TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Saber Yari 177239IVCM

CREATING CYBER SECURITY EXERCISES FOR OPEN SOURCE INTELLIGENCE AND REVERSE ENGINEERING

Master's thesis

Supervisor: Sten Mäses
MSc

Tallinn 2019

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Saber Yari 177239IVCM

OSINT JA PÖÖRDPROJEKTEERIMISE HARJUTUSTE LOOMINE

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: Saber Yari

13.05.2019

Abstract

Gamification is used extensively in the education context to improve the quantity and quality of the learning outcome. In this thesis, the core Open Source Intelligence and Reverse Engineering techniques, the game design elements and how to intercept and modify user's internet traffic to generate better gamification experience are discussed.

We design two gamified cyber security exercise to examine the feasibility of creating an easy to use, automated and customizable cyber security exercise, discuss the game design process in depth and how to decide the effectiveness of the cyber security exercise. For this research, cyber security exercise trainings will be implemented in a virtualized web-based lab.

The successfulness of the chosen gamification design and impact of gamification designs on the motivation of users will be measured. Performance of the students in the exercise lab is argued, and suggestions for future work and possible improvements are discussed.

This thesis is written in English and is 68 pages long, including 7 chapters, 25 figures, and 5 tables.

Annotatsioon

OSINT JA PÖÖRDPROJEKTEERIMISE HARJUTUSTE LOOMINE

Et parandada õppetöö tulemuslikkuse kvaliteeti ja kvantiteeti, kasutatakse hariduses laialdaselt mängustamist. Selles magistritöös arutatakse avatud jälitamise ehk OSINT-i (*Open Source Intelligence*) põhialuseid ja pöördprojekteerimise (*Reverse Engineering*) tehnikaid. Samuti pööratakse tähelepanu mängu disainielementidele ning sellele, kuidas on võimalik sekkuda kasutaja internetiliiklusesse ja seda muuta, et saavutada parem mängustamise kogemus.

Töö käigus disainitakse kaks mängustatud küberkaitseharjutust, millega soovitakse uurida lihtsalt kasutatavate, automatiseeritud ja mugandatavate küberkaitseharjutuste teostatavust. Samuti soovitakse põhjalikult arutada mängu protsessi ja seda, kuidas välja selgitada küberkaitseharjutuste efektiivsus. Tööprotsessis rakendati küberkaitseharjutusi virtualiseeritud veebipõhises keskkonnas.

Antud tööga mõõdetakse valitud mängustamise disaini edukus ja mõju kasutaja motivatsioonile. Tudengite tulemusi küberkaitseharjutustes analüüsitakse lähemalt ning töö lõpus antakse soovitused võimalikeks parandusteks ning edasiseks tööks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 68 leheküljel, 7 peatükki, 25 joonist, 5 tabelit.

List of abbreviations and terms

TalTech	Tallinn University of Technology
i-Tee	Intelligent Training Exercise Environment
OSINT	Open-source intelligence
CA	Certificate Authority
RE	Reverse Engineering
CD	Core Driver
SSL	Secure Sockets Layer
DNS	Domain Name Server
Q&A	Questions and Answers
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
EXIF	Exchangeable Image File Format

Table of contents

List of figures

List of tables

1 Introduction

Using hands-on cyber security exercises is a common practice in the cyber security education field [1]. This thesis takes a look at how to design and use game design elements to teach the basics of cyber security to bachelor students at Tallinn University of Technology (TalTech). Students will be using gamified cyber security exercises to learn about some of the cyber security topics practically.

1.1 Problem Statement and Contribution

In spring of 2019, approximately one hundred bachelor students at TalTech university participated in a gamified course called 'Foundations of Cyber Security'; these computer science students receive cyber security exercise labs built on the i-Tee platform to learn the concepts of cyber security [2]. Implemented exercise labs focused on cyber security topics of Open Source Intelligence (OSINT) and Reverse Engineering (RE). The objective of this study was for students to get familiar with these topics. Gamification design model was used to improve the engagement of the students.

For this research, cyber security exercise labs were oriented around two subjects, first Reverse Engineering (RE) and secondly and more comprehensively Open Source Intelligence (OSINT) techniques. Open Source Intelligence and Reverse Engineering were the chosen cyber security topics to be gamified, designed and implemented. One hundred and two students participated in the course.

In this work, special attention was given to creating the exercise training that would be **relevant, engaging, and individual** (difficult to cheat).

The primary objective of this work was the design and implementation of 2 cyber security exercises. Students used these gamified exercises to practice the chosen hacking techniques in a safe, gamified and easy to use environment. In the end, the results of students were measured and analyzed. This study was an experimental research by implementing the labs and measuring the performance of students. Importance of the

study could be stated as having a reusable and scalable cyber security training for the students of TalTech. Generally, the cyber security training does not build on a scientific design model and do not have justifications for their chosen game design elements, but this study will systematically approach the game design and ensures that these new cyber security exercises could be used as a guide for creating more exercises in the future.

1.2 Limitations

The study does come with limitations and key assumptions. Designing and implementing the exercise labs took more than five months. Some phases of the gamification design phases like ideation or design, require to include the students in the development and also to perform iterations design and to playtest in the design process. Due to time constraint, an appropriate amount of iteration and playtesting was not used; therefore poor implementation or inadequate balance in gamified exercise might have impacted the engagement of the students. Another limitation is that some of the OSINT requirements and techniques were not incorporated in the training lab.

Exercise labs were only given in one course, and therefore it was not possible to create control groups and deny some students of the significant gamification design elements or divide the class into traditional and gamified teaching. For example, denying the control group from specific design elements like a progress bar or mini-quests may impede the performance of the control groups, and therefore it would not be a fair study. Accordingly, this study was unable to show which of the exact game design elements had a positive or negative impact on the performance.

Lack of budget, inability to use proprietary scripts or virtualized Microsoft Windows operating system limited the options and solutions that can be used in designing the cyber security exercise labs. The other significant limitation was the sample size, which in this case was the number of students. If the number of participants is small, then it would be harder to make a meaningful relationship between the engagement level of the students and the gamified exercise. Another limitation of this research is that results could be affected by the novelty effect of the newly introducing gamified labs. It is also assumed that students in the case of playtesting and questionnaire answer truthfully.

2 Related Work

Gamification is defined as “the use of game design elements in non-game contexts” [3]. The primary goal of using Gamification is to improve performance and motivation in the given activity [4]. Various scientific studies have shown positive effects of using gamification, specifically in enhancing the engagement and motivation level of the participants [4], [5]. Business experts have asserted that more than 50 percent of the organizations will incorporate gamification in parts of their activities [6].

Various benefits for using gamification in education has been specified; gamification increases the motivation and engagement level of students, and gives students a chance for self-expression and control over their learning, and evokes emotions like passion, pride, and happiness [7]. Different terms have emerged in the game design field to define the game and education approaches; two of the common terms in this area are serious games and gamification [3]. Serious games are similar to gamification, but they are not synonym to each other. Both of these terms use gaming techniques for a non-gaming purpose; serious games are fully functional games to educate users, but gamification merely uses the game design elements in a non-gaming context [3]. Gamification uses part of the game, but gamification is not a game [8].

Some of the game design elements that could be used in gamification are badges, points, and scoreboard [9]. Kapp describes that badges, levels, points or scoreboard should not be the only game design elements to add to the games and this is a common misconception [8]. Kapp identifies two types of gamification, structural and content gamification [8]. Structural gamification does not apply the game design elements to the content, instead puts the game design elements around the content [8]. For example, by giving badges or achievement medals for watching a classroom video or for reading a book is a form of structural gamification. Content gamification applies the gamification design to the content as well; for example, adding storyline, goals and visual storytelling to a math training lab is a form of content gamification [8].

The goal of gamification is moving the player from the state of demotivation and not wanting to do something to a motivated state and performing activities or behaving differently [9]. Gamification studies have viewed motivation as a core part of gamification [9], [10]. Experts have identified multiple definitions for motivation, Bartol

and Martin and Farhad describe motivation as a drive for behavior, reinforces behavior and leads to continuing to do so [11], [12]. Osabiya defines the core of motivation as an impulse within the individuals which leads to striving for reaching a goal in order to achieve some expectation [13], [14]. Motivation is seen as two different types, extrinsic and intrinsic [9]. Extrinsic motivation focuses on using design elements like badges, points, scoreboard or money in a non-game context that stimulates behavior to perform a specific task; Therefore activity is done for attaining separable outcomes like rewards or grades [10], [15]. Intrinsic motivation, on the other hand, is an internal motivator that causes players to perform the tasks because of the activity itself and not external benefits; Therefore activity itself is perceived fun and long-lasting [10], [15]. Examples of intrinsic motivations are competition, sense of belonging, selflessness, love, passion, competence, and pride [16].

Reiss argues that splitting motivations into two extrinsic and intrinsic motivations is wrong [17]. Reiss states that psychologists have separated the motivations into survival based (physical) and psychological based(mental) category; but people are multifaceted and have different genes, needs, motivators, etc [17]. Reiss maintains that all motivations do have an internal source and motivators like money come from internally valued objectives [17]. Emphasizing on extrinsic motivations like rewards, points, leader board could have a negative impact on the intrinsic motivations, creativity and long term behavioral changes [15], [18]. This is due to the fact that after reaching the goals in the gamification or if the extrinsic motivations like monetary rewards stopped, it is possible to negatively affect the behavior of users and slow down their desire to perform as well as before [15], [18]. To address this issue studies have suggested to also use the intrinsic motivations in the gamification design to reduce or eliminate the possible negative effects of extrinsic motivation [15], [18], [19]. Intrinsic motivations also focus on integrating the 'flow' state. Flow state is the state of full immersion in the game, so that the user does not notice the passing of time and other external factors [20], [21].

Gartner states that the majority of the gamified implementations fails due to poor design [22]. These gaps come from the fact that gamification implementations contain only simplified and superficial manifestation of the game design elements like points, scoreboard, missions, and badges [23]. In these gamification designs, studies observed a short term increase in performance, but long-lasting motivation and engagement were missing in the common gamification designs [24]. Gartner stated that bad design was the

main reasoning for the failure of gamification projects and emphases on the importance of a strong design strategy to develop a useful gamification project [6], [22].

Increase in computer and internet usage have signified the importance of educating users, students and skilled employees in regards to cyberspace [25], [26]. One reoccurring theme in analyzing the outcome of emerging cyber threats and malicious activities is cyber education [27]. Higher education institutions have moved towards commercialization and requiring higher fees, and similarly, students are demanding a more hands-on education that guarantees their employability [28]. In general, lecturers are not well equipped with the latest techniques and practices of the professional field [28]. Therefore, curriculums and the implemented training do not cover the real world cyber-related skills in the ever-changing cyber field [28]. Training delivered by the professional providers are not the top priority of companies, and workplace training is typically targeted towards acquiring specific technical skills in regards to a product or system [28]. The methodical, holistic, affordable, fun and scalable solutions in cyber education are limited and professional training are generally not be implemented with an academic approach [5], [28], [29]. Different methods and techniques of cyber education have been introduced, but it is difficult to find the best and most suitable method of education for every case [26]. In recent decades, computer-based training has revolutionized the education field in improving collaboration and interaction [30]. Computer-based training or e-learning allows for a scalable, location independent and low budget education system [28].

Experiential Learning Theory can be applied to the education process to create a deeper e-learning training that is not just vocational [31], [32]. Experiential Learning Theory provides a learning process to achieve a holistic and mulileaner adult development [31]. Experiential Learning Theory demonstrates that learning is a process and not just the outcomes, and it is best occurs after connected experiences and personal feelings to form knowledge [31]. Studies have shown the usefulness of experiential and practical learning in improving performance and student satisfaction [33], [34]. E-learning is a suitable solution for heuristic learning [35]. Heuristic learning refers to self-determined learning which required learners to be independent and carry out problem-solving in discovering the solutions themselves as opposed to teacher-centric education method [26]. Computer-based training lacks in providing a social or personal education in comparison to classrooms; therefore a hybrid solution could provide a better learning experience that is

more interpersonal, holistic with a balanced ethical perspective for different environments [26], [28].

Gamification and practical training in education or the computer field are a new topic. Research has shown that students remember 10% of what they read, but they will remember 90% if students perform the activity themselves even if it is just a simulation [36]. Applying gamified education in the cyber security field is limited, but research has shown that applying gamification in the cyber security increases the interest of students in the cyber security field [37], [38]. Fouché and Mangle studied the secure coding education in the Code Hunt[1] platform and proposed a solution to broaden the audience of cyber security audience, but this solution was not practically implemented and tested [39]. Jin, Kim, and Tu developed an innovative game-based training camp for cyber security which was funded by a third party organization [38]. Their implemented game showed successful outcome in teaching the basic concepts of cyber security; but their study was done without analyzing the game design elements or the model [38]. Similar studies have also shown a positive increase in performance and interests of the participants by implementing a gamified cyber security training at work environment or university; but dissecting the game design elements, and game design implementation remains limited [40], [41].

[1] https://github.com/Microsoft/Code-Hunt

3 Design Method

To design a gamified cyber security training, the design method introduced by Morschheuser, Hassan, Werder, and Hamari, was the chosen method for this research. Hamari's Framework consists of thirteen design principle and seven design method phases [42]. This method was the result of interviewing twenty-five leading gamification experts and the systematic review of previous works. Table 1 describes an overview of this gamification design method.

Table 1. Summary mapping of the gamification design methods to design principles.

Method Phases	**Design Principles to consider for gamification projects**
Preparation	1. Identify and prioritize the objectives of the gamification. 2. Determine the applicability of the gamification. 3. Stakeholders must support the gamification. 4. Define the metrics for evaluating the effectiveness of the gamification. 5. Identify the requirements.
Analysis	1. Identify the user needs. 2. Define the context characteristics. (Platform, Technologies, constraints)
Ideas	1. Meet the needs of users in the design ideas. 2. Involve users in the brainstorming process.
Design	1. Perform tests as soon as possible. 2. Iterative design process. 3. Incorporate game design and human psychology knowledge. 4. Examine the legal, intellectual property and ethical issues. 5. Include users in the design phase.
Implementation	1. Frequently test and improve by using iterative design. 2. Enhance the design by constant use and playtesting. 3. Involve users in the implementation process.
Evaluation	1. Determine the effectiveness of gamification (playtesting, interview, survey, questionnaire, A/B testing, etc.)
Monitoring	1. Improve, optimize and re-design the game periodically if needed.

3.1 Preparation Phase

Foundations of Cyber Security course is taught annually at Tallinn University of Technology to the bachelor students. Preliminary objectives of the possible applicability of the gamification for this course were defined after meeting with the lecturer of this course.

Preliminary objectives and conditions of the gamified cyber security training in the order of importance are as follows:

1. Creating cyber security exercise labs from different topics of cyber security, for example, OSINT, SQL Injection, and Forensics
2. Exercise lab developers must have sufficient technical knowledge for timely implementation
3. cyber security exercise labs must be finished before the end of Spring 2019
4. Exercise labs must target users with basic knowledge of cyber security
5. Creating the labs in the TalTech instance of i-Tee [2] platform
6. Applying Gamification to the exercise labs via different game design elements
7. Successful-ness of the gamification needs to be measured by using the appropriate evaluation methods
8. Gamified labs should be reusable and documented
9. Labs must be customizable and preferably modular
10. Students should get immediate feedback on their progress and responses
11. For each student, a different answer must get generated to prevent possible cheating attempts

In Tallinn University of technology, there is a VirtualBox virtualization laboratory solution named i-Tee which allowed creating an automated virtualized environment for each student. i-Tee provides web-based remote access to a variety of the operating systems without requiring the user to install any software [2].

In the project preparation phase, the decision was made to develop the labs with high-level programming languages like Python, Ruby, SQL, PHP, and Bash. The source code is published in the GitLab repository of the TalTech [1]. The i-Tee platform was the chosen

[1]https://gitlab.cs.ttu.ee/

platform which is built on open source solutions [2]. Main stakeholders who are the lecturer, virtualization platform owner, and the university did not assign a specific budget for this project; therefore any purchases made were from the developer side.

3.1.1 Cyber Security Topics

Although multiple cyber security topics were supposed to be developed, due to heavy time constraints only two items were chosen, Open Source Intelligence and Reverse Engineering. More detailed reasoning in choosing the cyber security topics were mentioned in the ideation phase section (see section 3.3).

OSINT refers to intelligence that could be gathered from publicly available resources such as social media, Google, public and database [43]. Reverse Engineering in cyber security is defined as the process of analyzing the components of the software, reconstructing the relationship between parts and representing the software in a higher level language [44].

After extensive research on the available OSINT cyber security exercise labs, a minimal amount of gamified training was found that covered the OSINT topic, and if successful this exercise lab would be a comprehensive exercise lab dedicated to OSINT to educate the users and teach the core OSINT techniques in a gamified environment. The one similar exercise lab to the proposed OSINT topic in this research was a collection of cyber investigatory labs by Immersive Labs[1] which has only covered a limited amount of OSINT techniques. One of the goals for this lab was to cover the OSINT requirements more broadly.

3.1.2 OSINT Requirements

Users would have the opportunity to get familiar with the most used OSINT techniques. Choo and Quick have identified the core elements of OSINT [45], [46]. The following OSINT requirements are summarizing the main OSINT techniques:

- Learn to use a search tool: Users should learn how to use search engines and the search capabilities of the different resources; the most used source for this requirement is google.

[1]https://immersivelabs.co.uk/

- Extended search techniques: Google search techniques and google dorks should be taught to the students and be used to satisfy the needs for this requirement.
- Searching for deep web resources: Students should get familiar with deep web resources such as Shodan[1] , Wayback[2], TinyURL[3] or a comprehensive tool such as Maltego[4] which is a popular tool to use in OSINT [47].
- Review social media sites for valuable information: Students should learn how social media sites can give out useful information about its users and different methods for looking for targets in social media.
- Whois: GDPR is going to hinder fighting cybercrime in OSINT operations by forcing the WHOIS database owners to hide the confidential data about the individual domain owners [48]. Nonetheless, students should learn what Whois is and what kind of information they can gather from it.
- IP addresses: Students should learn to gather IP addresses from Shodan, Wayback machine, Nmap and email headers.
- Search for names, emails, EXIF data and breached email databases like HaveIBeenPwned[5]: In this exercise, multiple social media accounts were used in a way that students can use Google search engine to find social media accounts, use EXIF Meta viewer to see the hidden content inside images to find valuable information and earn points.
- Collect, Store and documentation tools: One of the crucial parts of any investigation is the documentation and data collection part, this is a requirement which may get overlooked, but it's a required technique for cyber security experts to incorporate.

[1] https://www.shodan.io/

[2] https://archive.org/web/

[3] https://tinyurl.com/

[4] https://www.paterva.com/

[5] https://haveibeenpwned.com/

3.1.3 Reverse Engineering Requirements

Unlike other cyber security topics, Reverse Engineering is a difficult subject to learn [49]. The following requirements were chosen to help students that do not have any previous knowledge of the assembly to learn the basics of Reverse Engineering:

- Learn how to use a decompiler: Students must learn how a decompiler works and practically use one to understand how it might help to reverse engineer a program.
- Learn the basic of using a disassembler: Students should be able to use a disassembler merely to understand the flow of code and reverse engineer a simple function.

3.1.4 Metrics

From the perspective of the participants, exercise labs were designed as a flag based lab. Flags are in a specific format that students must find and receive the points for submitting them. Training questions and answers can also be designed as a secondary progress metric, in summary:

- The progress of the students in the exercise labs was measured by the scoreboard and correct submission of flags.
- In training, students could have multiple quiz questions that require them to find the solution to individual questions and earn more points and progress in the game.

For this thesis, multiple evaluation metrics were applied in this research. Gamification experts have described the evaluation methods that they have employed in their gamification evaluations, these evaluation methods are expert interviews, questionnaires, usage data analysis and most importantly playtesting to evaluate the effectiveness of a gamification project [42]. Various methods were selected to obtain a better understanding of the research problem and examine the results. This will help conclude if the overall implemented gamification improved the performance of the students and if specific game design elements had any impact on the performance of the students. The chosen methods for this thesis were a questionnaire, A/B testing, expert interview, and assessment according to the gamification framework Octalysis. These metrics are described in more details in chapter 4.

3.2 Analysis Phase

According to the design principles of the analysis before performing a gamification design an extensive understanding of users and their technological needs is required. Gamification design model has multiple steps to conduct the analysis such as identifying target users, their needs, motivations to create different personas and profile for those target groups. The expert interviewee is Sten Mäses, the lecturer of the 'Foundations of Cyber Security' course and an expert in the chosen gamification platform area. After an initial interview for one hour these requirements were selected:

1. Target users have a basic knowledge of cyber security, and it is assumed that some of them might be more advanced in the cyber security field, but also skilled users should also find the training exercises educational, challenging and exciting enough.
2. Users expect to learn and test their knowledge in a practical way about the basics of the chosen cyber security.
3. Users should be able to run the cyber security platform with minimum interactions from their web browsers.
4. Since students are also classmates, there is an increased chance in communications and cheating amongst the students.
5. If there is scoreboard in the gamified lab, some users would not want their real name to be on the scoreboard.
6. Gamified exercise labs should be developed using i-Tee platform [2] as the basis".
7. Gamification of the platform is not heavily supported. Therefore the developer should have the technical knowledge and expect to solve the possible fixes.
8. Due to budget limitations, no commercialized software that required purchasing should be used.
9. Scripts and tools can be incorporated in the lab if it's acceptable by their license and terms of use. i-Tee platform is available under MIT license, which gives full permission for usage with no liability as long as the same copyright notice and license is included in the project [2], [50].
10. Labs automation could be done via Bash and scripting languages like Python or Ruby.
11. The training environment is running on virtualized Linux, and therefore developer's knowledge of the Linux system administration is required.
12. The developer should use PHP, HTML and similar web languages to set up the user interface.

3.3 Ideation Phase

This phase is the initial phase of design which focuses on having a creative process. This phase is about brainstorming different ideas with the stakeholders. Creativity plays a vital role in the ideation phase to create multiple design scenarios and narratives. Forcing the reliance on a specific framework in this phase might prevent the creativity level of this research gamification design; therefore, implementing platforms which might require complex modifications and extensive time was not possible.

In the initial ideation phase, a few discussions between the lecturer of the course and two cyber security masters student took place. The following were the initial ideas for gamification design:

1. Multiplayer training exercises which students require to cooperate and exchange information with each other.
2. Creating real-world hacking and exercises similar to Trace Labs[1]. Trace Labs is a Crowdsourced OSINT for Missing Persons; participants try to locate the real missing person by using OSINT techniques.
3. Students suggested not to choose SQL Injection or other well-known cyber security topics as they have been overdone in many gamified trainings and they showed interest in playing a gamified OSINT lab.
4. The scoreboard is a good idea, but it should not contain the name of users, both students thought having a low ranking in the scoreboard is embarrassing, and users should be able to stay anonymous in the scoreboard.
5. Exercise lab should give new and more advanced levels automatically.
6. Training should be accessible enough for everyone with basic cyber security knowledge.
7. The initial conversation with the students showed that they are not familiar with Linux or terminals in general, so the gamified lab should also teach about the basics of Linux and bash as well.

[1] https://www.tracelabs.org

8. Participants should have the freedom to make different choices in the gamified training. For example, if they can choose their own personas, or have the option to choose the subtasks or having the ability to solve the tasks via different tools or skills.

The current setup of the i-Tee [2] platform did not support any multiplayer feature. Some of the initial ideas looked promising, but not all of the ideas could have been implemented in time to deliver to the students for the spring of 2019. OSINT topic was considered to be a rarer topic in comparison to other cyber security topics. Therefore it might have seemed more exciting and appealing to the users.

3.3.1 Gamification Features

Nah, Telaprolu and Zeng conducted a literature review on gamification in the subject of education and learning and came up with multiple game design elements that have been used extensively in gamification to increase the quality and quantity of the education [51]. According to Nah, Zeng and Telaprolu's review the widely used game design elements are as follows [51] :

- Points and Progress bar: Points could be in two formats; they could be like experience points to show the improvement of users or in-game currencies to invest and buy more items. Using points users can track their progress[51], [52].
- Levels: Levels are used in gamification implementations to bring the feeling of advancements to the users. In gamification designs, the initial levels tend to be easier. But levelling by itself as a game design element may not improve the learning outcome of the students, the design of the level in the game should correspond with the long term goals of the users. [53].
- Badges: Badges or symbols are like elements similar to its usage in real life; badges mark the appreciation for achieving a goal or doing a task [51]. Badges increase the engagement and motivation level of the users [53].
- Scoreboard: Scoreboards are a typical game design element for creating a sense of competition and pride amongst the users [51]. A case study by Gain, Mariasshow, and O'Donovan show that scoreboard had the highest influence in increasing the motivation of the students [54].
- Awards: Rewards and prizes is an effective method for increasing the motivation of users; typical good use of the awards is designing small rewards instead of big rewards to have a better effect on users [51], [55].

- Storyline: Storyline helps learners to get more interested and stay motivated in the scenario [51]. The storyline also assists in teaching how a gamified exercise could be applicable in a real-world situation [51], [54].
- Feedback: Instant, numerous and clear feedback improves the engagement of the participants and allows for students to immerse in the game and don't feel the passing of time [20], [51], [56].

3.3.2 Octalysis Framework

Hamari states that nearly all of the expert interviewees reported using a gamification design framework for design and choosing the appropriate gratification techniques [42]. Studies show that only a few gamification frameworks exist [5], [57]. One of the most prominent gamification frameworks is the Octalysis Framework. A review of the gamification design elements that have been selected for this research based on Yu-Kai Chou's Octalysis Framework is discussed below [9].

As shown in Figure 1 [9] Octalysis framework is structured as an octagon, and it is composed of eight core human motivation drivers.

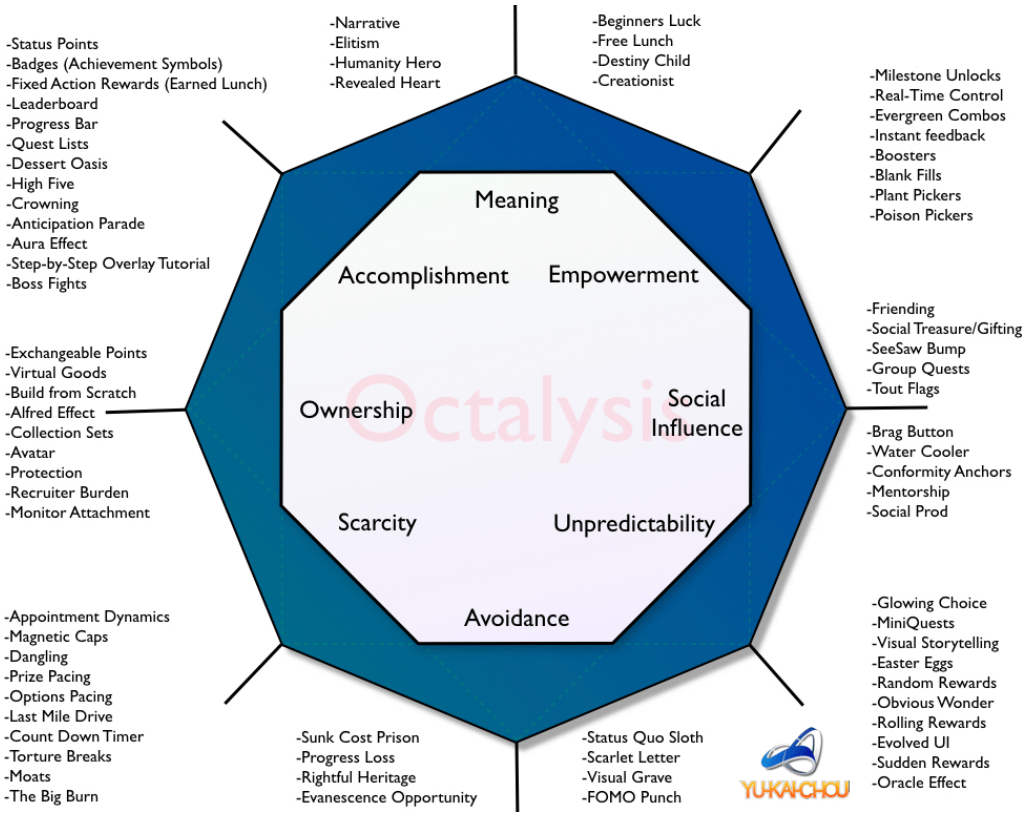

Figure 1. Octalysis Framework. Source copied from [9]

Chou states that most of the systems do not consider the human aspects and emotions in their design and focus on getting the job done, but instead in the Octalysis framework Chou emphasizes on human motivation instead of functions [9], [58]. The Gamification features chosen in this section were obtained from the Octalysis framework. Evaluation Phase is discussed in more details in the evaluation of the Octalysis framework and game design elements.

The eight core drivers(CD) of human motivation based on the Octalysis framework are as follow [3], [9], [59]:

1. CD#1 - Epic Meaning & Calling: When players believe that they are part of something bigger than themselves and users feel have been chosen to be part of this greater mission.
2. CD#2 - Development & Accomplishment: This is the most used driver in gamification containing badges, points, and the leaderboard. This core driver relates to the feeling of progress, developing the skills and leveling up.
3. CD#3 - Ownership & Possession: When users feel they own something, like virtual goods or avatars, this motivates them to get more possessions and protect their assets.
4. CD#4 - Empowerment of Creativity & Feedback: This core discusses if users are using creativity and feedback and designing different approaches to solving the challenges and receiving feedback for their solution to the problems.
5. CD#5 - Social Influence & Relatedness: This core motivator depends on social pressure and how friends perceive users
6. CD#6 - Scarcity & Impatience: It's a core driver of what users cannot obtain right away. Two typical methods to achieve this is enforcing limiting time or limited amount of resources.
7. CD#7 - Unpredictability & Curiosity: it plays with the user's desire to face the unknown and face the unknown [60].
8. CD#8 - Loss & Avoidance: This core refers to the avoidance of an adverse outcome and the possibility of losing if users do not act.

Left and right side of the Octalysis framework also has different meanings. Left side focuses on extrinsic motivators similar to the left side of the brain, and right side of the framework focuses more on the intrinsic motivators like the right side of the brain [9]. Activities in the gamification designs that are left-sided are more goal oriented, and users

tend to rely on extrinsic motivation to obtain something which could be a goal, goods or points. On the other hand, intrinsic motivators mean the activity itself is rewarding enough. For example, if an Octalysis graph of a gamification design is more lenient towards the right side, it means that motivations in that design are more towards activating the intrinsic motivations.

Octalysis Framework also separated the framework into two different sections, black hat and white hat [9]. White-Hat's engagement motivators are CD#1, CD#2, and CD#4 which focuses on positive emotions like developing skills, creativity and feeling an epic call, similar to the game design elements of the core drivers. Black-hat motivators are the remaining core drivers which emphasize on negative emotions such as fear of loss, lack of time and uncertainty. For example, if an Octalysis graph of a gamification design is towards the top, it means that motivations in that design are more towards activating the white hat behavior.

Table 2 displays the chosen gamification design element based on the Octalysis Framework. Twenty game design elements were selected for the OSINT and thirteen game design elements for the Reverse Engineering (RE) lab.

Table 2. Chosen gamification elements using Octalysis Framework

ID	Game Design Element	Core Driver	Training Lab
1	Narrative: Context for why to play the game	CD#1	OSINT, RE
2	Humanity Hero: Save the world, be a hero	CD#1	OSINT, RE
3	Elitism: Being Part of something elite, Train and Learn cyber security	CD#1	OSINT, RE
4	Free Lunch: Getting the flags for free to bring users in	CD#1	OSINT, RE
5	Status Point: Keeping the Score	CD#2	OSINT
6	Badges: Status Boxes	CD#2	OSINT
7	Progress Bar: To see where they at every given moment, Flag Counter	CD#2	OSINT
8	LeaderBoard: Having a scoreboard	CD#2	OSINT
9	Quest List: List of tasks that needs to be done	CD#2	OSINT, RE
10	Win Prizes: What to win from doing the game which is the grade and learning real skills for both labs and also scoreboard for OSINT lab	CD#2	OSINT, RE

ID	Game Design Element	Core Driver	Training Lab
11	Step by Step Tutorial: Only in the case of RE	CD#2	RE
12	Boss Fights: Having a final mission, in case of OSINT finding the hacker and in the case of RE solving the final task.	CD#2	OSINT, RE
13	High-Five: Getting the reward for correct behaviors	CD#2	OSINT, RE
14	Virtual Goods: Status and number of flags for OSINT. Decrypted files and password in RE	CD#3	OSINT, RE
15	Avatar: Pseudonyms	CD#3	OSINT
16	Collection Set: Status map in OSINT	CD#3	OSINT
17	General's Carrot: User is given a variety of tools and methods, now the user needs to strategize	CD#4	OSINT
18	Boosters: Training section in the OSINT	CD#4	OSINT
19	Instant Feedback: Instant feedback of submission	CD#4	OSINT, RE
20	Real-Time Control: They have full control over what they want to do	CD#4	OSINT, RE
21	Count Down: One week limit for each lab	CD#6	OSINT, RE
22	EasterEggs: Only in OSINT	CD#7	OSINT
23	Mini Quests: Training Quizzes in OSINT	CD#7	OSINT

3.4 Design Phase

Design phase focuses on creating an early prototype. Design and implementation phase was one of the most time-consuming stages of gamification designs. Design and implementation phase for the OSINT lab of this research took five months. Reverse Engineering lab was based on the OSINT template design and took less than one month to go through the design and implementation phase.

3.4.1 OSINT Design

The storyline and the game narrative for the OSINT exercise lab were decided as a mission for finding a hacker that has phished a victim. Students were supposed to use OSINT techniques and collect as much information as they could on the attack. In the end, users must be able to locate the hacker and obtain the attacker's name, picture, and location.

Security warnings and sending fake notifications are a common type of phishing attacks [61]. The lab's sample email was based on the phishing email that John Podesta the chairman of Hillary Clinton's 2016 presidential campaign received, causing him to fall into the trap of hackers [62]. Figure 2 [63] shows the email that is released by WikiLeaks which contained a malicious bil.ly link to hide the real destination of the malicious link [62], [63]. A fake Gmail password notification email with a URL shortener to create a sample phishing email was generated and used.

> *From:* Google <no-reply@accounts.googlemail.com>
> *Date:* March 19, 2016 at 4:34:30 AM EDT
> *To:* [redacted]ta@gmail.com
> *Subject:* *Someone has your password*
>
> Someone has your password
> Hi John
>
> Someone just used your password to try to sign in to your Google Account
> [redacted]@gmail.com.
>
> Details:
> Saturday, 19 March, 8:34:30 UTC
> IP Address: 134.249.139.239
> Location: Ukraine
>
> Google stopped this sign-in attempt. You should change your password
> immediately.
>
> CHANGE PASSWORD <https://bit.ly/1PibSU0>
>
> Best,
> The Gmail Team
> You received this mandatory email service announcement to update you about
> important changes to your Google product or account.

Figure 2. Phishing email sent to John Podesta. Source copied from [63]

Creating one gamified cyber security exercise to cover all of the OSINT requirements was not feasible for this thesis; therefore a section was added in the exercise lab to cover the secluded areas, for example by having a Q&A or quiz questions. It was decided to design a quiz section in the lab named 'training section' so students would be able to answer the questions and receive points. These questions should be aligned with the direction of the primary mission to help students find the hackers.

The first iteration concept was designed as setting up a few real websites. Each website had one subdomain for each student which was populated with exercise's fake

information and pushed into publicly available sites like Wayback and Shodan with flags for students to find. The exercise lab automatically assigned a random subdomain for users. The reason for having one subdomain for each user was to reduce cheating attempt and to have better management and performance monitoring. Researching the social media sites and populating data to Facebook as a sample social media site, revealed that it would be impossible to create a separate social media profile for each student to satisfy both of the requirements for using real websites in the exercise lab and having a custom flag for each user.

The second design concept was developing a simulated version of top tools and websites instead of populating data into a real website. Simulating the actual website in the lab was a drastic change in the gamification design. To create a prototype, some simulated sites had to be built like Shodan, TinyURL, and HaveIBeenPwned. TinyURL was chosen as a test project. TinyURL is a link shortening service that provides shortened links for URLs. Link shorteners are commonly used by an attacker to spread malicious links [64]. Initial development for simulating TinyURL revealed that this design concept is not an acceptable one. The following issues were identified in choosing this design concept:

1. Lack of free and non-proprietary tools and scripts to build the simulated sites
2. Creating simulated sites for multiple complex websites such as Facebook and Twitter would take a great deal of time.
3. Simulated sites cannot be as sophisticated and user-friendly as the real site on the internet.
4. The user will only learn how to work with simulated tools and not the real sites that are investigated in real-life cyber security investigations.

To address these issues, a combination of techniques was chosen to simultaneously allow students to use real-world websites and modify the traffic on the fly. Each user had to initialize their own set of virtualized machines. For this research's setup, the exercise lab was constructed out of two virtual machines, an Ubuntu-based desktop machine for users and a Linux machine acting as the training panel and proxy server. User's traffic to the Internet is captured on the Linux server, and customized HTML elements were inserted to the responses. This approach made it possible to, for example, embed a flag inside a profile link in Twitter's traffic, and users should be able to find the flag only if they visit the correct profile link.

Preliminary research for this design concept showed that the following techniques would make it possible to modify the user's traffic without disrupting the normal flow of browsing experience of the students:

- Self-Generated Certificate Authority: Certificate Authorities (CA) issue SSL certificates for websites and browsers use Certificate Authorities to verify the website's identity and secure sockets layer (SSL) certificates [65]. Self-generating CA is a common practice by corporation and anti-viruses to monitor the victim's activities; this is an also a known method employed by hackers to install a fake CA in victim's machines to perform a man in the middle attacks on their victims [66], [67]. This technique allowed us to issue fake SSL certificates for the chosen public websites and avoid showing a warning message of insecure communications to users.
- Reverse Proxy: Reverse Proxy is a type of proxy server used in Load Balancing and cache optimization and intrusion detection by retrieving the resources on the client's behalf [68]. Reverse Proxy could also be used with malicious intent to hide the back-end server from prying eyes [69], [70]. Implementing a reverse proxy in the exercise lab, allowed to retrieve the HTML responses that have been redirected via DNS cache poisoning. Reverse Proxies can also modify the received content [71], [72]. This feature allowed us to alter the responses and put flags and monitor the user activities if needed. Appendix 3 shows the configuration used in this thesis to achieve this technique. Users must find as many flags as possible; each flag earns a fixed point. Flags consist of sixteen random characters, and they were automatically generated after each lab initiation. Flags were put in places to ensure users will get a reward for correct behavior and looking at the right place. Flags were typically included in the HTML response, if users investigate the URLs outside of the lab environment, they only see a normal response without any flag inside of it, or they will see a random string. Reverse Proxy replaced these random strings with the real flags. With this technique, it can be ensured that users should receive a different flag and they can only find the flags if they browse the URL on their lab instance Appendix 2 shows the script used in this thesis to attain this goal.
- DNS cache poisoning: DNS cache poisoning or DNS spoofing is one of the most prominent DNS attacks where DNS resolver stores an invalid record for the DNS

queries and subsequently domain maps to a malicious IP address [48], [73]. Virtual machines of the exercise labs were in full control of the lab developer. Consequently, it was possible to deliberately resolve the DNS queries of users to an internal server instead of their original IP. Figure 3 illustrates a summary of this design technique used in the lab. An example script on how the implementation of this technique was created in the OSINT lab is shown in Appendix 1.

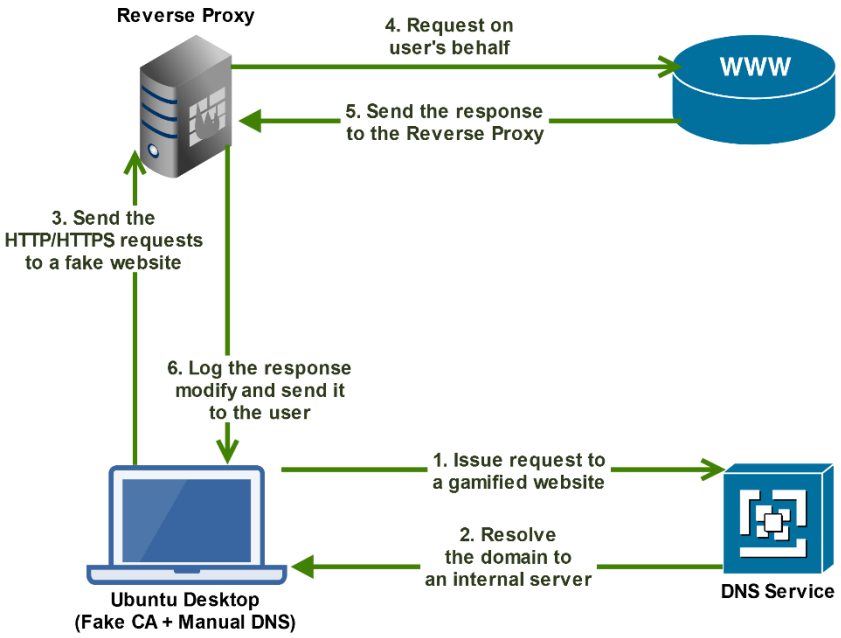

Figure 3. Modifying user traffic using DNS Cache Poisoning

- Redirecting HTTP and HTTPS traffic: In the initial prototype DNS Poisoning were translating the domains to an internal IP. Subsequently user's machine receives a fake IP address instead of the real address. In the OSINT lab requirements, users were supposed to investigate the IP address, but DNS poisoning prevents students from accessing the real mapped IP address of the domain and consequently examining the IP address if needed. To address this limitation, iptables can be used to change the destination of packets and eventually redirect HTTP and HTTP traffic to a different IP [74]. Redirecting users without DNS poisoning allows users to see the real IP address of the destination via regular DNS queries, but the lab server redirects the traffic to an internal web server without notifying the users. Figure 4 shows a diagram of how modifying user traffic using iptables works.

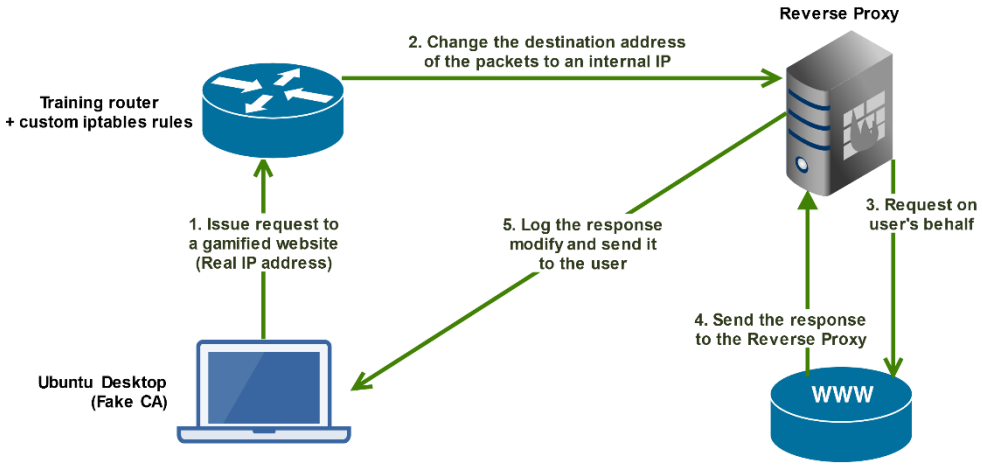

Figure 4. Redirecting and modifying user traffic using iptables

3.4.2 Reverse Engineering Design

The Reverse Engineering lab was designed and built off of the OSINT exercise lab. Reverse Engineering lab was heavily based on the OSINT lab; therefore it was much easier to rely on the success of the previous. The design phase of the Reverse Engineering lab took around three days. The overall design of the Reverse Engineering lab consisted of 2 main tasks:

- Reverse Engineering a Ransomware: Students were provided with ransomware written in Java and were asked to decompile the program and decrypt the files that have been infected by this malware.
- Finding the unique strings in the executable files: In Reverse Engineering training there are challenges called CrackMe which require users to reverse engineer the program to extract a particular text within them [75]. In the lab's problems, the unique strings as passwords were designed, these challenges developed with incremental difficulty, and with each level, the strings inside the program become a bit harder to find.

3.4.3 Legal and Ethical Constraints

Gamification design method was proposed by Hamari, Hassan, Morschheuser, and Werder; this design method highlights the importance of ethical and legal constraints in gamification designs [42].

There are a few ethical issues that must be addressed before implementation and deployed in training. These ethical and legal constraints are as follows:

- There is a potential risk that cyber security exercise labs could encourage the student to use these techniques maliciously. OSINT lab was designed as a training exercise where students assume they work for a company that has been targeted with a phishing attack and therefore they are allowed to investigate the incident and find the hacker using the OSINT techniques. In the Reverse Engineering lab, students were given malware for analysis and CrackMe challenges to recover the passwords of the files. Before deploying the training labs, students were briefed not to use these techniques unethically.
- An inquiry by the lecturer of the course from data protection officer of the university revealed that if performance and questionnaire results were anonymized, therefore there was no need for explicit permission from each student to use their anonymized information in the lab.
- Development should only be done with the help of scripts and tools if their license allows to use them for free in the creation of the exercise labs. Otherwise alternative codes must be used or developed in house.
- In the OSINT exercise lab, one of the most significant ethical constraints to address was the use of public information and creating fictional characters in the exercise lab. To construct names and personas, random name generators like Fake Name Generator[1] were used for non-malicious and random word generators for malicious characters. The reasons for using random words to create malevolent personas is that giving random names to malicious characters will not associate the name to any ethnicity or race. Also, randomized names made it easier for the users to narrow down their search results and investigation in finding the attackers.
- In the 'training' section of the exercise, students must find information on breached emails and leaked password in databases like HaveIBeenPwned. Author of thesis used their domain and the breached fake accounts of that domain to address this issue. All of the breached emails in this domain belong to the thesis author and are false and are owned by the author. No other information of real people was used in the exercise.

[1]https://www.fakenamegenerator.com

- The last ethical issue that was identified in the design phase was the pictures used in the lab, to prevent usage of any real picture multiple photos were chosen that were labelled for reuse with modifications for proprietary purposes. Later on, photos were morphed into each other to create a none existing profile picture of malicious characters. Figure 5 shows the combined profile picture of multiple images with morphthing.com[1] .

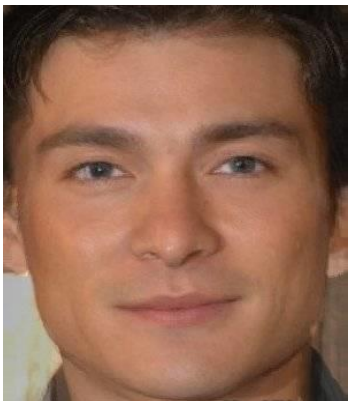

Figure 5. Morphed image used in the lab with morphthing.com

3.5 Implementation Phase

The implementation phase began with the development and coding of the exercise. For this research, the training labs were designed and developed by the author to the existing i-Tee [2] platform. The implementation phase was an iterative process of implementation and test to make sure implemented gamification is allowing us to reach the target goals [42]. The iterative procedure makes it possible to test, identify and fix the technical issues of each gamification design element as soon as possible [76].

3.5.1 OSINT Implementation

In the OSINT storyline, students were supposed to investigate a phishing attack and find the hacker. To create the user's web panel, free to use icons, graphics, and bootstrap templates were used. Figure 6 displays the initially chosen web panel for the OSINT lab.

[1]http://www.morphthing.com

Figure 6. Initial web panel of the OSINT lab using free tools

In the design phase (see section 3.4), the reasons for choosing the type of phishing email were described. To create a phishing email, registered email addresses were created and sent a real phishing email to the supposed victim, and modified the email headers to the correspondent to the lab's scenario.

Similar to the sample email phishing defined in the design phase a link shortener address in the lab's malicious email was embedded. To reward the students for investigating the malicious Link shortener, DNS Spoofing and Reverse Proxy were used as mentioned in the design phase (see section 3.4) to put a customized flag in the HTML responses. Figure 7 is showing an HTML response from bitly.com that has a customized flag inside of it. These flags were put in a hash mark(#) as a fragment URL, to ensure that embedded code did not destabilize the functionality of the real website [77].

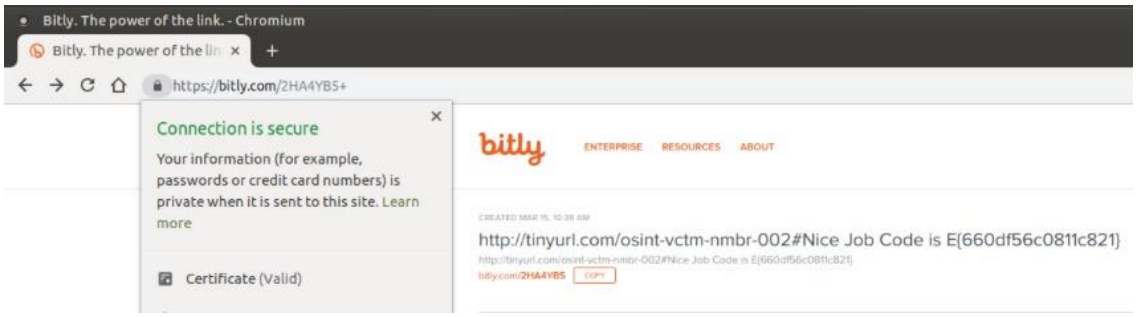

Figure 7. Inserting a customized flag inside an HTML response

The flags were chosen based on the OSINT core techniques described in section 3.1.2 of this research paper. In the implementation phase, twenty-six different flags were created for users to find. These flags have been inserted in pictures, app profiles, emails, ten public sites, and four in-house developed websites to simulate the sites that are owned by the hacker. Since flags are individual strings, it is easy to remove or add new flags to the

game in a modular manner. This makes it possible to change the location of the flags or customize them more efficiently.

This exercise lab had two main sections; the mandatory challenge is that users must find the specific number of flags (at teacher's discretion) to pass the lab and an optional lab which contains a different variety of quiz questions to teach the techniques required to find the flags to users.

25 different quiz questions were designed with two different types:

1. Some of the questions were created to teach the techniques required to investigate the incident, find the flags and the hacker. For example, users must be able to investigate the certificates of the attacker's domain and to help users achieve this a question was added to the quiz section and asked the students to examine the SSL certificates of another website. By answering the questions correctly, users get to learn the technique.
2. Remaining questions were not related to the flags, and those techniques would not help in finding the flags, but instead, it covers the other OSINT techniques, that phishing attack scenario of the OSINT lab was not able to cover.

Each question has a specific reward point from 5-15 depending on the complexity of the question. Hints also added to the question section, that shows the solution of the question. Students will lose the reward points if they choose to see the hints. Figure 8 is showing a sample question from a pool of questions; users randomly get the questions. Users can change the questions until they receive a question that is interesting enough for them to solve and learn from it.

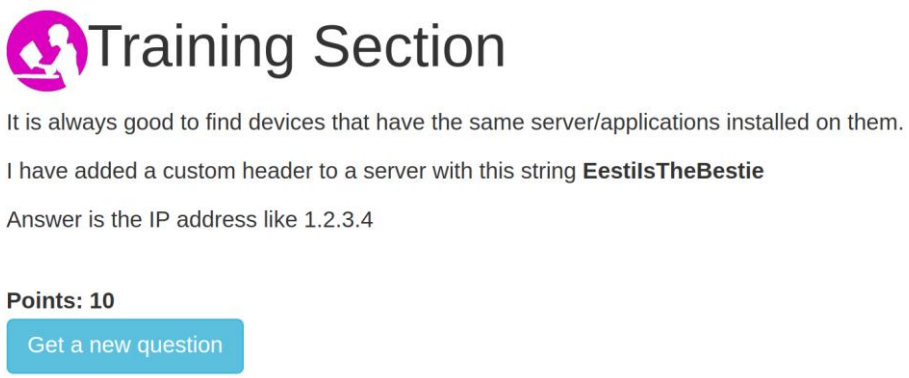

Figure 8. Example quiz question in the training section of the OSINT lab

Numerous playtesting occurred after each major implementation phase. Five students of TalTech Cyber Security master's program were asked to playtest after each significant iteration.

There was some notable issue that arose from observational playtesting. Initial playtesting showed that some players did not understand what they are looking for, what users are supposed to do or what the meaning of flags is. Some students have never participated in any Capture the Flag competitions. To ensure students could understand the lab better, lab guides were revised, and multiple instructions were put in different places to teach them precisely what they ought to do.

After revising the lab guidelines, another playtesting was conducted with a new set of students to verify the usefulness of the new guidelines, but observations showed students are not reading the instructions and they start doing the lab without carefully reading the lab guides. To make sure students can understand the game, a very easy flag to find was added to the lab; this way users can see an example of a flag that they should try to look for and find. The feedback message for the quiz questions and wrong flags was modified to specify to the students to pay attention to the questions and to try to use the quiz techniques to find more flags.

Third playtesting showed some improvement in understanding the game and finding the flags, but users who identified themselves as non-tech savvy only were finding one or two flags, and they could not think of places for the flags. To help users and to gamify the lab more a flag minimap of all of the flags and their relation to each other was created. The goal of using the map was so users could see their achievements after each progress. Users can check and see if their current flags could help them to find more flags or not. Figure 9 shows a portion of the map of flags and how they related to each other. The first box in Figure 9 shows the user itself, all of the other flags are hidden in the beginning after each correct submission of the flag users can see the name of the flag they found, and its relation to the other flags in the training lab. For example, by looking at Figure 9 user can deduce that they must retrace their steps to find the missing flag before discovering the Twitter profile of the hacker. It also shows that using hacker’s twitter profile they can find four more flags.

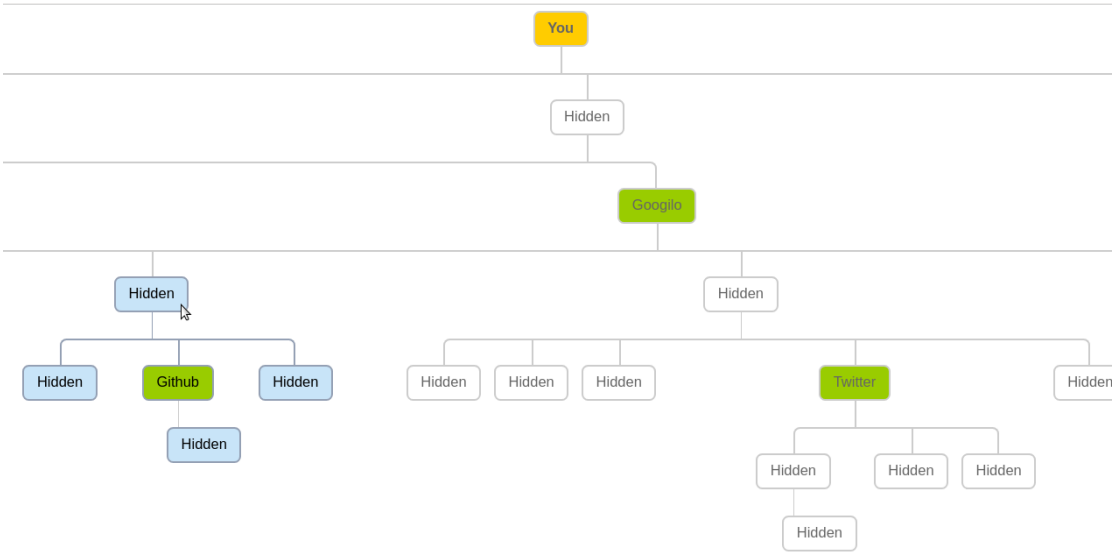

Figure 9. OSINT flag map to display the status of the achievements

Final playtesting showed that users were able to draw conclusions from the quizzes and the map section. Due to time pressure and the fact that playtesters didn't have time to find all of the flags, each playtesting occurred in 30-45 minutes of playing.

Table 3 shows the OSINT requirements that was covered in the main mission or the Quiz questions. The exact flags and quiz details are deliberately not shown to keep the flags and their placement out of public records.

Unfortunately, it was not possible to cover two OSINT requirements: Whois and documentation. It was not possible to populate false Whois information on the internet because research showed that inserting fake data into Whois is against the law and terms of use of domain registrars [78], [79]. Author of the thesis was not able to design a suitable scenario for documentation and data collection requirement of the OSINT investigation. Therefore documentation was not included in the lab.

Table 3. Overview of the OSINT requirement and their coverage by flags or quizzes

OSINT Requirement	**Number of Flags**	**Number of Quiz Questions**
Search Techniques	Two Flags	Seven Questions
Deep web Sources (Shodan, Wayback, etc.)	Five Flags	Five questions
Social Networks	Four Flags	Was not covered
Investigate IP address and domain names	Three Flags	Three Questions
Whois	Was not covered	Was not covered

OSINT Requirement	Number of Flags	Number of Quiz Questions
Personal Information (Name, usernames, Friends)	Two Flags	One Question
Breached Emails	Was not covered	Three Questions
Geo Location (GPS, Google Maps, etc.)	Two Flags	Two Questions
Investigate website contents (Source code, links, etc.)	Four Flags	Four Questions
Images, EXIF and file metadata	Three Flags	One Question
Data collection and documentation	Was not covered	Was not covered

Design, Coding, populating information about the hacker on the internet (personal websites of the hacker, social media and deep sites presence) took around 3800 lines of code in multiple programming languages in approximately five months. The main languages used in the creation of lab was Python, PHP, Ruby, Bash, HTML, and JavaScript. Plenty of implementation iterations conducted to fix the bugs and issues and adding the new flags.

3.5.2 Reverse Engineering Implementation

As described in the design phase, this lab consisted of two main sections Ransomware and CrackMe challenges:

To build a ransomware, a fake ransomware in Java was developed. Java programs are easy to decompile and reveal the source; this is because Java programs are in a form called Byte Code, which is closer to the source language than assembly [80]. JD-GUI[1] was preinstalled in the lab to help students decompile the ransomware without installing any decompiler. JD-GUI is a free and easy to use java decompiler. Figure 10 shows the ransomware for the reverse engineering lab; students must read the step by step instructions and decompile the program and decode the decryption code from source code.

[1] http://java-decompiler.github.io

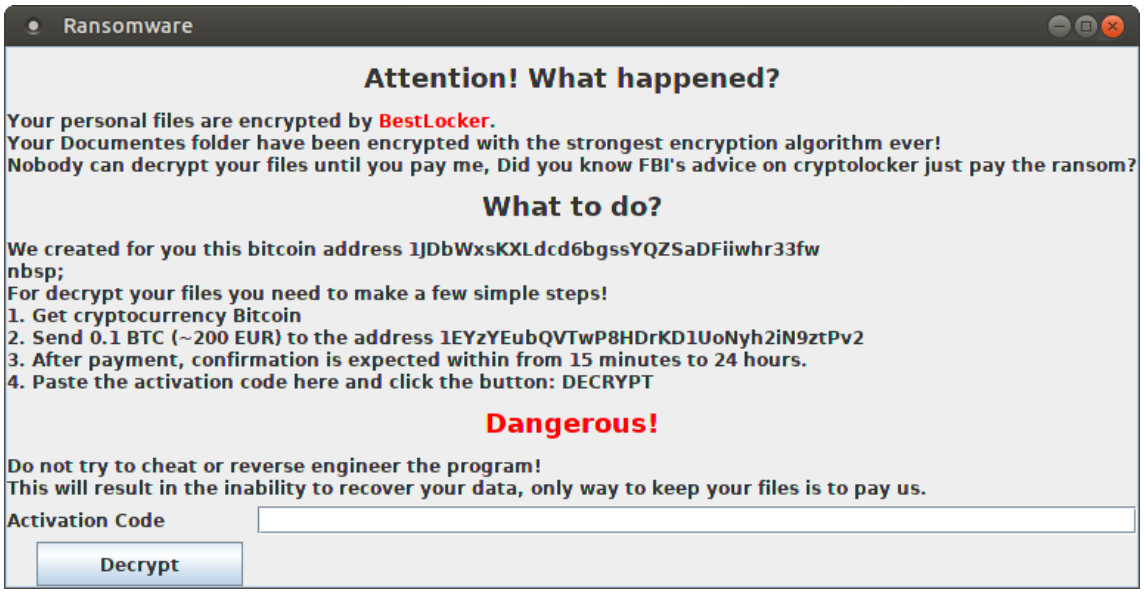

Figure 10. Simulated ransomware file for the Reverse Engineering Lab

The second type of tasks that users must do is to crack the six Crackme challenges written by the author in C language. A step by step guideline for the users was created to reverse engineer the program and find the input string that prints the successful message. In the CrackMe challenges, some strings are manipulated, for example, strings were reversed, capitalize; then users must reverse engineer the program to find out what exactly occurs in the program and what is the final correct string (password) after the conversion.

Radare2[1] is a free to use reverse engineering framework. Radare2 was preinstalled in the lab so users could use it to reverse engineer the program. Since assembly for users who do not have any previous experience in reverse engineering is a difficult task, a plugin was added to Radare2 to convert the assembly code to pseudo-code in C language. Figure 11 is showing an example of a CrackMe challenge used in the lab, that users must follow to figure out how the variable rax is getting converted to the final string.

[1] https://rada.re

```
uint64_t check5 (void) {
    char * s2;
    char * var_8h;
    rax = "VKCEDGKMYH";
    var_8h = rax;
    rax = *(obj.stdout);
    fwrite ("What is the password?\n", 1, 0x16, rax);
    rax = &s2;
    fgets (rax, 0xb, *(obj.stdin));
    rax = &s2;
    rax = strlen (rax);
    if (rax != 5) {
        eax = 0;
    } else {
        rax = var_8h;
        rcx = rax + 3;
        rax = &s2;
        eax = strncmp (rcx, rax, 4);
        if (eax == 0) {
            eax = 1;
        } else {
            eax = 0;
        }
    }
```

Figure 11. Example CrackMe pseudo-code challenge in the Reverse Engineering lab

Step by step instruction is given in the hints, and all of the strings in the Reverse Engineering lab are dynamically calculated after each lab initialization to ensure users cannot submit other user's answers as their answer. Implementing the Reverse Engineering lab was a simple process, reusing the OSINT lab codes made it possible to develop the lab in less than a month.

4 Results and Discussion

In this chapter, the results of the lab for the evaluation phase of the design model is measured and discussed. There was no control data for each game design element other than one experiment; therefore objective deductions for each game design elements are avoided. Instead, this research focused on subjective, questionnaire and performance results, and describe how the cyber security exercise labs performed in general and more significant and precise questions are left for future works.

4.1.1 Reverse Engineering Results Discussion

RE lab contained thirteen different tasks that were given to students. The last task was a bonus challenge which is a CrackMe task without any hint. Solving this task will give students a bonus grade for the class. In the live production of the training lab, no technical issue was observed other than a typo in the instruction portion of the lab that was reported by the students. Flags were dynamically generated for each of the users. Reverse Engineering implementation was not designed to notify the students of cheating attempts. Based on exercise logs, two students submitted the flag of other students to cheat the exercise. Reverse Engineering lab was given to a total of 100 students between the two groups. Each group had 6 days to finish the lab. 47 students participated in the first week and 43 in the second week's group. As shown in Figure 12 results from both weeks were similar. 62.5% of the students were able to finish the required tasks to get the full points for this lab.

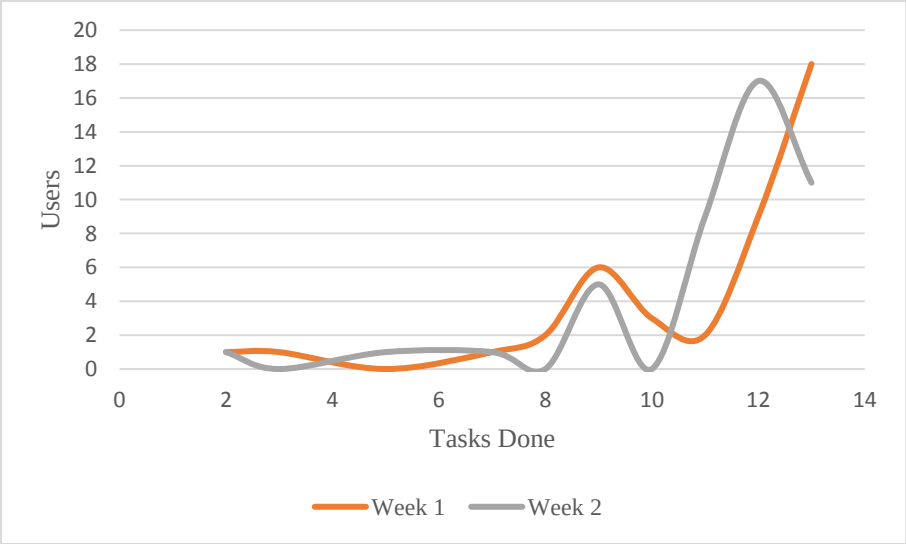

Figure 12. Tasks done by users in Week 1 and Week 2

One of the observations from the log analysis was the timeline of the correct submission of the tasks. This shows that students in both weeks were working on and submitting more on the last few days of the exercise. Submission on the last hours and days before the deadline could be seen as an attempt by students to finish the lab and get the grade and not be immersed in the lab.

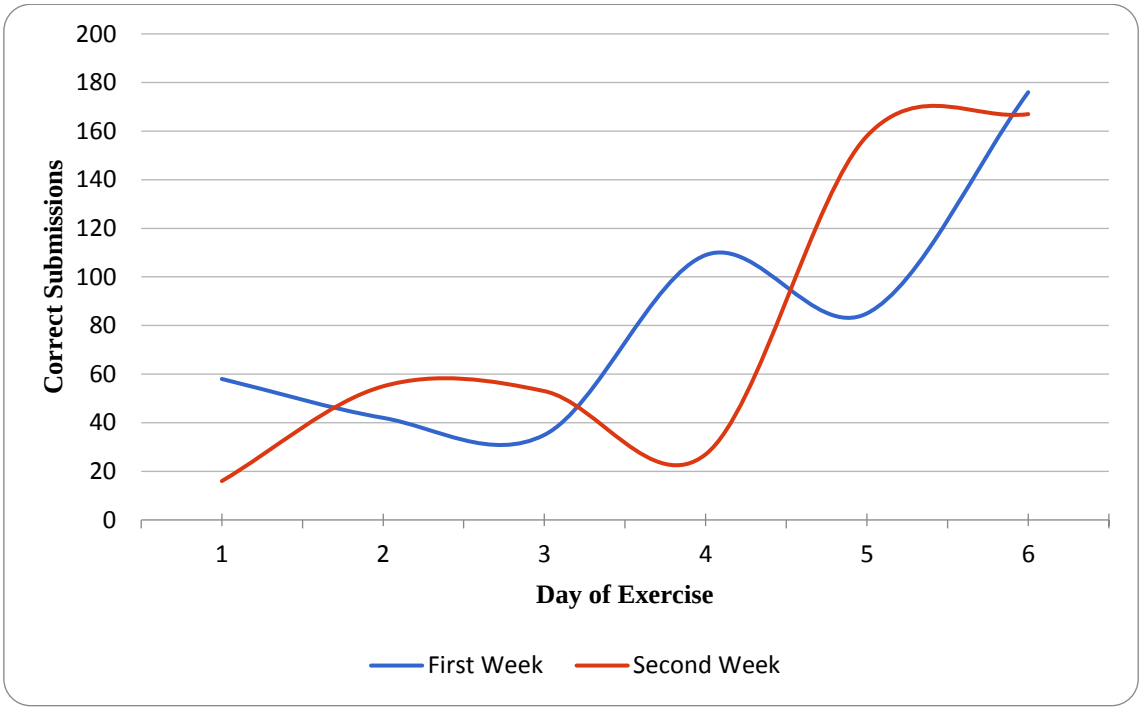

Figure 13. Correct submissions of the tasks

4.1.2 Reverse Engineering Octalysis Framework Motivation Assessment

Octalysis framework provides a metric system to measure how specific gamification fits into the Octalysis framework. Octalysis tool[1] creates a visual diagram of the core drivers based on the metrics given to it. Each core driver received a score from 0 to 10. The given metrics are discussed and evaluated by the lecturer of this course as an expert to measure and validate the metrics. Nevertheless, it should be noted that Octalysis scores are subjective by design and are therefore reflecting opinion, not an objective truth.

In Table 2, the chosen game design elements for each exercise lab was specified. The importance of the selected game design elements in their corresponding core drivers and discussion is as follows:

[1] https://www.yukaichou.com/octalysis-tool

1. CD#1 - Epic Meaning & Calling: One of the essential drivers in the RE lab is emphasizing on the student's will to be a hero and break the obstacles that malicious actors have put on their files. In the ransomware task students were supposed to decompile the ransomware malware and find the decryption key. In the CrackMe section, each program was an individual challenge to be cracked. There was no connection between Crackme challenges or with the ransomware challenge. Therefore in the RE lab, there was no universal narrative, but each task had its context and narrative. Another major game design element of the RE lab was elitism; students learned reverse engineering and how to break some programs practically. Pride in being able to learn reverse engineering was a powerful motivator with this gamification design. (Score: 7)
2. CD#2 - Development & Accomplishment: This gamification design emphasized for this core driver was step by step instructions, quest list and boss fight. The leaderboard was not used in this lab. (Score: 7)
3. CD#3 - Ownership & Possession: RE lab design did not cover the motivation elements of this core driver other than providing virtual goods for the students, but virtual goods was not an essential part of the lab. (Score: 2)
4. CD#4 - Empowerment of Creativity & Feedback: Instant feedback was an essential game design element in this design and students do have control in the program, and how it is run, each correct answer unlocks the next section and get a new challenge to defeat. (Score: 3)
5. CD#6 - Scarcity & Impatience: The only game design element used in this motivation driver was the count down which was the time limit of 6 days for students to finish the lab. (Score: 1)

Game design elements other than the ones mentioned in Table 2 were not used in this lab and received a score of zero.

In Figure 14 [81] the graph generated by the Octalysis tool is shown. Octalysis tool also provides an overall score which is the sum of squared scores of the core drivers; therefore for this lab, the overall score would be 112.

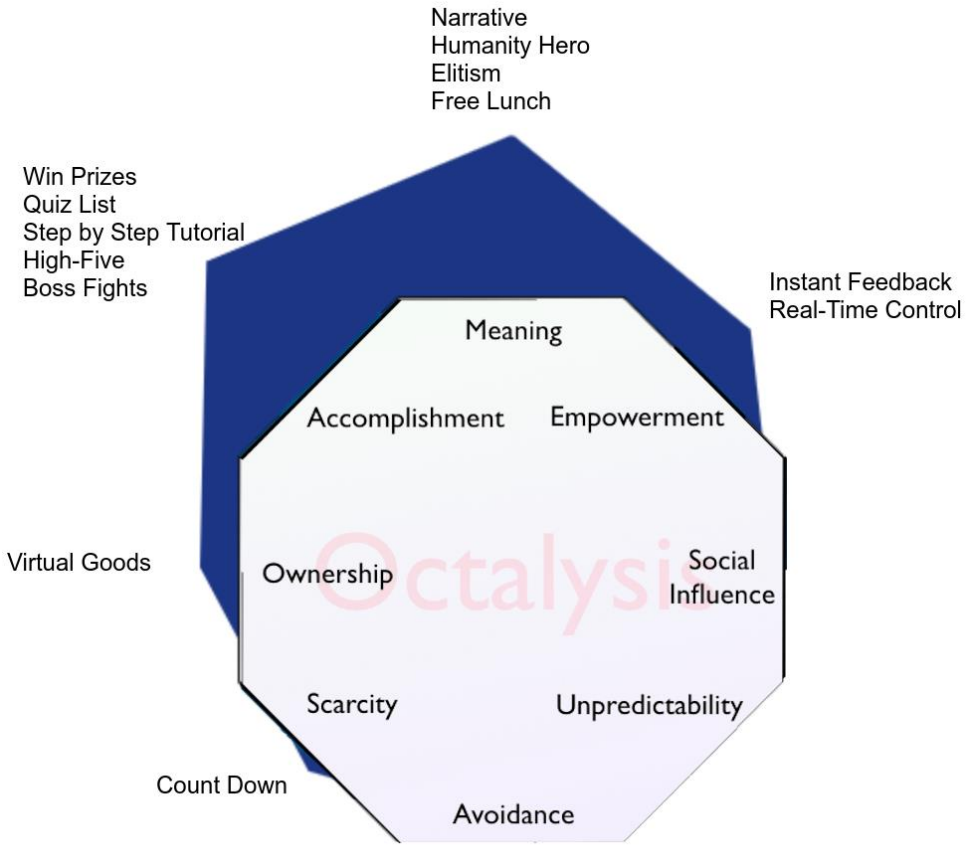

Figure 14. Octalysis Graph of RE lab. Source copied from [81]

As shown in Figure 14 [81] the graph is towards up and left which is an indication of successfully designing gamification what is oriented around the white hat behavior and activating extrinsic motivation. The pro sides of a gamification design such as this lab, are that it activates positive feelings of empowerment and successfully encourages the students for achieving a goal. But the shortcomings of this gamification design is that it fails to evoke long term motivation that could last after students have completed the training lab.

4.1.3 Reverse Engineering Button Text Influence Experiment

A/B testing is used by researchers to provide two different isolated variable in the same environment to test the effect of the isolated variables [82]. Reverse Engineering lab contains hints and users can see the hints by clicking on the hint button. To make the most use of the reverse engineering gamification design users randomly assigned to two groups, users in the control group had a hint button named ‘hint’ and users in the experimental group had a hint button named ‘free hint’. There was no other difference between the lab implementations of two groups, and only the button text was different.

Both of the hints would not cost any points from the students. The hypothesis for this experiment is that users will click more and sooner on the hint button if the text reads ‘free hint.’ Essentially users take the easy route if they want a way to get out of doing the task.

In the Reverse Engineering lab students in the experimental group who received the button named ‘free hint’ had more clicks in general as seen in Figure 15. This figure shows the increase in the number of clicks by average, median and overall behavior of these two groups. Hint button text which read ‘free hint’ might be seen as encouraging the students to click more on it as it says ‘free’ or students might think they will be receiving free or different hints that other users. Students in the experimental group overall clicked 14 times on average on the hint button per user and with a median of 13 clicks. Students in the control group overall clicked 10.5 times on average on the hint button and with a median of 10 clicks per user. Using the button with ‘free hint’ showed a 33% increase in average clicks per user and a 30% increase in the median clicks per user.

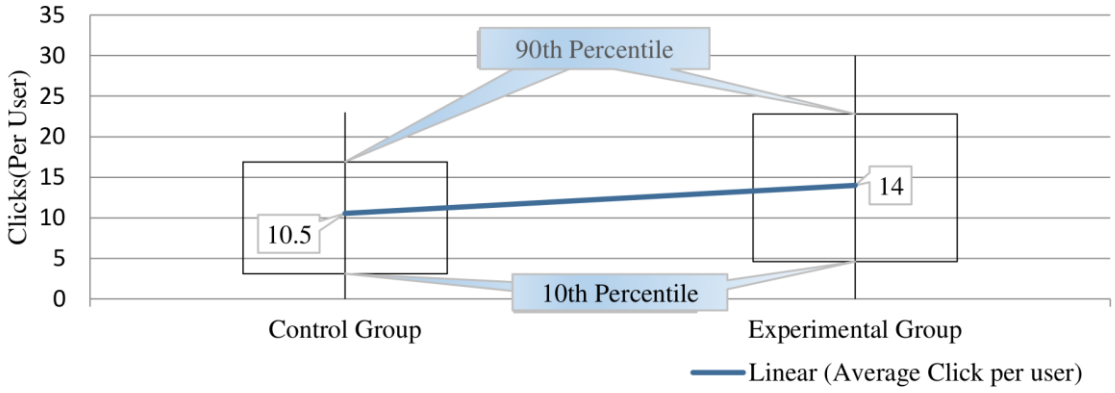

Figure 15. Candlestick chart of average clicks on the RE hint button per user

Also in the Reverse Engineering lab students in the experimental group who received the button named ‘free hint’ after receiving the task waited less and clicked more quickly on the hint button as seen in Figure 16. This figure shows the less waiting time on the average, median and overall behavior of these two groups. Hint button text which read ‘free hint’ might be seen as encouraging the students to escape the frustration of trying to solve the task and click faster on the hint button. Students in the experimental group waited 80 seconds with a median of 39 seconds after they received the task before clicking on the hint button. Students in the control group waited 150 seconds with a median of 100

seconds after they received the task before clicking on the hint button. Using the button with ‘free hint’ showed a 61% decrease in median waiting time and a 47% decrease in the average waiting time.

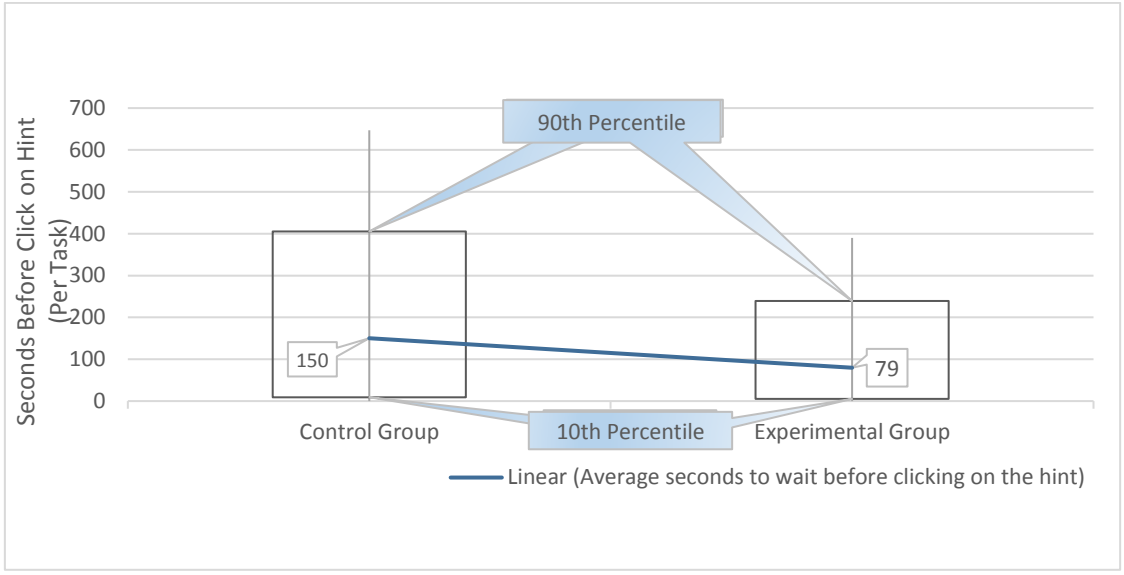

Figure 16. Candlestick chart of average waiting time before clicking the hint button

As this experiment validates the hypothesis, it shows the importance of effect and interpretation of the design element text within a gamification design. By using a more natural or even positive reinforcement, users might refrain from using hints and be more emerged in the lab itself. Positive reinforcement in this example would’ve been seen as a button which displays a message box that shows a positive message to students and encourages them not to view the hint text. This hypothesis and other A/B testing experiment could be done in future work.

4.1.4 OSINT Results Discussion

94 student participated in the OSINT lab. Similar to RE lab students were split into two groups to do the lab in two weeks; each group was given one week to complete the lab. Overall there were twenty-six flags in the lab and finding three flags would have given one point, six flags two points, nine flags three points, fifteen flags four point and finally twenty-two flags five points. The top three performers of each week received an extra point for the course. To earn points, student should find flags or answer the quiz style questions without any hints.

Since flags that students were able to find were scattered between 1 and 22, multiple graphs were used to show how students performed in each week. The number of flags found per users is shown in Figure 17, as it clearly shows students were able to find much more flags in the second week's group. One possible reason could be the fact that students were using the hints and location of flags that have been found by the first week's group.

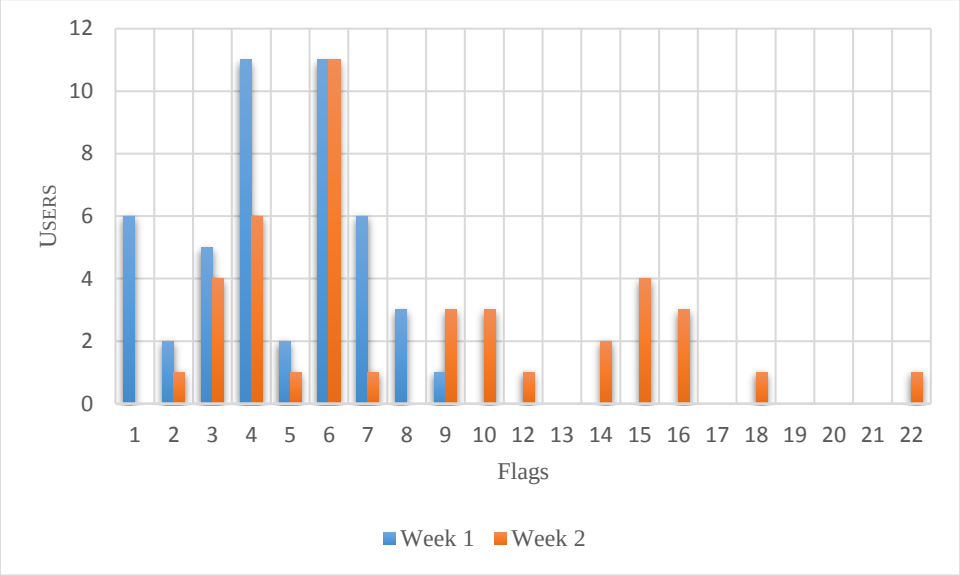

Figure 17. Flags found by users per each week group

Figure 18 shows how the user's performance is spread in each week. It displays a candlestick chart of flags found by users in each week. Median of the second group's flags is close to the first group but some students in the second week's group have been able to find more flags. Also, interestingly enough students in the second were able to find 40% more flags.

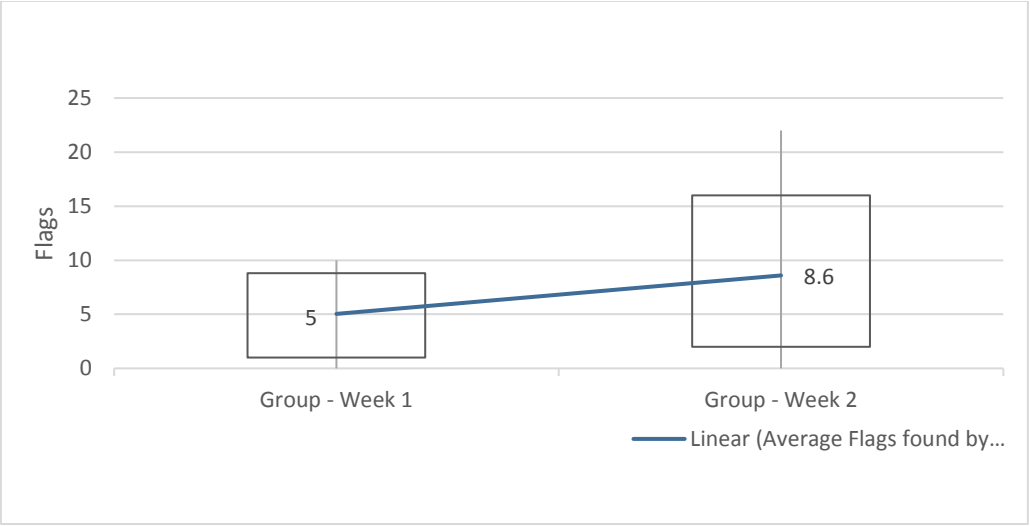

Figure 18. Candlestick chart – flags found by users per week

Another interesting observation of the OSINT lab result which is seen in Figure 19 is that in the first day of the exercise, students in the second week were able to find four times the amount of flags found in the first day of the first week's group. This sudden increase at the beginning of the second group's exercise is most probably because students shared the way of finding the flags with other students. It took more than three days for the first group to find as much as flags as the second group's first day. Another observation is that to receive one of the flags; students must have sent a real email, after eight days the OSINT lab was given to students, one student sent the correct email. Few hours after the first correct email, six different correct emails were also sent which is another indicator of how fast methods of receiving flags can be conveyed amongst the students.

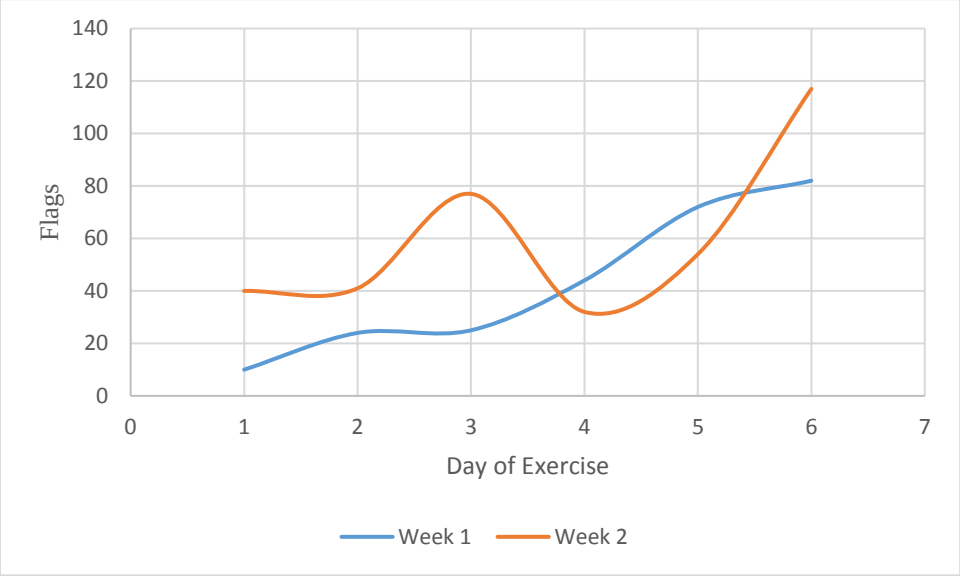

Figure 19. Correct flags found by students collectively per day

Even though knowledge sharing between the students is not an issue, it could be a limitation for this implementation because the thought process of reaching and finding a flag could get significantly influenced by knowledge sharing.

Questions in the quiz section meant to teach OSINT techniques, and some of those techniques were related to finding the flags in the main mission. One sample question in the quiz pool has been chosen to be discussed. It is difficult to make assumptions out of all the quiz questions for two reasons. First, not all of the questions were directly related to flags. Secondly, some flags relied on others to be found (i.e. they could not be found before another set of flags was found). For example, it will not be useful for students to find a flag based on a question if they haven't reached the location that the flag is located, and reaching to locations is reliant on finding the other flags. It was not the intention to

directly relate each question to an individual flag, but the intention was to teach the technique to find the flag, and it was up to students to strategize and use their creativity to use or not use the question in their investigation. SSL flag in the lab was not dependent on the other flags; Figure 20 shows how students performed in answering the question or finding the flags. Another thought-provoking observation was that 50% of the students who correctly answered the question found the correct flag in less than fifteen minutes. But direct questions like this may also decrease the level of creativity of the students, which seems is a trade-off between designing questions which exactly overlap with the main mission or designing questions that only teach the technique without any hint to the game's specific scenario.

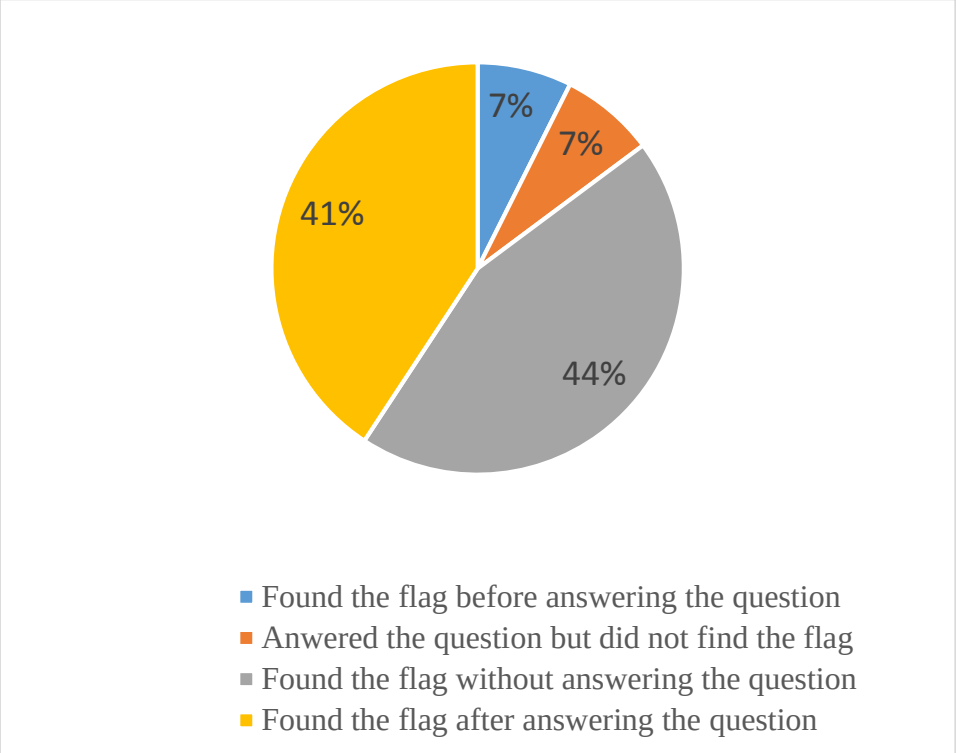

Figure 20. Comparison between the SSL question and the SSL flag

4.1.5 OSINT Octalysis Framework Motivation Assessment

Octalysis graph is used for OSINT as well similar to the RE lab (See section 4.1.2) to create a graph and score of this exercise lab. In Table 2, the chosen game design elements for each exercise lab was specified. The given metrics are discussed and evaluated by the lecturer of this course as an expert to measure the validity of the metrics. Nevertheless, similar to the RE lab, it should be noted that Octalysis scores are subjective by design and are therefore reflecting opinion, not an objective truth. The significance of the chosen game design elements and their core drivers is as follows:

1. CD#1 - Epic Meaning & Calling: Similar to the RE lab, one of the essential drivers was emphasizing on the student's will to be a hero and become a cyber security investigator and try to find a hacker. Techniques used in this lab were techniques used in real OSINT analysis and practicing some of those techniques helps students to become part of an elite group knowledgeable of these investigatory techniques. (Score: 7)
2. CD#2 - Development & Accomplishment: Most of the game design elements defined in this core driver has been used in the OSINT lab such as leaderboard, progress bar, badges, and points. This reliance on development core driver motivators shows the OSINT lab is focusing on being a goal based exercise. (Score: 7)
3. CD#3 - Ownership & Possession: OSINT lab design heavily relied on the status map. Students were able to choose pseudonyms as an avatar, status map and the flags could also be seen as virtual good. Overall a great deal of this core driver was implemented in the OSINT lab. (Score: 7)
4. CD#4 - Empowerment of Creativity & Feedback: Instant feedback was an essential game design element in this design also students do have complete control in the game students were able to freely investigate the incident in a free world style format and find the clues populated on the internet. (Score: 6)
5. CD#6 - Scarcity & Impatience: Similar to the RE lab, the only game design element used in this motivation driver was the count down which was the time limit of 6 days for students to finish the lab. (Score: 1)
6. CD#7 - Unpredictability & Curiosity: Other than the main mission there are also mini-quests in the quiz (Training) section of the game. Also, an EasterEgg has been implemented in the game which increases the randomness and curiosity of the exercise lab. (Score: 2)

The core drivers that no game design was chosen out of them received a score of zero. In Figure 21 [81] the graph generated by the online Octalysis tool is shown. Octalysis tool score generated for the OSINT lab is 188 which is a higher number than RE lab. This is mostly because of the usage of points, maps, badges, and leaderboard in the OSINT game which was not used in the RE lab.

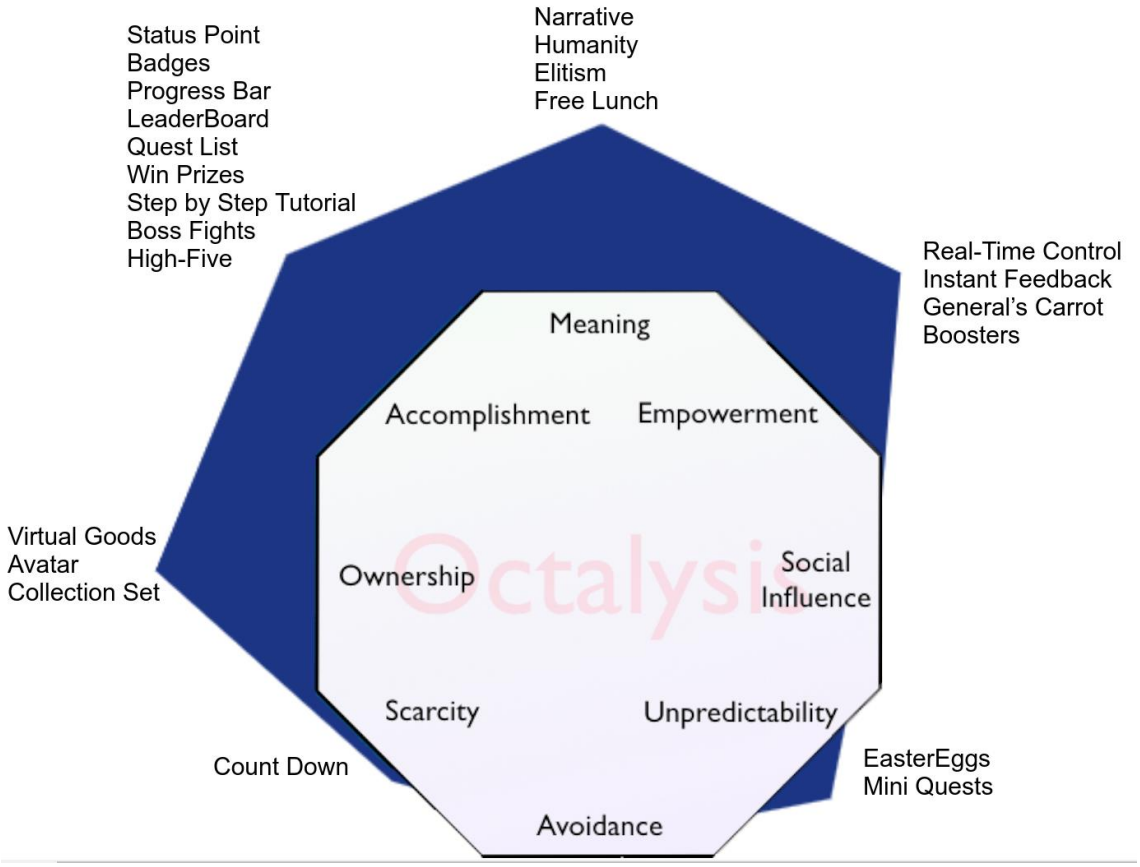

Figure 21. Octalysis Graph of OSINT lab. Source copied from [81]

Similar to the RE lab as shown in Figure 21 [81] the graph is towards up and left which is again an indication of successfully designing a gamification exercise which is oriented around the white hat attributes and activating extrinsic motivation and not the intrinsic motivations. The pro side of this design is that this gamification encouraged positive feelings of empowerment and successfully called for achieving a goal, the cons are that it fails to evoke long term motivation that could last after students have completed the training lab and failed to bring the social aspect of the gamification into it.

4.2 Questionnaire Results of the OSINT and Reverse Engineering

At the end of the course, students were given a questionnaire to assess how useful, interesting or hard each lab was. The Questionnaire consisted of three 6-point Likert scale questions for each lab. Also, students were given an open-ended question to write their comments about the labs.

89 students answered some or all of the questions and 4 questions were asked for 9 labs created in the course; these 4 questions which were in the Estonian language are as follows:

1. How hard was the lab? (6-Point scale from 0 to 5)
2. How interesting was the lab? (6-Point scale from 0 to 5)
3. How useful was the lab? (6-Point scale from 0 to 5)
4. The open-ended question in the form of comments for each lab.

Figure 22 shows an overview of the labs and how students perceived its usefulness, difficulty or how interesting it was. In this figure, the average of the points given by the students is shown.

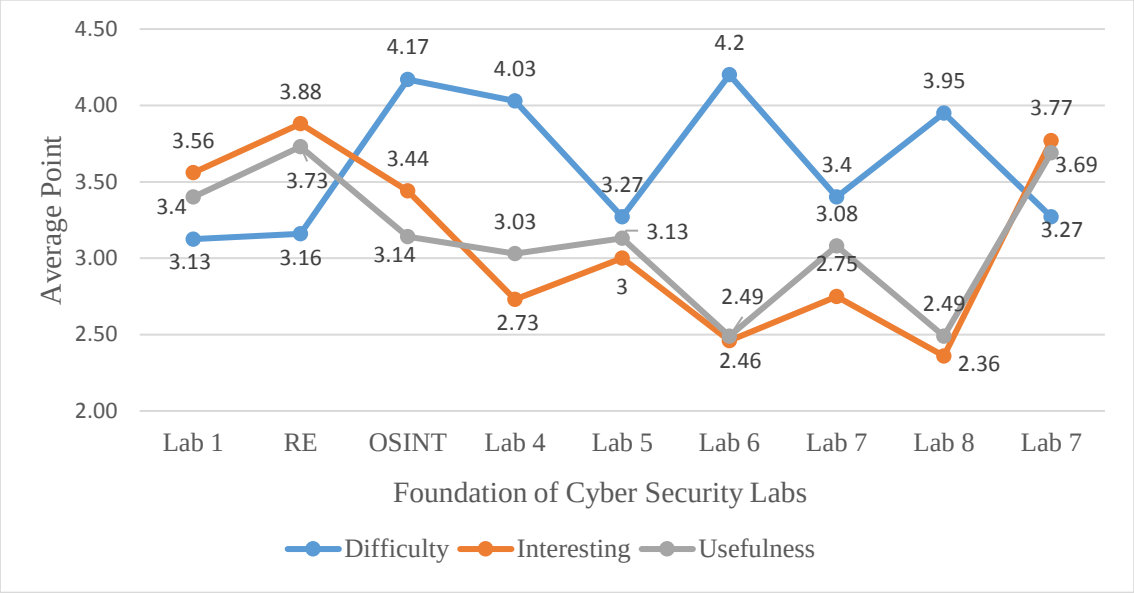

Figure 22. Comparison of the Foundation of the Cyber Security Course's lab

As seen in Figure 22, in contrast to other labs in the course, RE lab was perceived as the most useful and interesting lab and the second easiest lab, but OSINT did not perform well, and it was seen as the second hardest lab and fourth most interesting and fourth most useful lab amongst the 9 labs.

4.2.1 Reverse Engineering Questionnaire

Out of 89 students, up to 4.5% did not answer the questions for the Reverse Engineering lab. Table 4 shows an overview of the Reverse Engineering lab's questionnaire with the translated questions in English. Average point is from 0 to 5 which is given by students for each question.

Table 4. Reverse Engineering lab questionnaire overview

Translated Question	Number of students who answered the questions	N/A(did not answer)	Average Point
How hard was it?	86	3 (3.8%)	3.16
How interesting was it?	86	3 (3.8%)	3.88
How useful was it?	85	4 (4.5%)	3.73

Figure 23 shows in detail the percentages of total points given by students for the RE lab in regards to its difficulty, usefulness or how interesting it was.

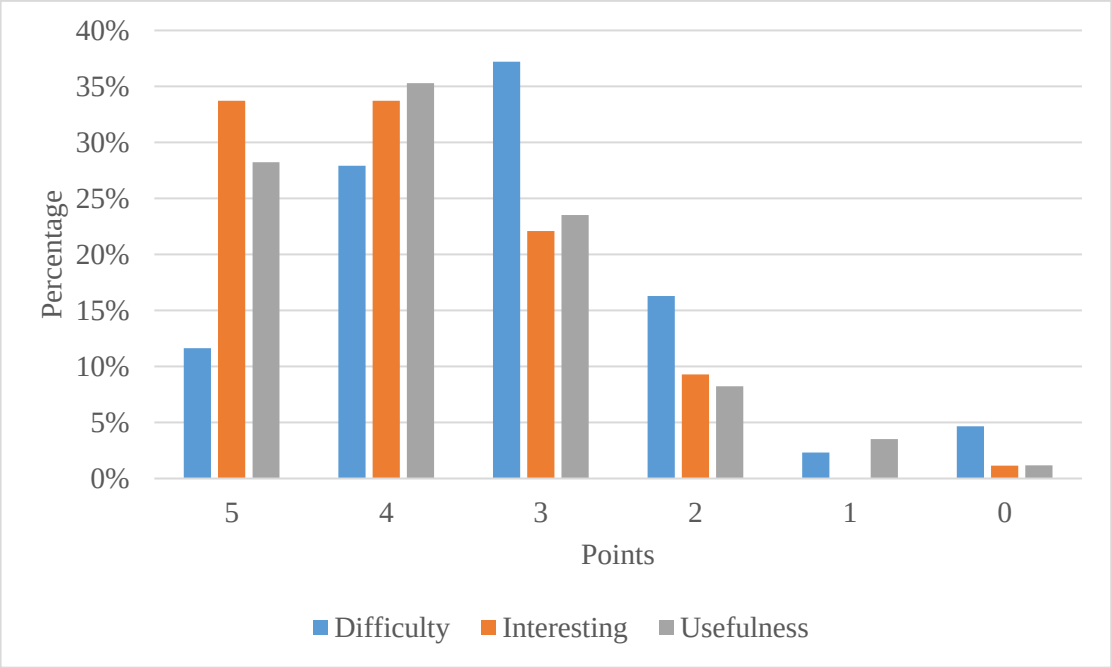

Figure 23. Questionnaire results of the RE lab

RE lab was designed in a step-by-step instruction based style and comments of the students in the open-ended section of the questionnaire indicated that students thought this lab in their opinion was an easy and yet useful and interesting lab.

4.2.2 OSINT Questionnaire

Out of 89 students, up to 4.5% did not answer the questions for the OSINT lab. Table 5 shows an overview of the OSINT lab's questionnaire with the translated questions in English. Average point is from 0 to 5 which is given by students for each question.

Table 5. OSINT lab questionnaire overview

Translated Question	Number of students who answered the questions	N/A(did not answer)	Average Point
How hard was it?	87	2 (2.3%)	4.17
How interesting was it?	86	3 (3.8%)	3.33
How useful was it?	85	4 (4.5%)	3.14

Figure 24 shows in detail the percentages of total points given by students for the OSINT lab in regards to its difficulty, usefulness or how interesting it was.

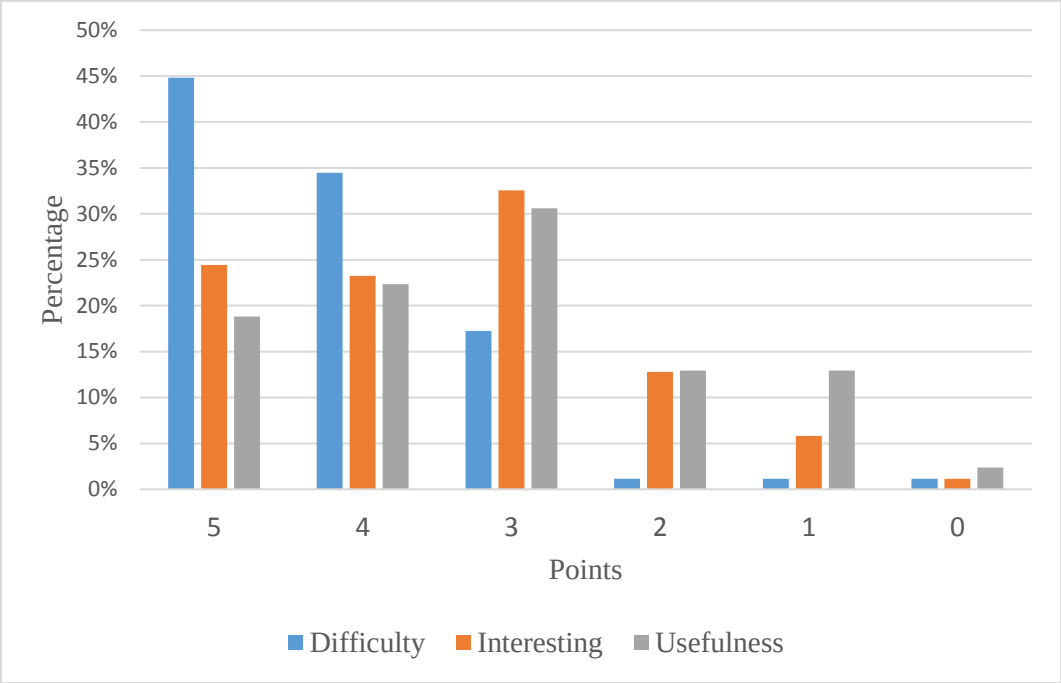

Figure 24. Questionnaire results of the OSINT lab

Comments of the students in the open-ended section of the questionnaire showed that students thought that OSINT lab in their opinion was frustrating and very difficult, but yet still an interesting lab idea. Some students also mentioned that they did not believe this lab was suitable for cyber security training. OSINT lab was designed for bachelor students, and this perceived difficulty and frustration in solving the lab was not an expected result. A pre introductory session before giving the OSINT lab to students could have helped in addressing those comments.

Since questionnaire results showed the RE lab (see Figure 22Figure 23) to be the most useful and interesting lab, it is possible that low points given to the OSINT lab's

usefulness and engagement (interesting) are shaped by the difficulty of the OSINT lab in finding the flags. In section 4.1.4 the performance of each week's group in the OSINT lab was compared to each other. Out of 89 students who participated in the questionnaire, three of them did not participate in the OSINT lab; Figure 25 shows the opinion of students in each OSINT group. First OSINT group participated in the lab one week before the second group.

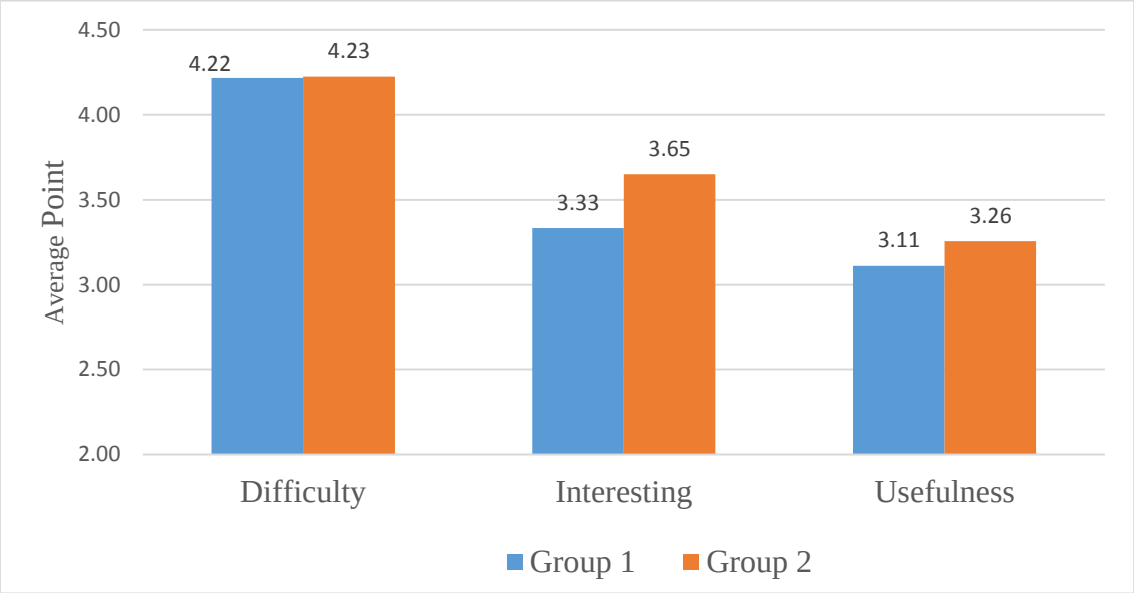

Figure 25. Average points of OSINT questionnaire per each week group

Students in the second group (second week's group) were able to find more flags and subsequently able to work with different OSINT techniques and get a better grade; one observation was that students communicated with each other and therefore students in the second week performed better than the first group (see section 4.1.4).

Students in the second group found the OSINT lab to be with the same difficulty, but it was curious that the average point for 'usefulness' or 'interesting' question, slightly increased in the second group's opinion. The first group did not benefit from the experience and share of knowledge that was available for the second group. It is possible that lack of balance[83] between the skills required to solve the lab and the high level of challenge and difficulty of the lab especially amongst the first group, contributed to the increased frustration of the students and lack of an ideal flow and engagement in the lab.

5 Lessons Learned

The OSINT lab proved to be too difficult for the chosen target group, and possibly this lab is more suitable for the master students and not bachelor students. Feedback revealed that the training lab instructions failed to give a proper overview of the lab to the students, therefore providing an introductory session in the class before kick-off would have helped to achieve this. Even though some students performed well in the OSINT lab, the feedback received from the students showed that it is hard to create an engaging lab that interests everyone.

Comparing the results of each week's performance revealed that students communicate with each other, regarding the lab's answer. This demonstrates the importance of implementing anti-cheating mechanisms and having an individual flag per students. Students had different approaches in solving the tasks, some students tried to guess or brute force the answers, and some tried to follow the step by step instruction; people are different and have different likes and interests, creating various type of tasks within an exercise lab creates a more engaging environment for a wide range of audience.

The target audience should have been involved in the design phase and playtesting to create a more appropriate exercise for the target group. This involvement possibly would have shown the high level of complexity of the OSINT lab before its deployment. The complications of the gamification design revealed that extensive knowledge of the gamification platform and its infrastructure is a necessity in creating a well-designed gamified training.

Predicting the possible wrong answers of the students, helped to create special hints for specific wrong answers, which resulted in directing the students to the correct answer. Lack of sufficient logs failed to address this method more in the study, but live monitoring of the lab showed this to be a useful technique in educating students on why their approach was wrong and what they have to do to find the right solution.

6 Future Work

The user interface of the currently designed labs are aesthetically mediocre, creating a more user-friendly environment in the future will build a more engaging exercise lab for the students. Similarly, incorporating user experience design techniques in future works is highly suggested. This would require the developers to have extensive knowledge in the user experience design field to create a more efficient, elegant and interactive interface.

A/B testing of the ‘Hint’ and ‘Free Hint’ showed the effects of negative reinforcement on student's behavior, but the effects of positive reinforcement and other user experience research techniques could be measured in future work. Also, the current gamification design did not incorporate the core drivers that related to intrinsic motivations such as social influence or evolved user interface; future works can study these core drivers more in-depth and redesign the exercise to have a long-lasting effect on the student’s learning process.

Live monitoring of the OSINT lab logs showed that some students were using useful uncovered techniques in the OSINT lab, but there was no flag within the locations that students are looking into. Exercise lab did not reward those correct behaviors. For future work creating a comprehensive lab which covers a complete list of relevant techniques and giving rewards for using them is suggested.

Using specific hints for particular wrong answers showed to be a promising technique in guiding the students to the right solution. Analyzing the wrong answers of students and providing more comprehensive special hints instead of the typical ‘wrong answer’ messages is a suggested research for future work.

7 Conclusion

In this work, gamified OSINT and Reverse Engineering cybersecurity exercises were designed with the aim to be relevant, engaging and difficult to cheat. The training labs were implemented for the TalTech students of the ‘Foundations of Cyber Security’ course. The design process of the OSINT lab showed to be a complex and challenging and at the same time a promising method of teaching OSINT to the students. The requirements of the OSINT and Reverse engineering topic and game design steps were analyzed, which resulted in an applicable and relatable exercise lab.

The implemented training labs included individual flags for each student to avoid and detect the cheating attempts by the participants. Other researchers and developers can replicate the process of the lab's game design. This process could be used by others to create similar training labs for the students. A novel way of inserting flags to any web site's source code was described. That enables to add flags to web sites that are not controlled by the exercise designers and therefore helps to create realistic, yet individual, labs using public web sites. This flag inserting system was used successfully in OSINT lab.

Gamification design in both of the labs showed positive results in increasing the extrinsic motivation of students, but not the intrinsic motivations. The implemented training labs showed to be modular and scalable, which makes it easy to add more content or flags to the developed training labs or support more users by just adding more hardware capability to the virtualized platform.

References

[1] L. J. Hoffman, T. Rosenberg, R. Dodge, and D. Ragsdale, “Exploring a national cybersecurity exercise for universities,” *IEEE Secur. Priv.*, vol. 3, no. 5, pp. 27–33, Sep. 2005.

[2] M. Ernits, J. Tammekänd, and O. Maennel, “i-tee: A fully automated Cyber Defense Competition for Students,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 113–114, Aug. 2015.

[3] S. Deterding, M. Sicart, L. Nacke, K. O’Hara, and D. Dixon, “Gamification. using game-design elements in non-gaming contexts,” in *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA ’11*, 2011, p. 2425.

[4] M. Sailer, J. U. Hense, S. K. Mayr, and H. Mandl, “How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction,” *Comput. Human Behav.*, vol. 69, pp. 371–380, Apr. 2017.

[5] J. Hamari, J. Koivisto, and H. Sarsa, “Does gamification work? - A literature review of empirical studies on gamification,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2014, pp. 3025–3034.

[6] Gartner, “Gartner Says By 2015, More Than 50 Percent of Organizations That Manage Innovation Processes Will Gamify Those Processes,” *Www.Gartner.Com*, 2011. [Online]. Available: https://www.gartner.com/newsroom/id/1629214. [Accessed: 25-Nov-2018].

[7] G. Surendeleg, V. Murwa, H.-K. Yun, and Y. S. Kim, “The role of gamification in education–a literature review,” *Contemp. Eng. Sci.*, vol. 7, pp. 1609–1616, 2014.

[8] K. M. Kapp, *The gamification of learning and instruction : game-based methods and strategies for training and education*. .

[9] Y.-K. Chou, *Actionable Gamification: Beyond Points, Badges, and Leaderboards*, vol. 1542. [California]: Octalysis Media, 2016.

[10] S. Nicholson, “Strategies for meaningful gamification: Concepts behind transformative play and participatory museums.,” 2012.

[11] H. D. Farhad, A. R. Ghatari, and A. Hasiri, *Employees Morale in Public Sector: Is Organizational Trust an Important Factor?*, vol. 46, no. 3. EuroJournals, 2010.

[12] D. C. Martin and K. M. Bartol, “Performance appraisal: Maintaining system effectiveness,” *Public Pers. Manage.*, vol. 27, no. 2, pp. 223–230, Jun. 1998.

[13] S. N. Shuaib, “Impact of Leadership Style on Employees’ Motivation: A Case Study of Access Bank Nigeria,” 2016.

[14] B. J. Osabiya, “The effect of employees motivation on organizational performance,” *J. Public Adm. Policy Res.*, vol. 7, no. 4, pp. 62–75, 2016.

[15] R. M. Ryan and E. L. Deci, “Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions,” *Contemp. Educ. Psychol.*, vol. 25, no. 1, pp. 54–67, Jan. 2000.

[16] J. A. Middleton and P. A. Spanias, “Motivation for Achievement in Mathematics: Findings, Generalizations, and Criticisms of the Research,” *J. Res. Math. Educ.*,

vol. 30, no. 1, p. 65, Jan. 2006.
[17] S. Reiss, “Intrinsic and Extrinsic Motivation,” *Teach. Psychol.*, vol. 39, no. 2, pp. 152–156, Apr. 2012.
[18] M. Battaglini, R. Bénabou, and J. Tirole, “Self-control in peer groups,” *J. Econ. Theory*, vol. 123, no. 2, pp. 105–134, 2005.
[19] D. J. Miller and D. P. Robertson, “Using a games console in the primary classroom: Effects of ‘Brain Training’ programme on computation and self-esteem,” *Br. J. Educ. Technol.*, vol. 41, no. 2, pp. 242–255, Mar. 2010.
[20] F. F. Nah and F. Hall, “Flow in gaming : literature synthesis and framework development Brenda Eschenbrenner Qing Zeng and Venkata Rajasekhar Telaprolu Sepandar Sepehr,” 2014.
[21] J. Hamari and J. Koivisto, “Measuring flow in gamification: Dispositional Flow Scale-2,” *Comput. Human Behav.*, vol. 40, pp. 133–143, Nov. 2014.
[22] C. (Gartner I. . Pettey and R. (Gartner I. . Van der Meulen, “Gartner Says by 2014, 80 Percent of Current Gamified Applications Will Fail to Meet Business Objectives Primarily Due to Poor Design,” 2014. [Online]. Available: https://www.gartner.com/newsroom/id/2251015. [Accessed: 24-Nov-2018].
[23] M. Foucault, X. Blanc, M.-A. Storey, J.-R. Falleri, and C. Teyton, “Gamification: a Game Changer for Managing Technical Debt? A Design Study,” Feb. 2018.
[24] S. Alexander, *The Gameful World: approaches, issues, applications*, vol. 59, no. 2. 2015.
[25] J. Ricci, F. Breitinger, and I. Baggili, “Survey results on adults and cybersecurity education,” *Educ. Inf. Technol.*, vol. 24, no. 1, pp. 231–249, Jan. 2019.
[26] I. B. Alvarez, N. S. A. Silva, and L. S. Correia, “Cyber education: towards a pedagogical and heuristic learning,” *ACM SIGCAS Comput. Soc.*, vol. 45, no. 3, pp. 185–192, Jan. 2016.
[27] FireEye, “Spear Phishing Attacks — Why They are Successful and How to Stop Them Combating the Attack of Choice for Cybercriminals,” 2013.
[28] O. Maennel, “Gaps in European Cyber Education and Professional Training,” 2018.
[29] A. Dabrowski, M. Kammerstetter, E. Thamm, E. Weippl, and W. Kastner, “Leveraging Competitive Gamification for Sustainable Fun and Profit in Security Education,” 2015.
[30] R. Lukman and M. Krajnc, “Exploring Non-traditional Learning Methods in Virtual and Real-world Environments,” 2012.
[31] A. Y. Kolb and D. A. Kolb, “Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education,” *Source Acad. Manag. Learn. Educ.*, vol. 4, no. 2, pp. 193–212, 2005.
[32] M. B. Richards and S. W. Marshall, “Experiential Learning Theory in Digital Marketing Communication: Application and Outcomes of the Applied Marketing & Media Education Norm (AMEN),” *J. Mark. Dev. Compet.*, vol. 13, no. 1, Mar. 2019.
[33] L. Valenzuela, O. M. Jerez, B. A. Hasbún, V. Pizarro, G. Valenzuela, and C. A. Orsini, “Closing the gap between business undergraduate education and the organisational environment: A Chilean case study applying experiential learning theory,” *Innov. Educ. Teach. Int.*, vol. 55, no. 5, pp. 566–575, Mar. 2018.
[34] X. J. H. J.-C. C.-C. Zhai, “An Experiential Learning Perspective on Students’ Satisfaction Model in a Flipped Classroom Context.,” *Educ. Technol. Soc.*, vol. 20, no. 1, pp. 198–210, 2017.
[35] V. Rao, “E-learning: As a medium of new education,” *Sansmaran Res. J.*, vol. 6,

no. 2, pp. 3–7, 2016.
[36] M. R. Findley, “The Relationship between Student Learning Styles and Motivation during Educational Video Game Play,” *Int. J. Online Pedagog. Course Des.*, vol. 1, no. 3, pp. 63–73, Jul. 2011.
[37] S. Tang and M. Hanneghan, “A model-driven framework to support development of serious games for game-based learning,” in *Proceedings - 3rd International Conference on Developments in eSystems Engineering, DeSE 2010*, 2010, pp. 95–100.
[38] G. Jin, M. Tu, T.-H. Kim, J. Heffron, and J. White, “Game based Cybersecurity Training for High School Students,” in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE ’18*, 2018, pp. 68–73.
[39] S. Fouché and A. H. Mangle, “Code hunt as platform for gamification of cybersecurity training,” in *Proceedings of the 1st International Workshop on Code Hunt Workshop on Educational Software Engineering - CHESE 2015*, 2015, pp. 9–11.
[40] H. Hosseini and L. Perweiler, “Are You Game?: Assessing Stu-dents’ Perception of Learning, Instructors’ Perspective, and Learning At-titude,” 2019.
[41] R. J. Baxter, K. Holderness, and D. A. Wood, “The Effects of Gamification on IT Compliance Training: Evidence from the Field and Lab,” *SSRN Electron. J.*, Oct. 2014.
[42] B. Morschheuser, J. Hamari, K. Werder, and J. Abe, “How to Gamify? A Method For Designing Gamification,” *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, Jan. 2017.
[43] M. Bazzell, *Open Source Intelligence Techniques: Resources for Searching and Analyzing Online Information*. 2018.
[44] E. J. Chikofsky and J. H. Cross, “Reverse Engineering and Design Recovery: A Taxonomy,” *IEEE Softw.*, vol. 7, no. 1, pp. 13–17, Jan. 1990.
[45] D. Quick and K. K. R. Choo, “Digital forensic intelligence: Data subsets and Open Source Intelligence (DFINT+OSINT): A timely and cohesive mix,” *Futur. Gener. Comput. Syst.*, vol. 78, pp. 558–567, Jan. 2018.
[46] “Toddington_International. Online Investigator’s Checklist. Toddington International Inc.,” 2016. [Online]. Available: https://1x7meb3bmahktmrx39tuiync-wpengine.netdna-ssl.com/wp-content/uploads/TII_Online-Investigators-Checklist_v2-1.pdf.
[47] Q. Brown, B. Tate, G. Irwin, R. Brewer, J. Nias, and L. Anthony, “Using gamification to motivate children to complete empirical studies in lab environments,” in *Proceedings of the 12th International Conference on Interaction Design and Children - IDC ’13*, 2013, pp. 388–391.
[48] D. Atkins and R. Austein, “RFC3833: Threat Analysis of the Domain Name System (DNS),” *IETF Req. Comments*, pp. 1–17, 2004.
[49] E. J. Byrne, “Software reverse engineering: A case study,” 1991.
[50] Open Source Initiative, “The MIT License | Open Source Initiative,” 2018. [Online]. Available: https://opensource.org/licenses/MIT. [Accessed: 21-Apr-2019].
[51] F. F. H. Nah, Q. Zeng, V. R. Telaprolu, A. P. Ayyappa, and B. Eschenbrenner, “Gamification of education: A review of literature,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8527 LNCS, Springer, Cham, 2014, pp. 401–409.
[52] C. Santos, S. Almeida, L. Pedro, M. Aresta, and T. Koch-Grunberg, “Students’ perspectives on badges in educational social media platforms: the case of SAPO

campus tutorial badges," in *Proceedings - 2013 IEEE 13th International Conference on Advanced Learning Technologies, ICALT 2013*, 2013, pp. 351–353.
[53] G. Goehle, "Gamification and Web-based Homework," *Primus*, vol. 23, no. 3, pp. 234–246, Mar. 2013.
[54] S. O'Donovan, J. Gain, and P. Marais, "A case study in the gamification of a university-level games development course," in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference on - SAICSIT '13*, 2013, p. 242.
[55] R. Raymer, "Gamification: Using Game Mechanics to Enhance eLearning," *eLearn*, vol. 2011, no. 9, p. 3, Sep. 2012.
[56] K. Berkling and C. Thomas, "Gamification of a software engineering course and a detailed analysis of the factors that lead to it's failure," in *2013 International Conference on Interactive Collaborative Learning, ICL 2013*, 2013, pp. 525–530.
[57] K. Seaborn and D. I. Fels, "Gamification in theory and action: A survey," *Int. J. Hum. Comput. Stud.*, vol. 74, pp. 14–31, Feb. 2015.
[58] Y. Chou, "Octalysis: Complete Gamification Framework - Yu-kai Chou," 2013. [Online]. Available: https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/%0Ahttps://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/#.WzF0A1UzZhE. [Accessed: 24-Mar-2019].
[59] R. D. da Silva Brito, L. H. Contreras Pinochet, E. Luiz Lopes, and M. A. de Oliveira, "Development of a gamification characteristics measurement scale for mobile application users.," *Desenvolv. uma escala mensuração Caracter. gamificação para usuários Apl. em Dispos. móveis.*, vol. 13, no. 1, pp. 1–16, Jan. 2018.
[60] G. Zichermann and C. Cunningham, *Gamification By Design - Implementing Game Mechanics in Web and Mobile Apps*. [Verlag nicht ermittelbar], 2011.
[61] J. Hong, "The state of phishing attacks," *Commun. ACM*, vol. 55, no. 1, p. 74, Jan. 2011.
[62] N. Asanka, G. Arachchilage, and M. A. Hameed, "Integrating "self-efficacy" into a gamified approach to thwart phishing attacks."
[63] "The phishing email that hacked the account of John Podesta - CBS News." [Online]. Available: https://www.cbsnews.com/news/the-phishing-email-that-hacked-the-account-of-john-podesta/. [Accessed: 31-Mar-2019].
[64] N. Gupta, A. Aggarwal, and P. Kumaraguru, "Bit.ly/malicious: Deep dive into short URL based e-crime detection," in *eCrime Researchers Summit, eCrime*, 2014, vol. 2014-Janua, pp. 14–24.
[65] "What are Certificate Authorities & Trust Hierarchies?" [Online]. Available: https://www.globalsign.com/en/ssl-information-center/what-are-certification-authorities-trust-hierarchies/.
[66] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS certificate ecosystem," pp. 291–304, 2013.
[67] L. S. Huang, A. Rice, E. Ellingsen, and C. Jackson, "Analyzing forged SSL certificates in the wild," in *Proceedings - IEEE Symposium on Security and Privacy*, 2014, pp. 83–97.
[68] D. Sarkar, *Nginx 1 Web Server Implementation Cookbook*. 2011.
[69] SANS, "InfoSec Reading Room: A Reverse Proxy Is A Proxy By Any Other Name," 2002.
[70] N. Yoshida, "Dynamic CDN against flash crowds," 2008.

[71] “Substitutions | NGINX.” [Online]. Available: https://www.nginx.com/resources/wiki/modules/substitutions/. [Accessed: 22-Mar-2019].
[72] “mod_substitute - Apache HTTP Server Version 2.4.” [Online]. Available: https://httpd.apache.org/docs/2.4/en/mod/mod_substitute.html. [Accessed: 22-Mar-2019].
[73] S. Son and V. Shmatikov, “The hitchhiker’s guide to DNS cache poisoning,” in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 50 LNICST, Springer, Berlin, Heidelberg, 2010, pp. 466–483.
[74] J. Kaur and T. Elsner, “Behind the Scenes: Accepting Untrusted Certificates in a Web Browser.”
[75] M.-L. Potet *et al.*, “BINSEC/SE: A Dynamic Symbolic Execution Toolkit for Binary-Level Analysis,” 2016.
[76] K. Werbach and D. Hunter, *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton, 2012.
[77] HttpWatch, “6 Things You Should Know About Fragment URLs | HttpWatch Blog,” 2011. [Online]. Available: http://blog.httpwatch.com/2011/03/01/6-things-you-should-know-about-fragment-urls/. [Accessed: 31-Mar-2019].
[78] “Law Enforcement Recommendations Regarding Amendments to the Registrar Accreditation Agreement 1.”
[79] “Intellectual Property Protection and Courts Amendment Act of 2004. (eBook, 2005) [WorldCat.org].” [Online]. Available: https://www.worldcat.org/title/intellectual-property-protection-and-courts-amendment-act-of-2004/oclc/849616287&referer=brief_results. [Accessed: 04-Apr-2019].
[80] T. a Proebsting and S. a Watterson, “Krakatoa: Decompilation in java (does bytecode reveal source,” 1997.
[81] Yukaichou, “Octalysis / Gamification Building Developing Online Tool - by Yukai Chou.” [Online]. Available: http://www.yukaichou.com/octalysis-tool/. [Accessed: 09-Apr-2019].
[82] E. Andersen, Y.-E. Liu, R. Snider, R. Szeto, and Z. Popović, *Placing a value on aesthetics in online casual games*. 2011.
[83] J. Nakamura and M. Csikszentmihalyi, “The concept of flow,” in *Flow and the Foundations of Positive Psychology: The Collected Works of Mihaly Csikszentmihalyi*, Dordrecht: Springer Netherlands, 2014, pp. 239–263.

Appendix 1 – Example Script for Redirecting Domain Traffic

The code below shows how to modify the content of a domain example.com with 1.1.1.1 as an IP address. The script generates a fake certificate and redirects the users with DNS Spoofing or iptables depending on the need. This script allows embedding a flag inside of it without destabilizing the browsing experience of training lab users. This bash script also relies on custom configuration and preconfigured tools, but the script below shows in general how the game design for inserting flags works.

```
#!/bin/bash

domain="example.com"
domainIP="1.1.1.1"

#Generate a CA, only issue it once
openssl req -new -config customopenssl.cnf  -newkey rsa:4096 -sha256 \
       -keyout OSINT.key -x509 -days 7300 -text -out OSINT.crt

#Create the SSL key
openssl req -config openssl.cnf -new -newkey rsa:4096 -sha256 \
       -keyout "${domain}.key" -subj "/CN=${domain}" \
       -text -out "${domain}.csr"

#Sign the CSR
openssl ca -config customopenssl.cnf -in "${server}.csr" –out \
       "${server}.crt" –batch \
       <(printf "\n[SAN]\nsubjectAltName=DNS:${domain},DNS:*.${domain}"))

#DNS Spoofing - Manual DNS records to redirect the users to our reverse proxy
#Only if users do not need to work with the real IP address of the domain:
echo -e 'port=53\ndomain-needed\nbogus-priv\nstrict-order \
       \nlisten-address=192.168.8.254\nno-hosts \
       \naddn-hosts=/etc/dnsmasq.hosts' >> /etc/dnsmasq.conf
echo -e '192.168.8.254    ${domain}"' >> /etc/dnsmasq.hosts

#Rediect port packets to 80 and 443 to our revese proxuy
#Only If users need to know to see the real IP address:
iptables -t nat -A PREROUTING -p tcp -d ${domainIP}  --dport 80 –j \
       DNAT --to-destination 192.168.8.254:80
iptables -t nat -A PREROUTING -p tcp -d ${domainIP}  --dport 443 –j \
       DNAT --to-destination 192.168.8.254:443
```

Appendix 2 – Example Script for Flag Generation

The script below shows how the flag generation process works in detail for the domain example.com. Some parts of the script have been redacted to limit the possibility of brute forcing the flags. New flags are generated every time a user starts the lab, due to randomization and a low number of users, each user gets their own set of flags.

```
#!/bin/bash

domain="example.com"

#Flag number - Some domains might have multiple flags within them
flagNumber="1"

#Generate a random string only for this lab instance
#This script should run every time a user starts the lab
PHP=`which php`
$PHP generateSecret.php

#Get the randomly generated secret to create the hash
flag_secret=$(cat secret.txt)

#Generate the flag name based on domain and flag number
flag_name="flag_${domain}_task_${flagNumber}"

#Get the username of the lab participant
username=$(cat username.txt)

#Create the raw flag text that needs to be hashed
flag_raw="$username$flag_secret$flag_name"

#Create final flag from the username, flag number, domain, and the secret
hash
#First 16 character of the generated hash will become the final flag
flag=$(echo -n $flag_raw | hashGenerator | awk '{ print $1 }' | head -c 16)

#put the generated flag in the site reverse proxy configuration file to be
inserted in the user traffic
sed -i "s/$flag_name/$flag/g" ${domain}.conf
```

Appendix 3 – Example Reverse Proxy Apache Configuration

The configuration file blow shows the configuration for reverse proxying the port 443 traffic of the domain example.com with Apache. This configuration allows the Reverse Proxy to use the fake SSL certificates signed by the lab's CA and pass the traffic to the real website. Then Apache substitutes the traffic based on its conditions and replaces the specified HTML response with the unique flag generated.

```
<VirtualHost *:443>
	ServerName example.com
	SSLProxyEngine on
	SSLCertificateFile example.com.crt
	SSLCertificateKeyFile example.com.key
	SSLCACertificateFile OSINT.crt
	SSLOptions +ExportCertData
	ProxyRequests Off
	<Proxy *>
		Order allow,deny
		Allow from All
	<Proxy>
	RequestHeader unset Accept-Encoding
	"s|<div>Please Replace me |Nice Job! E{12345678901234} |i"
	FilterDeclare NEWPATHS
	FilterProvider NEWPATHS SUBSTITUTE "%{Content_Type} =~ m|^text/html|"
	FilterChain NEWPATHS
	ProxyPass / https://example.com/
	ProxyPassReverse / https://example.com/
	ErrorLog /var/log/apache2/example-ssl.local-error.log
	CustomLog /var/log/apache2/example-ssl.local-access.log combined
</VirtualHost>
```