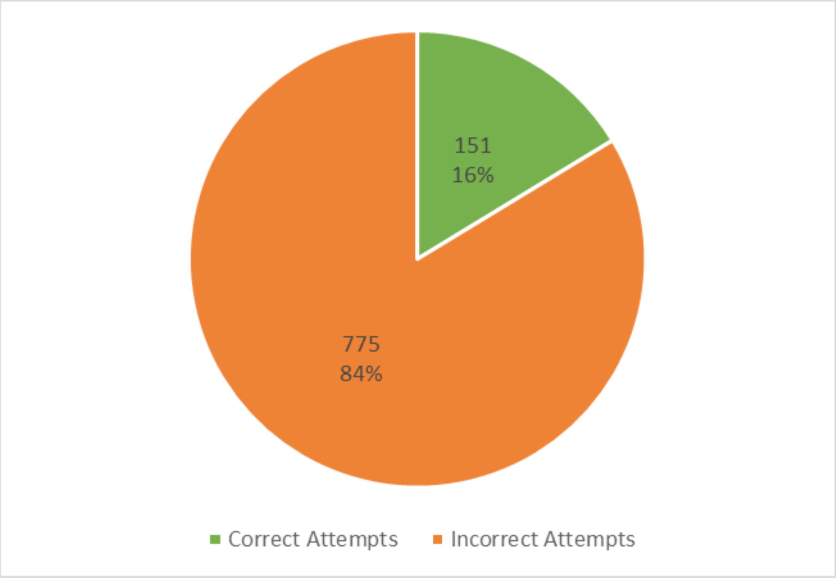


Figure 20: Valid flag submission attempts distribution summary

Figure 21 shows a summary of correct and incorrect flag submission attempts of all students. As can be seen, all students had more incorrect submission attempts than correct ones. However, the ratio between the amount of correct and incorrect flag submission varies significantly from one participant to another.

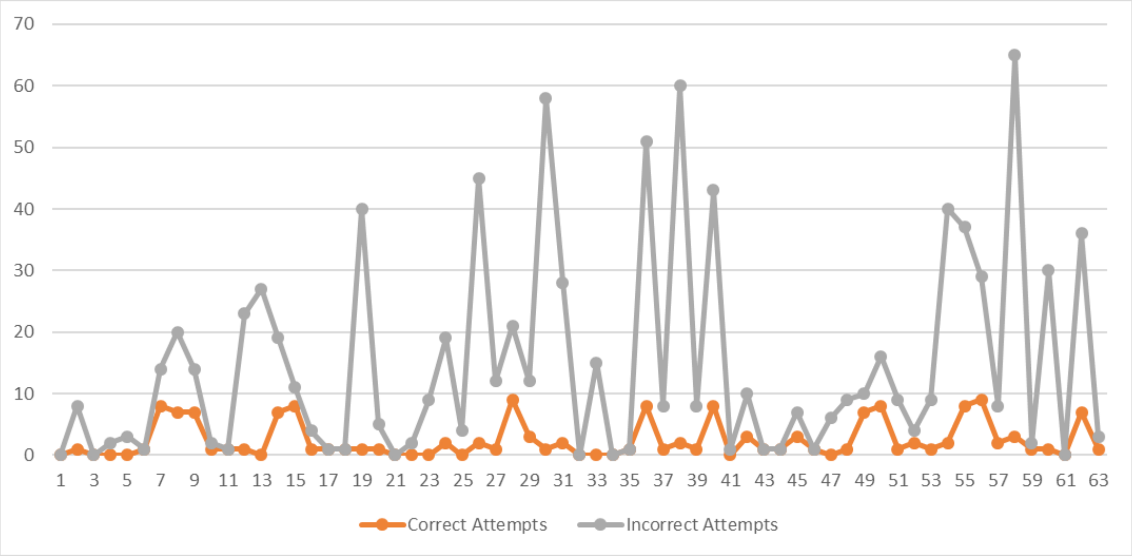


Figure 21: Summary of Correct and Incorrect flag submission attempts of all students

Further analysis of individual user activity logs and the numbers of submission attempts for each module of the lab separately provides more insights about the reasons behind the high rates of incorrect submission attempts.

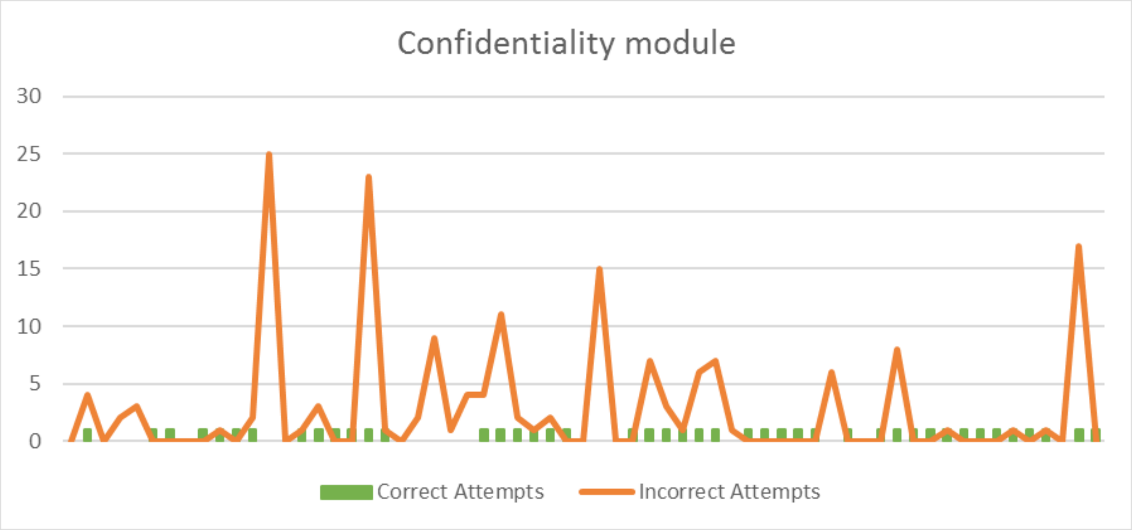


Figure 22: Summary of valid flag submission attempts in Confidentiality module

In Confidentiality module, most of the students were able to resolve the given task with low count of incorrect attempts, or no incorrect attempts (see Figure 22). Analysis of available activity logs allowed to conclude that the main reason behind the high rates of incorrect flag submission attempts in this case was due to manual brute-forcing of an answer by the participants when they: (a) were able to find the log showing the attacker's attempts to find an account with vulnerable password, but did not manage to filter out the successful attack, and ended up trying to submit all the same usernames that the attacker was attempting to access, (b) were not able to find the correct log file and tried to guess the answer instead, or (c) were submitting system account usernames from wrong server log files.

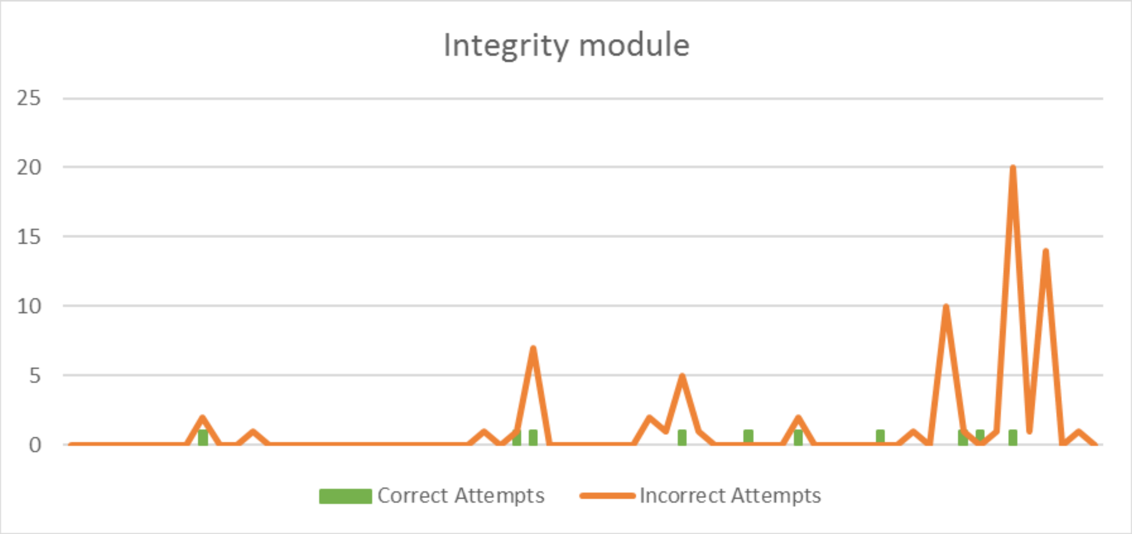


Figure 23: Summary of valid flag submission attempts in Integrity module

In Integrity module, incorrect submissions consisted of mistyped flags, filename of the archive containing the flag, passwords that were used to protect the archive with the flag, or random words. Figure 23 provides a summary of valid flag submission attempts for the module.

Analysis of submission attempts (see Figure 24 for statistics) and logs related to the Availability module showed that many students were inattentive when reading provided instructions. As a result, there were many flag submissions that were of the wrong format (for example, an extra word added to the flag). Several students attempted to brute-force correct answers by combining suggested attack options (and sometimes their own ideas of attack abbreviations outside of the list) with various IP addresses they could see in the network traffic, without analyzing the information they could see in the traffic to define what kind of communication or attack is happening. There were multiple cases of submissions containing typographical errors, as well. Some students were unable to capture the traffic and attempted to submit flags with random IP addresses.

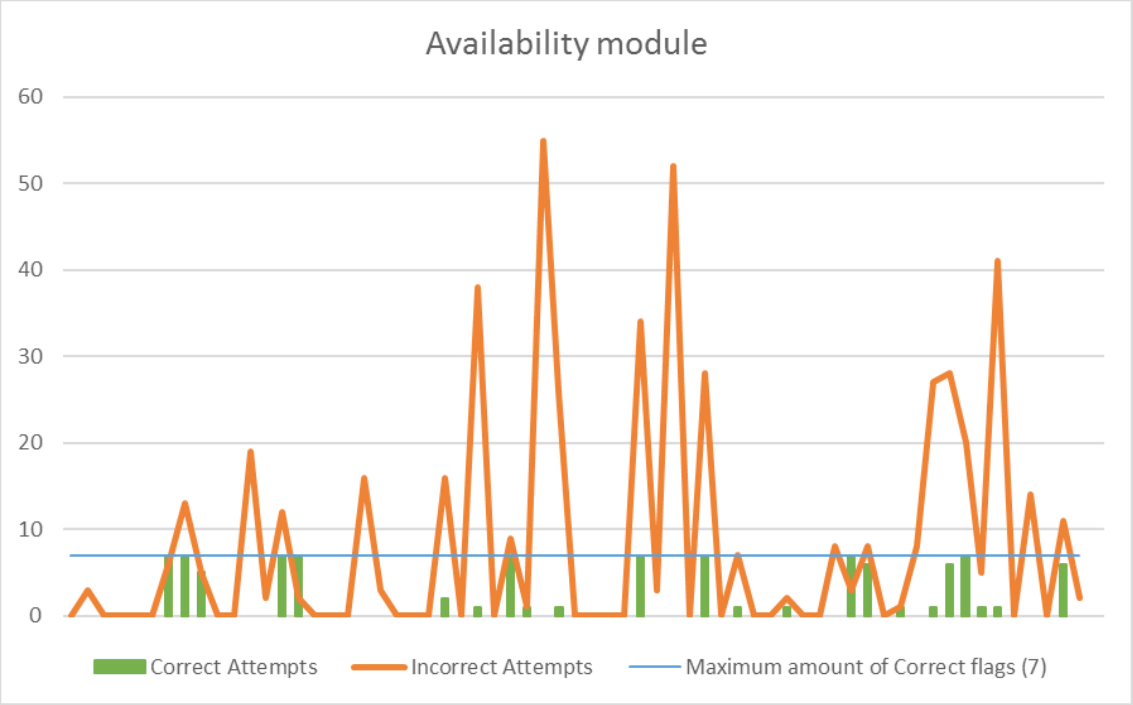


Figure 24: Summary of valid flag submission attempts in Availability module

So, most of the incorrect submissions in the lab were due to:

- students making typographical errors;
- attempts to brute-force the answers;
- inattentiveness;
- level of pre-existing knowledge lower than expected from the target group;
- misunderstanding of instructions.

Despite this, it was still possible to detect the tools and commands used by students. Moreover, logs actually helped to identify the main reasons behind incorrect submissions, so these issues can be indicated in the skill profiles of the participants to be addressed as needed. Further analysis of user activity logs regarding tools and commands used by participants will follow in the next sections of the chapter.

4.6.3 User activity logs analysis

The data obtained from log files for logkeys keylogger, `sa -u` and `lastcomm` commands provided biggest amount of information about user activity that is not traceable in cybersecurity exercises from simple flag submission logs. In cases where Bash was exited as required, bash history file sometimes also provided useful information about user activity in Terminal on the virtual Desktop side. Further, several user activity log samples are provided, showing what types of information could be extracted. Identifying information (such as timestamps) is modified in the log samples, while the rest of the log that is illustrating a case is left without changes.

A. Commands typed in and tools used by the users

Figure 25 provides an extract from of logkeys keylogger file, illustrating various console commands that a student used when working on the virtual lab.

```
2019-01-01 05:10:06+0100 > tcpdump
2019-01-01 05:10:18+0100 > ifconfig
2019-01-01 05:10:23+0100 > sudo tcpdump
2019-01-01 05:10:30+0100 > <Ctrl>c
2019-01-01 05:10:39+0100 > tcpdump -h
2019-01-01 05:10:47+0100 > <Up><Up>
2019-01-01 05:12:25+0100 > <Ctrl>c
2019-01-01 05:12:38+0100 > <Up> <LShft>> tcplog.txt
2019-01-01 05:13:09+0100 >
<Up><Left><#+9><Left><Left><Left><Left><Left><Left><Left><Left><Left>-
2019-01-01 05:13:20+0100 > ls
```

Figure 25: Example of keylogger log file structure, illustrating various logged commands

As can be seen, the keylogger file contains timestamps that allow to establish the timeframe in which the user performed certain actions and match them to timestamps from flag submission database, if needed.

Figure 26 contains an extract from a bash history file with commands that indicate that a user connected and disconnected from a virtual Server, copied a file from it to the Desktop machine, attempted to run Wireshark, then installed it after discovering that the tool had not yet been installed on the Desktop.

```
ssh student@192.168.8.6
exit
sudo scp /home/student/tcpdump.pcap student@192.168.8.6:/home/student/
wireshark
sudo apt install wireshark-qt
```

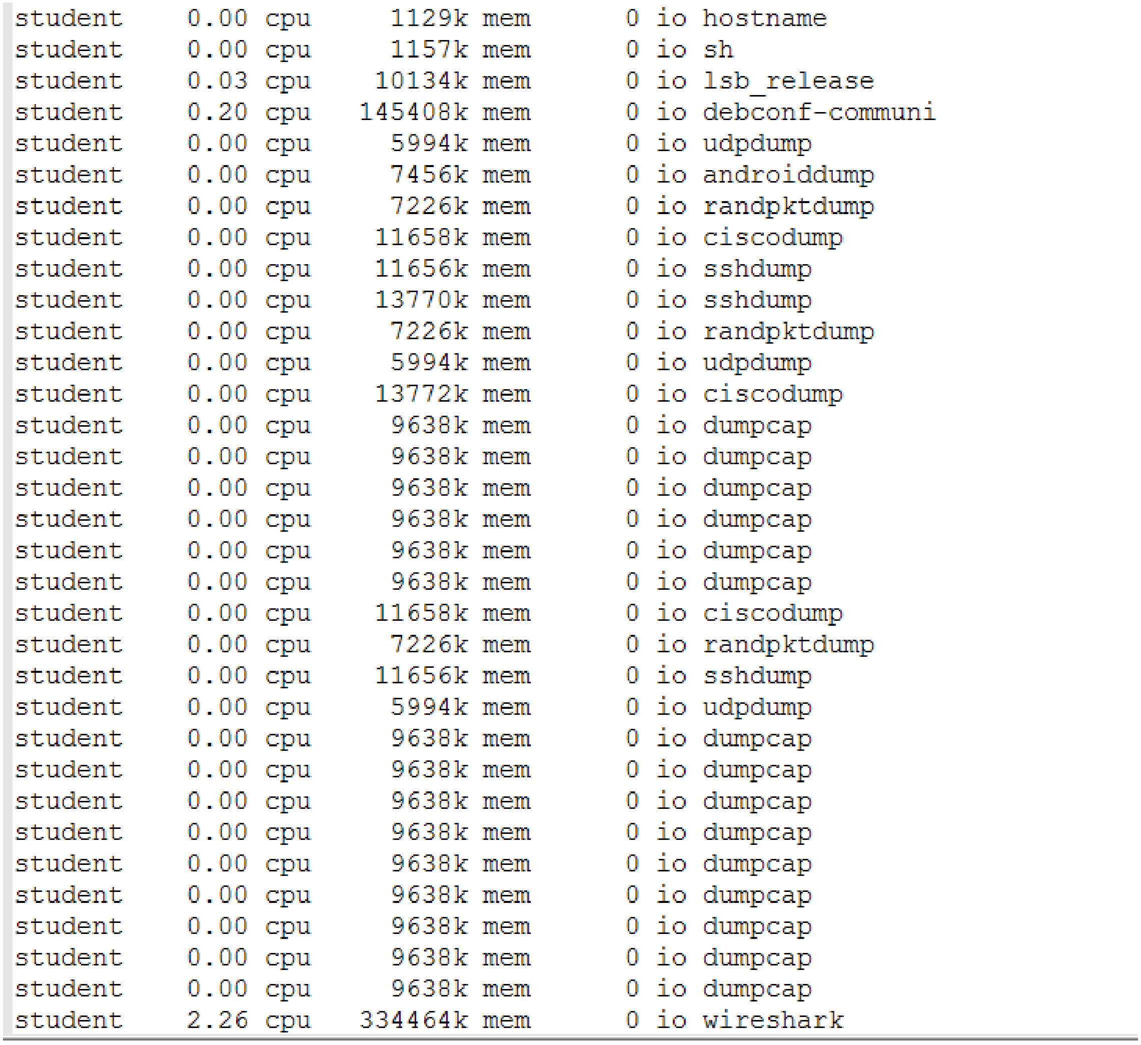
Figure 26: Example of a bash history file section

Figure 27 illustrates how the same attempt of running and installing Wireshark looks like in keylogger log file.

```
2019-01-01 12:00:41+0100 > wireshark
2019-01-01 12:00:55+0100 > sudo apt install i<BckSp>wireshark-qt
2019-01-01 12:01:04+0100 >
```

Figure 27: Keylogger log file section illustrating an attempt to run and install Wireshark

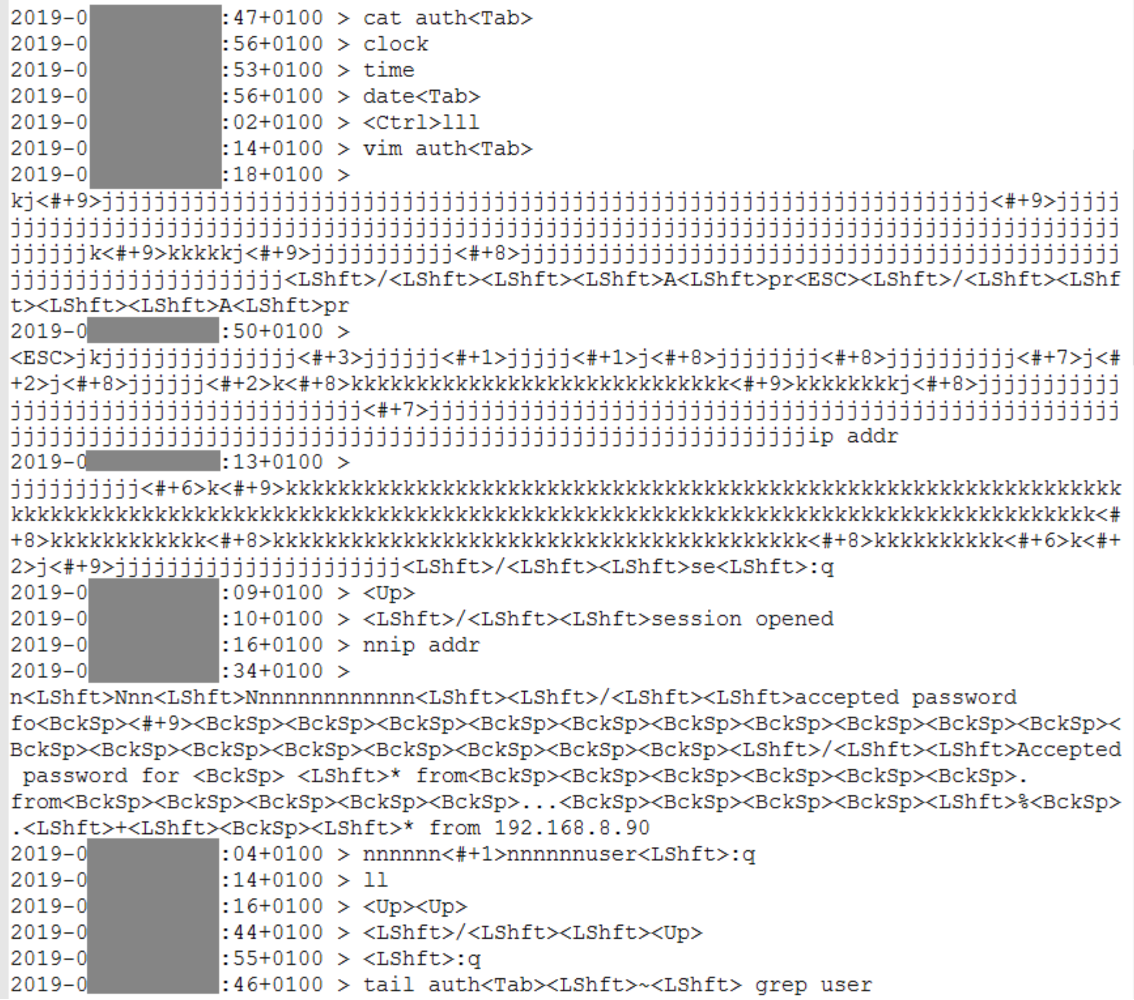
Figure 28 contains a section of `sa -u` command log file with indicators that Wireshark was run on the virtual Desktop machine.



```
student    0.00 cpu     1129k mem      0 io hostname
student    0.00 cpu     1157k mem      0 io sh
student    0.03 cpu    10134k mem      0 io lsb_release
student    0.20 cpu   145408k mem      0 io debconf-communi
student    0.00 cpu     5994k mem      0 io udpdump
student    0.00 cpu     7456k mem      0 io androiddump
student    0.00 cpu     7226k mem      0 io randpktdump
student    0.00 cpu    11658k mem      0 io ciscodump
student    0.00 cpu    11656k mem      0 io sshdump
student    0.00 cpu    13770k mem      0 io sshdump
student    0.00 cpu     7226k mem      0 io randpktdump
student    0.00 cpu     5994k mem      0 io udpdump
student    0.00 cpu    13772k mem      0 io ciscodump
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu    11658k mem      0 io ciscodump
student    0.00 cpu     7226k mem      0 io randpktdump
student    0.00 cpu    11656k mem      0 io sshdump
student    0.00 cpu     5994k mem      0 io udpdump
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    0.00 cpu     9638k mem      0 io dumpcap
student    2.26 cpu   334464k mem      0 io wireshark
```

Figure 28: Example of `sa -u` command log file section with indicators that Wireshark was run

From the example in Figure 29 it can be seen that the format of logging in some cases makes it difficult to understand which command a user decided to use after multiple self-corrections, in case of reuse of a previously used console commands if user navigates among them using arrows on the keyboard, and in cases when the command-line tool needs to be navigated in a way that results in generation of log data that is difficult to read.



```
2019-0[redacted]:47+0100 > cat auth<Tab>
2019-0[redacted]:56+0100 > clock
2019-0[redacted]:53+0100 > time
2019-0[redacted]:56+0100 > date<Tab>
2019-0[redacted]:02+0100 > <Ctrl>lll
2019-0[redacted]:14+0100 > vim auth<Tab>
2019-0[redacted]:18+0100 >
kj<#+9>jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj<#+9>jjjjj
jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj
jjjjjjk<#+9>kkkkj<#+9>jjjjjjjjjjj<#+8>jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj
jjjjjjjjjjjjjjjjjjjj<LShft>/<LShft><LShft><LShft>A<LShft>pr<ESC><LShft>/<LShft><LShf
t><LShft><LShft>A<LShft>pr
2019-0[redacted]:50+0100 >
<ESC>jkjjjjjjjjjjjjjjj<#+3>jjjjjj<#+1>jjjjj<#+1>j<#+8>jjjjjjj<#+8>jjjjjjjjj<#+7>j<#
+2>j<#+8>jjjjjj<#+2>k<#+8>kkkkkkkkkkkkkkkkkkkkkkkkkkkk<#+9>kkkkkkkj<#+8>jjjjjjjjjjj
jjjjjjjjjjjjjjjjjjjjjjjjjjj<#+7>jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj
jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjip addr
2019-0[redacted]:13+0100 >
jjjjjjjjjj<#+6>k<#+9>kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk
kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk<#
+8>kkkkkkkkkkk<#+8>kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk<#+8>kkkkkkkkk<#+6>k<#+
2>j<#+9>jjjjjjjjjjjjjjjjjjjjj<LShft>/<LShft><LShft>se<LShft>:q
2019-0[redacted]:09+0100 > <Up>
2019-0[redacted]:10+0100 > <LShft>/<LShft><LShft>session opened
2019-0[redacted]:16+0100 > nnip addr
2019-0[redacted]:34+0100 >
n<LShft>Nnn<LShft>Nnnnnnnnnnnn<LShft><LShft>/<LShft><LShft>accepted password
fo<BckSp><#+9><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><
BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><LShft>/<LShft><LShft>Accepted
 password for <BckSp> <LShft>* from<BckSp><BckSp><BckSp><BckSp><BckSp><BckSp>.
from<BckSp><BckSp><BckSp><BckSp><BckSp>...<BckSp><BckSp><BckSp><BckSp><LShft>%<BckSp>
.<LShft>+<LShft><BckSp><LShft>* from 192.168.8.90
2019-0[redacted]:04+0100 > nnnnnn<#+1>nnnnnnuser<LShft>:q
2019-0[redacted]:14+0100 > ll
2019-0[redacted]:16+0100 > <Up><Up>
2019-0[redacted]:44+0100 > <LShft>/<LShft><LShft><Up>
2019-0[redacted]:55+0100 > <LShft>:q
2019-0[redacted]:46+0100 > tail auth<Tab><LShft>~<LShft> grep user
```

Figure 29: Keylogger log file section illustrating `vim` text editor being used extensively

The keylogger does not capture such events as mouse clicks or switching between different tools. Due to this, in some cases it may be difficult to differentiate whether the logged line is a command the user tried to run in console, or an Internet search query.

B. Internet search queries

Considering the specifics of target audience of the virtual lab, it was easy to spot the sections of keylogger log file containing Internet search queries that some of the students used to search for information that could help them to solve a task (see Figure 30 for an example).

```
2019-01-01 01:24:08+0100 > <Up>how to cvapt<BckSp><BckSp><BckSp><BckSp>apture
tcpdump l<BckSp>flags
2019-01-01 01:27:28+0100 >
<Left><Left><Left><Left><Left><Left><Left><Left><#+9><Left><Right><Right><Rig
ht><Right><BckSp><BckSp><BckSp>rst
```

Figure 30: Keylogger log file section containing an Internet search query (Example 1)

In some cases, the search queries contained indicators that the user misunderstood the task or misinterpreted some of the information obtained when performing previous steps needed to find a flag for a task (see Figure 31).

```
2019-01-15 05:35:48+0000 > <Ctrl>zzhow to decrypt binary file ubuntu
2019-01-15 05:36:25+0000 > aa<Ctrl>ahow to see
<BckSp><#+8><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><BckSp><B
ckSp><BckSp><BckSp><BckSp><BckSp>llinux how to see what encryption is used
2019-01-15 05:39:42+0000 > <Up>
2019-01-15 05:40:45+0000 > y<Ctrl>zsl 0<BckSp><BckSp><BckSp><BckSp>ls -al
2019-01-15 05:41:18+0000 > h<Ctrl>ahow to crack
txt<BckSp><BckSp><BckSp>encrypted txt files
2019-01-15 05:43:31+0000 >
```

Figure 31: Keylogger log file section containing Internet search queries (Example 2)

The queries of the users can be used to define if they have awareness or knowledge about a certain topic.

```
2019-01-15 01:14:02+0100 > open pcap file wireshark
2019-01-15 01:16:14+0100 > open wiresahrk
<BckSp><BckSp><BckSp><BckSp><BckSp>hark in terminal
```

Figure 32: Keylogger log file section containing Internet search queries (Example 3)

```
student  250.88 cpu   643712k mem      0 io firefox
student    0.00 cpu     4998k mem      0 io bash
student    0.43 cpu    27520k mem      0 io sshd           *
```

Figure 33: Section of `sa -u` command log file with indicator that Firefox browser was run

```
sshd             SF    student  __         0.43 secs Mon Jan  7 13:57
bash                   student  pts/1      0.00 secs Mon Jan  7 13:57
firefox          S   X student  __       250.88 secs Mon Jan  7 11:40
```

Figure 34: Section of `lastcomm` command log file with indicator that Firefox browser was run

When combined with data from other log files, the information can be used to define, for example, if the user was successful in locating the Internet resources that provided necessary information and applying it to complete a task.

Figure 32, Figure 33, and Figure 34 contain samples of logs with indicators that a browser was used and another example of an Internet search query.

For some students, the search queries indicated that an alternative way to solve a task was chosen. For example, in Integrity module there were attempts to crack the password by uploading the archive into an online tool for password cracking[1] after copying it from Server machine to Desktop, instead of using a command-line tool.

Log files of `sa -u` and `lastcomm` commands, while being useful to determine when a browser or a tool such as Wireshark were used, provided little specifics about commands executed in Terminal or existing connections via SSH, for example. Also, due to the specifics of the commands used to generate these logs, the structure of the files was difficult to properly interpret and contained multiple repeating lines.

The logs generated from the commands `sudo aureport`, `ac -p`, and `sa -m` did not contain information about user activity that could be helpful to define skills of the lab participants.

4.6.4 User skills evaluation

As was mentioned previously, 43 out of 64 participants (approximately 67%) submitted their activity logs. One of these participants was the outlier that was removed from the dataset. The activity logs submitted by the users varied in their completeness. It was noticed that in several cases keylogger started logging the keystrokes with a delay, or stopped and restarted the logging process at some point during the lab, which also made some of the logs provided by the users, incomplete.

The logs were reviewed and analyzed to retrieve information regarding the tools and commands used by the participants and search queries they performed inside the virtual machine, where possible.

[1] https://passwordrecovery.io/

Out of 42 students that provided their activity logs, 9 participants (approximately 21%) submitted logs of bad quality. These logs were submitted without following the log submission process instructions, which resulted in logs containing only small parts of information about the activities of the users. As such, these logs were unsuitable for deeper analysis. Logs of the rest 33 participants contained enough information to provide insights about the users' activities in the lab.

6 of the students had low activity level, as they only attempted one of the modules of the virtual lab. As a result, it was possible to evaluate only knowledge, skills, and ability to performed tasks that were mapped to corresponding modules, but not modules that were not attempted, or those non-specific to lab modules (see Table 8).

Table 8: Evaluation of participants with low activity level

	Correct flags	**Confidentiality module**				**Some of the tools used (if such information available)**
Student 2	1/1	🟩 K0132	🟩 K0179	🟦 T0036	🟦 S0067	
Student 16	1/1	🟩 K0132	🟩 K0179	🟩 T0036	🟩 S0067	
Student 18	1/1	🟩 K0132	🟩 K0179	🟦 T0036	🟦 S0067	
Student 33	0/1	🟥 K0132	🟦 K0179	🟦 T0036	🟦 S0067	
Student 47	1/1	🟩 K0132	🟩 K0179	🟦 T0036	🟦 S0067	
	Correct flags	**Availability module**				**Some of the tools used (if such information available)**
Student 35	0/7	🟦 K0255	🟦 K0301	🟨 T0240	🟦 S0156	tcpdump
	Color legend					
🟩	Student completed the corresponding activity and succeeded in demonstrating the skill/knowledge/task/ability					
🟨	Student partially succeeded in demonstrating the skill/knowledge/task/ability while attempting to complete the corresponding activity (for example, by showing the awareness about related tools or commands, but failing to successfully complete the related activity)					
🟥	Student failed to demonstrate the skill/knowledge/task/ability while attempting to complete the corresponding activity					
🟦	Student did not attempt related activities due to problems with previous steps of the module / There was no information in the logs that could provide insights about the skill/knowledge/task/ability					
⬜	(No attempt was made to complete the related lab module)					

Other 27 participants attempted to complete more than one module of the lab. As a result, it was possible to gain more insights from the activity logs they provided. For the demonstration, skill evaluation of several active participants can be found in Table 9. The color legend used in the process of creation of evaluation table is the same as in Table 8.

Table 9: Skill evaluation table example

	Student 27	**Student 44**	**Student 45**	**Student 57**	**Student 58**	**Student 59**
Forensic lab (requirements, non-specific to lab modules)	🟨 A0043	🟨 A0043	🟨 A0043	🟩 A0043	🟨 A0043	🟨 A0043
	🟩 K0077	🟦 K0077	🟨 K0077	🟩 K0077	🟩 K0077	🟩 K0077
	🟨 K0119	🟦 K0119	🟨 K0119	🟩 K0119	🟨 K0119	🟩 K0119
	🟨 K0224	🟨 K0224	🟨 K0224	🟩 K0224	🟨 K0224	🟨 K0224
	🟨 T0027	🟦 T0027	🟨 T0027	🟩 T0027	🟨 T0027	🟨 T0027
	🟩 T0286	🟦 T0286	🟥 T0286	🟩 T0286	🟨 T0286	🟩 T0286
	🟨 S0065	🟨 S0065	🟨 S0065	🟩 S0065	🟨 S0065	🟨 S0065
Confidentiality module	🟩 K0132	🟩 K0132	🟩 K0132	🟩 K0132	🟩 K0132	🟩 K0132
	🟩 K0179	🟩 K0179	🟩 K0179	🟩 K0179	🟩 K0179	🟩 K0179
	🟦 T0036	🟦 T0036	🟦 T0036	🟩 T0036	🟦 T0036	🟦 T0036
	🟦 S0067	🟦 S0067	🟦 S0067	🟩 S0067	🟦 S0067	🟦 S0067
Integrity module	🟨 K0187	🟥 K0187	🟦 K0187	🟩 K0187	🟨 K0187	🟩 K0187
	🟩 T0398	🟩 T0398	🟩 T0398	🟩 T0398	🟩 T0398	🟩 T0398
	🟩 T0167	🟦 T0167	🟦 T0167	🟩 T0167	🟦 T0167	🟦 T0167
	🟩 S0092	🟥 S0092	🟥 S0092	🟩 S0092	🟩 S0092	🟩 S0092
Availability module	🟥 K0255	⬜ K0255	🟦 K0255	🟩 K0255	🟥 K0255	🟥 K0255
	🟨 K0301	⬜ K0301	🟦 K0301	🟩 K0301	🟨 K0301	🟨 K0301
	🟩 T0240	⬜ T0240	🟨 T0240	🟩 T0240	🟩 T0240	🟩 T0240
	🟥 S0156	⬜ S0156	🟦 S0156	🟩 S0156	🟥 S0156	🟥 S0156

Searched for task-related information using a browser in the virtual environment	No	Yes	Yes	Yes	No	Yes
Some of the tools used (if such information available)	fcrackzip tcpdump	Firefox	Wireshark tcpdump	Firefox Wireshark tcpdump vim	Firefox hashcat tcpdump vim	Firefox fcrackzip tcpdump nmap
Other notes	Signs of manual brute-force activity found	Attempts to guess password in Integrity module	Multiple typographical errors	Inattentive to instructions – did not follow flag format example at first	Signs of manual brute-force activity found; multiple typographical errors	Attempts of SQL injection attack on a flag submission field

According to the available activity logs, at least 14 students used a browser inside the virtual environment to search for information that could help them to complete the tasks. This is approximately 22% of all participants of the lab. Of course, it is important to note that some of the participants most likely used browsers outside of the virtual environment to look for help.

5 participants looked for help to solve the first task (Confidentiality module), 11 participants – to complete Integrity module, and 9 – Availability module. Figure 35, Figure 36, and Figure 37 show how successful were the students in completing the lab modules after conducting their research on the Internet.

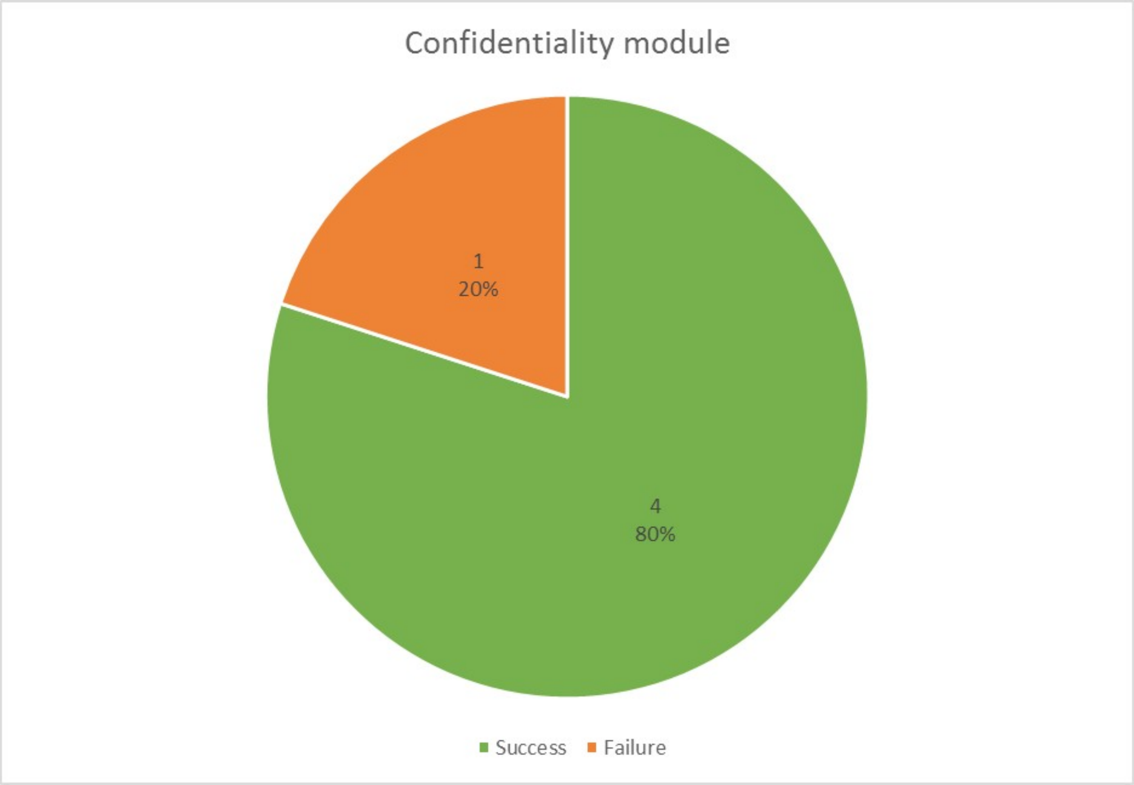


Figure 35: Task completion rate of participants after using Internet (Confidentiality module)

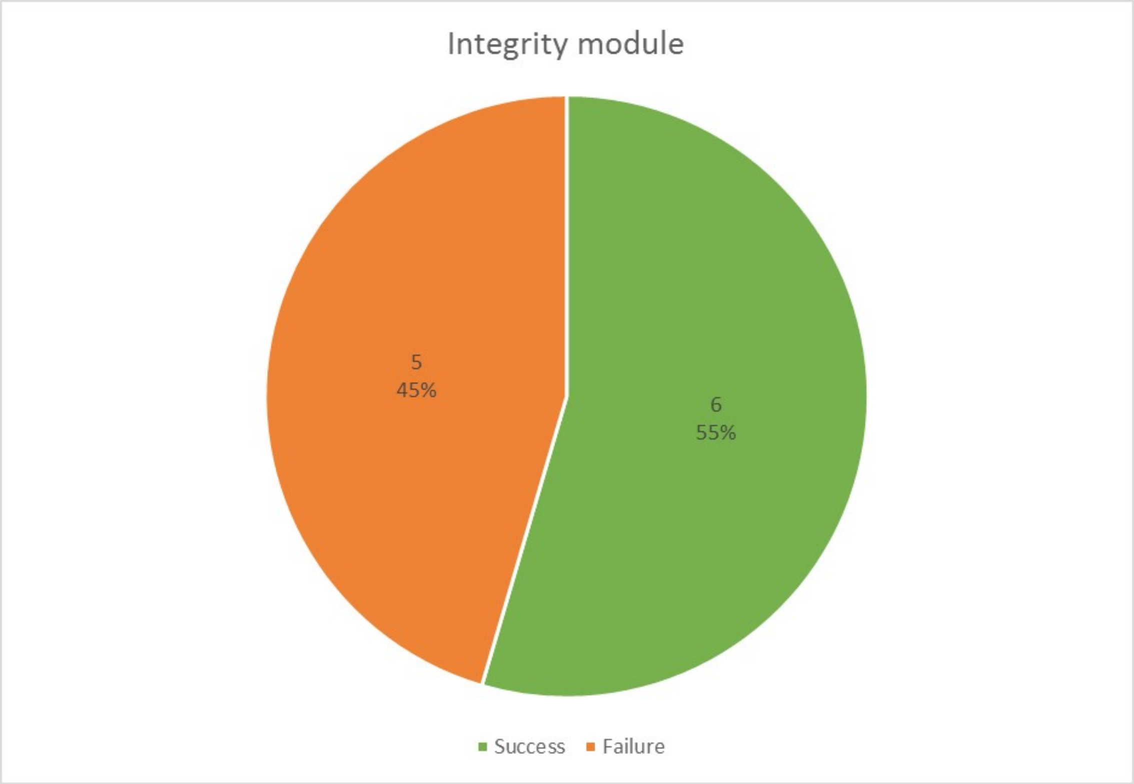


Figure 36: Task completion rate of participants after using Internet (Integrity module)

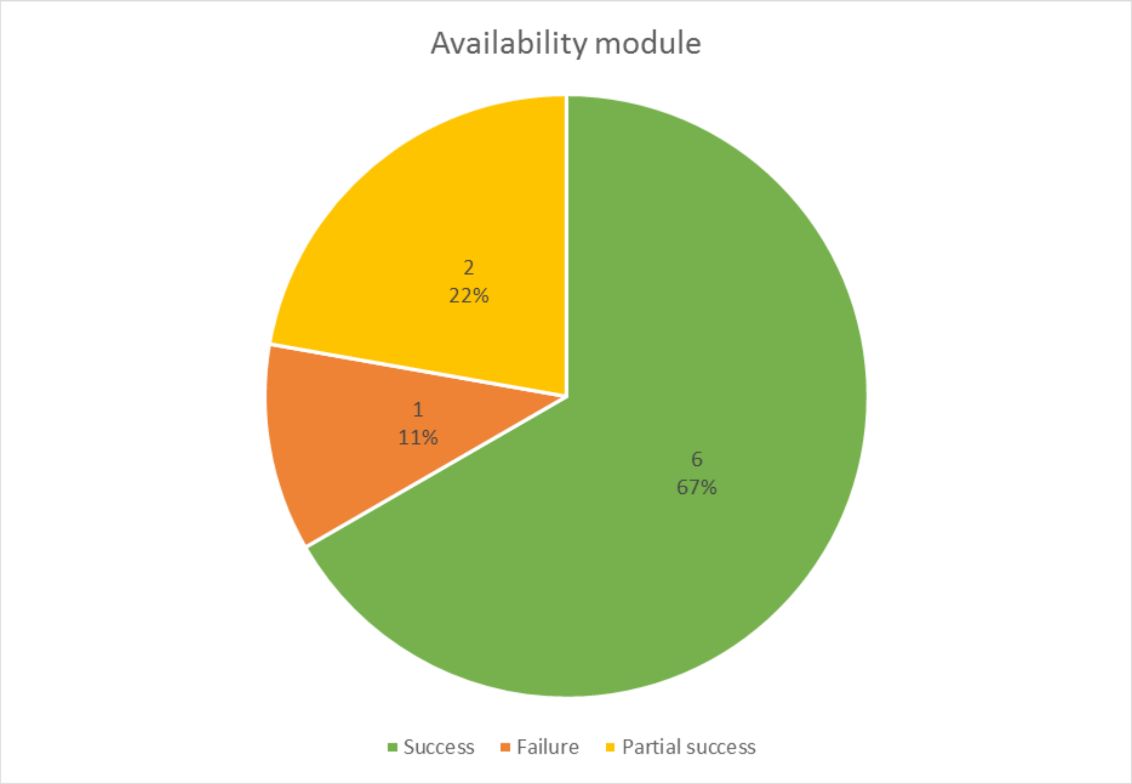


Figure 37: Task completion rate of participants after using Internet (Availability module)

Partial success, mentioned in Figure 37, means that a participant was able to correctly submit several of the flags in Availability module, but missed some of them due to incomplete information received from the Internet search.

Also, deeper analysis of logs and search queries allows to detect cases in which, for example, a participant tried to use various password-cracking tools without success (among such tools – john[1], hashcat[2], fcrackzip[3]), or copied the archive to the virtual Desktop and then uploaded it to an online password cracker – website passwordrecovery.io[4] was a relatively popular choice among those participants that successfully reached the corresponding stage of the lab.

Activity logs contain a lot of information that can provide insights about how successful students were in using certain tools. Manual log review showed that in multiple cases participants attempted several tools and commands (in Integrity module – to crack the

[1] https://manpages.ubuntu.com/manpages/xenial/man8/john.8.html

[2] https://hashcat.net/hashcat/

[3] https://manpages.ubuntu.com/manpages/xenial/man1/fcrackzip.1.html

[4] https://passwordrecovery.io/

password, and in Availability module – to capture and analyze the traffic, before in some cases finding the tool or command they were able to correctly use to solve the task. However, to provide detailed statistics in regards to tool usage count and success rates, quality of the logs has to be higher, and a tool that is capable of parsing the logs, matching the timestamps of usage of various tools by the user against timestamps of the flag submissions and search queries, and then extracting appropriately matched and structured data from the logs, has to be developed and implemented. Due to this, it is not possible to provide clear statistics of tool usage in the lab at the current stage of the research on the topic.

4.6.5 Results discussion and lessons learned

Overall, in case user activity log files were submitted as instructed and all expected information had been logged, it was possible to extract from the logs and define:

- commands run by users;
- used tools;
- cases in which tools with graphical interface were used instead of console-based;
- cases in which students searched information on the Internet to resolve a task;
- cases in which students attempted to brute-force an answer by trying to submit all possible flag combinations;
- cases when students were looking for an answer in a wrong place or used command incorrectly.

Sections of flag submission database containing brute-force patterns could be matched to the appropriate records in the logs by using timestamps and vice-versa. So, it was sometimes possible to determine whether it was indeed an attempt to brute-force a flag, or the user was confused about the format of the flag and possible options (in particular, for Availability module).

It is important to note, that when a tool with graphical user interface was used (for example, Wireshark, or an Internet browser), it was usually difficult to define what exactly the user had been doing, why some conclusion has been reached by the user, or how the tool was used, as keystrokes data was the only source of information. In such cases, screenshots would have been particularly useful.

Also, while keylogger gathers information without differentiating between Desktop and Server virtual machines, other logs were only gathered from the Desktop. Introducing correctly configured monitoring and logging tools on all virtual machines instead of only one may significantly raise the amount of useful data about user activity, that can be used to make a user's skill profile more precise.

After the initial analysis of the log files it became quite apparent that the format of the logs makes it quite difficult if not impossible to speed up the process of log analysis by creating a fully automated script that could parse the data in the logs, extract information about tools or commands that were used, and define whether the users' actions were successful or not, etc. The format of the lab allows too much freedom of choice for the users, making it complicated to define enough patterns that could be used to program an algorithm that would remove or lessen the need of human involvement to analyze the results of users' activities. Otherwise, if the log analysis is not automated, the skills evaluation can be done for relatively small groups of users. In such case, the potential usefulness of the data would most likely be ignored to avoid spending significant amounts of resources and time needed to manually check and assess the logs of user activity.

So, either the tasks and options for tools and commands that can be used even more when designing a lab with user activity monitoring elements for skills evaluation, or alternative and more precise logging tools need to be used. The logs of the tools have to be in a format that is simple to parse and that leaves no possibility for ambiguous interpretations. The functionality of the tools used for activity monitoring should precisely match the requirements of the tasks in the virtual lab and cover as many tools that can be used to complete those tasks, as possible.

5 Suggestions for future work

One of the possibilities in regards to the future work related to the topic include researching possible solutions for automated log parsing and analysis in combinations with different activity monitoring tools than the ones used in this work. Or, as an alternative, development of a new activity monitoring tool that can properly capture the data needed for skills evaluation, from scratch.

Another direction for future work could involve researching other options for virtualized lab environment that would be more efficient and stable, providing better user experience. It may be interesting to compare if a physical infrastructure would be more stable and suitable for cybersecurity exercises with user activity monitoring elements, than virtualized environment. Comparing extensiveness of user activity logs in different setups (number of machines involved in the activity monitoring, different operating systems, lab network structures, etc.) could also provide interesting insights.

There is definitely a need for a deeper research into user activity monitoring with emphasis on ethical considerations, and possible solutions to maintain as much user privacy as possible while still being able to gather enough data to build a comprehensive skill-based profile of the user, for example.

Research also has to been done to define if or how strongly log detail level would depend on the exercise scenario and type of the task that needs to be solved, or if the limitations can be worked around by introducing some new activity monitoring solutions. Last but not least, there is a need to find a way to comprehensively visualize the results of skills evaluation to the user or other interested parties.

As can be seen, there is a lot of research to be done in the area related to skills evaluation of participants of cybersecurity exercises through analysis of user activity.

6 Conclusion

In this thesis author explored the possibilities of using information about user activity for skills evaluation of participants of virtualized cybersecurity exercises. A virtual lab with user activity logging functionality was implemented and the logs were analyzed to determine what kind of information useful for skill-based profiling can be extracted from them, whether it is possible to evaluate certain skills of users or identify possible issues a user might come across while completing the tasks, and which activity monitoring tools were useful or had some significant shortcomings.

It was determined that extensive skills evaluation of participants of cybersecurity exercises in virtualized environments is indeed possible through analysis of user activity data. The main condition for the success of the evaluation is to design the virtualized exercise in such a way that all used activity monitoring tools are fine-tuned to gather data according to the specifics of the tasks, lab objectives, and expectations as to what it is necessary to measure or monitor. Another important requirement for successful and efficient skills evaluation is to automate the whole log analysis process, or significant part of it to make it possible to quickly and easily build skill profiles for big amounts of users.

There are tools that allow to capture information about running processes, tools and commands used by a participant of a virtualized cybersecurity exercise – in particular, one based on a Linux-based operating system. However, the more freedom is given to the participants in the environment, the higher chance that the results of activity analysis will be ambiguous.

References

[1] M. Granåsen and D. Andersson, “Measuring team effectiveness in cyber-defense exercises: a cross-disciplinary case study,” *Cogn. Technol. Work*, vol. 18, no. 1, pp. 121–143, Feb. 2016.

[2] K. Maennel, “Improving and Measuring Learning Effectiveness at Cyber Defence Exercises,” University of Tartu, 2017.

[3] K. Maennel, R. Ottis, and O. Maennel, “Improving and measuring learning effectiveness at cyber defense exercises,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10674 LNCS, pp. 123–138.

[4] A. Malviya, G. A. Fink, L. Sego, and B. Endicott-Popovsky, “Situational awareness as a measure of performance in cyber security collaborative work,” in *Proceedings - 2011 8th International Conference on Information Technology: New Generations, ITNG 2011*, 2010, pp. 937–942.

[5] B. Gros, “Digital games in education: The design of games-based learning environments,” *Journal of Research on Technology in Education*, vol. 40, no. 1. Routledge, pp. 23–38, Sep-2007.

[6] A. Furtună, V. V. Patriciu, and I. Bica, “A structured approach for implementing cyber security exercises,” in *2010 8th International Conference on Communications, COMM 2010*, 2010, pp. 415–418.

[7] J. Brynielsson, U. Franke, M. Adnan Tariq, and S. Varga, “Using cyber defense exercises to obtain additional data for attacker profiling,” in *IEEE International Conference on Intelligence and Security Informatics: Cybersecurity and Big Data, ISI 2016*, 2016, pp. 37–42.

[8] S. Kapetanakis, A. Filippoupolitis, G. Loukas, and T. S. Al Murayziq, “Profiling cyber attackers using case-based reasoning,” in *Nineteenth UK Workshop on Case-Based Reasoning (UK-CBR 2014)*, 2014.

[9] S. Mäses, B. Hallaq, and O. Maennel, “Obtaining Better Metrics for Complex Serious Games Within Virtualised Simulation Environments,” *Eur. Conf. Games Based Learn.*, pp. 428–434, 2017.

[10] S. Mäses, L. Randmann, O. Maennel, and B. Lorenz, “Stenmap: Framework for Evaluating Cybersecurity-Related Skills Based on Computer Simulations,” in

Learning and Collaboration Technologies. Learning and Teaching, 2018, pp. 492–504.

[11] J. McClain *et al.*, "Human Performance Factors in Cyber Security Forensic Analysis," *Procedia Manuf.*, 2015.

[12] R. G. Abbott, J. McClain, B. Anderson, K. Nauer, A. Silva, and C. Forsythe, "Log Analysis of Cyber Security Training Exercises," *Procedia Manuf.*, vol. 3, 2015.

[13] "NICE Cybersecurity Workforce Framework | NIST." [Online]. Available: https://www.nist.gov/itl/applied-cybersecurity/nice/resources/nice-cybersecurity-workforce-framework. [Accessed: 22-Apr-2019].

[14] W. Newhouse, S. Keith, B. Scribner, and G. Witte, "National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework," 2017.

[15] J. Hallett, R. Larson, and A. Rashid, "Mirror, Mirror, On the Wall: What are we Teaching Them All? Characterising the Focus of Cybersecurity Curricular Frameworks," *2018 {USENIX} Workshop on Advances in Security Education ({ASE} 18)*. {USENIX} Association, Baltimore, MD, 2018.

[16] I. Mayer, G. Bekebrede, H. Warmelink, and Q. Zhou, "A Brief Methodology for Researching and Evaluating Serious Games and Game-Based Learning." pp. 357–393, 2013.

[17] "Linux keyloggers – Tuxdiary." [Online]. Available: http://tuxdiary.com/2015/06/08/linux-keyloggers/. [Accessed: 22-Apr-2019].

[18] "List of Linux Key loggers | Hacking & Tricks." [Online]. Available: https://tipstrickshack.blogspot.com/2013/05/list-of-linux-key-loggers.html. [Accessed: 22-Apr-2019].

[19] "Accounting Utilities Manual." [Online]. Available: https://www.gnu.org/software/acct/manual/accounting.html. [Accessed: 22-Apr-2019].

[20] K. Gilbertson, "Process Accounting | Linux Journal," 2002. [Online]. Available: https://www.linuxjournal.com/article/6144. [Accessed: 22-Apr-2019].

[21] "Understanding System auditing with auditd – The Geek Diary." [Online]. Available: https://www.thegeekdiary.com/understanding-system-auditing-with-auditd/. [Accessed: 22-Apr-2019].

[22] M. D. Merrill, L. Drake, M. J. Lacy, and J. Pratt, "Reclaiming Instructional Design," *Educ. Technol.*, 1966.

[23] "ADDIE Model - InstructionalDesign.org." [Online]. Available: http://www.instructionaldesign.org/models/addie/. [Accessed: 22-Apr-2019].

[24] S. Kurt, “ADDIE Model: Instructional Design - Educational Technology.” [Online]. Available: https://educationaltechnology.net/the-addie-model-instructional-design/. [Accessed: 22-Apr-2019].

[25] M. Ernits, J. Tammekaend, and O. Maennel, “i-tee: A fully automated Cyber Defense Competition for Students,” *ACM SIGCOMM Comput. Commun. Rev.*, 2015.

[26] “CIA Triad - IT Security Training & Resources by Infosec.” [Online]. Available: https://resources.infosecinstitute.com/cia-triad/. [Accessed: 22-Apr-2019].

[27] C. Perrin, “The CIA Triad - TechRepublic.” [Online]. Available: https://www.techrepublic.com/blog/it-security/the-cia-triad/. [Accessed: 22-Apr-2019].