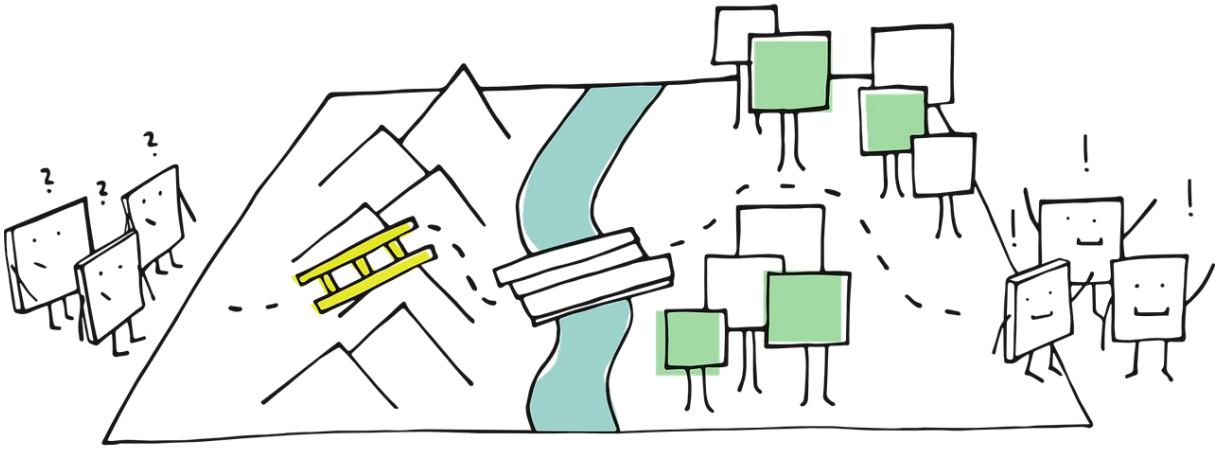


IO3: Training path of the game on digital security for secondary school pupils

Co-funded by the
Erasmus+ Programme
of the European Union

Spis treści

Co-funded by the
Erasmus+ Programme
of the European Union

Training kit

Students Kit has materials:

- The path for students how to be trained about the game
- Gamer types of explanation
- The scenario
- How to play the game (guide, hints, and badge/ranking system)

Path

Training materials for students should be structured: topic, objective/learning outcome, length, format, skill level/skills needed, exercise descriptions and guide, materials needed, examples/supporting materials for the teacher) and how to solve the challenges regarding CS topics and game mechanics.

Path for the students:

Nr	Task	Time	Teachers	Students
1	CS theoretical materials	1-3 hours	Explain the key CS theoretical contents, explain learning outcomes	Read, watch videos, do tests, discuss, etc
2	Pre-game	1-2 hour	Explain the game mechanics, and gamer types.	Read, video, evaluate their gamer types, divide roles
3	On Game	1-5 hours	Facilitate game, deal with raised issues	Play the game in pairs/group
4	After game	1 hour	Recap the learning and what happened in the game. Discuss with students the learning that took place.	Give feedback, do self-evaluation

Content of the training module:

Module 1 - cybersecurity (CS) theoretical materials (1-3 hours)

Objectives - This module introduces cybersecurity concepts and skills needed to learn cybersecurity in the 7.-12. grade level as well to play the game.

Skills and knowledge acquired by the participants at the end of the training - be able to:

- explain key concepts of cybersecurity (see Keywords for further content development)

Co-funded by the Erasmus+ Programme of the European Union

Training methods – lecture:

Module 2 - Pre-game competencies (1-2 hours)

Objectives - This module introduces cybersecurity commonly known as gaming type “Capture the flag” (CTF), different goals and needs when one plays games (gamer types). This module introduces the Cyber4Schools project and the developed game mechanics (how it's played as a player (blue and red gamer). Learn how to use supporting materials (topics in the game), hints, etc.

Skills – be able to:

- Explain different CTF types
- Give examples of websites where to learn and get experiences of CTFs
- Understand different gamer types and their goals while playing the game
- Evaluate their gamer type
- Knows how to play the game, get help, use the game mechanics

Contents - materials about gamer types (slides), testing tool about their gamer type (on the website), CTF type's introduction sheet, game environment, introductory videos for students (how the game works), guide of the game (game manual for students)

Training methods – lecture, practice hands-on on the game environment (hands-on), investigate materials, ask questions

Module 3 - Game training (experience) (1-5 hours)

Objectives – This module gives hands-on experience in the game as a red and blue player.

Skills - be able to:

- Play the game as a player (red)
- Play the game as a player (blue)

Contents - game environment, guide of the game

Training methods - practice hands-on in the game environment

Module 4 - After game/Wrap-up the learning (1 hour)

Objectives – This module is helping to wrap up the learning for the students, share ideas and emotions, and ask questions about how one could solve the issues in the game differently and their need for future learning possibilities.

Skills - be able to:

- Explain how the game is linked to real life and skills that were discussed in module 1.
- Wrap-up the learning for the students (self-evaluation sheet)

Contents – materials for wrap-up (slides), self-evaluation sheet

Training methods – discussion, experience sharing, lecture

The student's kit should focus on the game (how to play it, objectives and learning outcomes, deliverable of measured skills/competencies, and explanation of the badge/ranking system and gamer types). For students,

Co-funded by the Erasmus+ Programme of the European Union

there should be provided some theoretical materials about CS that help them in the game, and gamer types. Also, a self-evaluation form after the game should be taken.

Needed supporting materials to be developed:

- Theoretical materials about CS for students (and teacher's commentary with links to the curriculum). Materials for students may include videos, slides, and self-evaluation tests. Teacher material is reading material.
- Guide for playing the game, including gamer types of theory explanation. Materials for students may include video, slides, or a guidebook, a tool to analyze your gamer type. Teacher material is reading material.
- Self-evaluation form after the game

Gamer types

The game should explain and take into account Marczewski's User Types Analysis: Player, Socializer, Philanthropist, Disruptor, Free Spirit of Achiever. Before the actual game, every participant will evaluate their type and goal in the game using a similar tool, to understand that according to the goal and type the game experiences differ: https://gamified.uk/UserTypeAnalysis/#.YH2fMuhKjSF

Game types analysis possibilities will be introduced in both teacher and student training and are used as a self-evaluation tool to understand your play on the game. Exercise is optional.

Keywords for further content development

Event	Keywords
1. Social Engineering	Emails, Safe Email attributes, Email Server, Phishing, Malicious Links, HTML, HTML tags, Link Redirection, Plain-text Emails, HTML-based Emails
2. Malware	Malware, Malware Behavior, Reverse Engineering, File Backup, Corrupted Files, Encryption, Obfuscation, Antivirus
3. Network Security	Types of Networks, VPN, In/Secure Access Points, Web Servers, Network Traffic, Ip-Tables, MITM (Man-in-the-middle) Attacks, Page Index, Websites
4. Credentials and Authentication Process	Camera Configuration, Camera Security, Frames, Brute-forcing, Password Cracking, Authentication, Strong/Weak Passwords, Wordlists, Hash, Account Security
5. OWASP top Application Security Risks	Website, Vulnerabilities, Exploitation, Information Gathering, Enumeration, Ports and Services, Administrator Privileges, User Rights, Web Server Security

Co-funded by the Erasmus+ Programme of the European Union

3D environment

The 3D environment will provide a more immersive experience for the game. At a later stage of the development, we will validate the need and its applicability of it.

Game implementation

The class will be organized into groups of 6 people. Each group is represented by 3 Blue team members and 3 Red team members. Each group member will perform a practical role interchangeably.

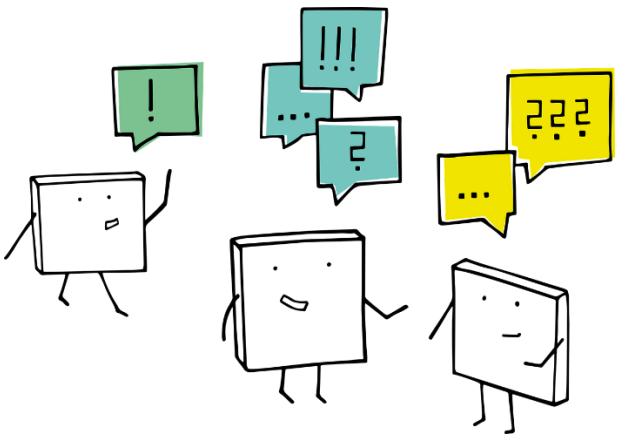
System description for the group

- 1 virtual Windows 10 machine (available for Blue Team)
- 1 virtual Kali Linux machine (available for Red Team)
- 1 virtual Linux server for simulating different services (file hosting, email server, web server, etc.) (explicitly not visible, but detectable)
- 1 virtual router to simulate router and connect all systems in the virtual network (explicitly not visible, but detectable)

Developing the scenario

The scenario “Spies at the school” was chosen (see O1). To develop the scenario, we decided to do:

- Initial brainstorming with students' impact at 20.09.21
- Iteration rounds to develop the plot and technical story 21.09-28.10.21
- Piloting with students 1.11-14.11.21
- Finalize the plot open answers at the 22.11.21 meeting
- Scenario finalized 5.12.2021



Co-funded by the Erasmus+ Programme of the European Union

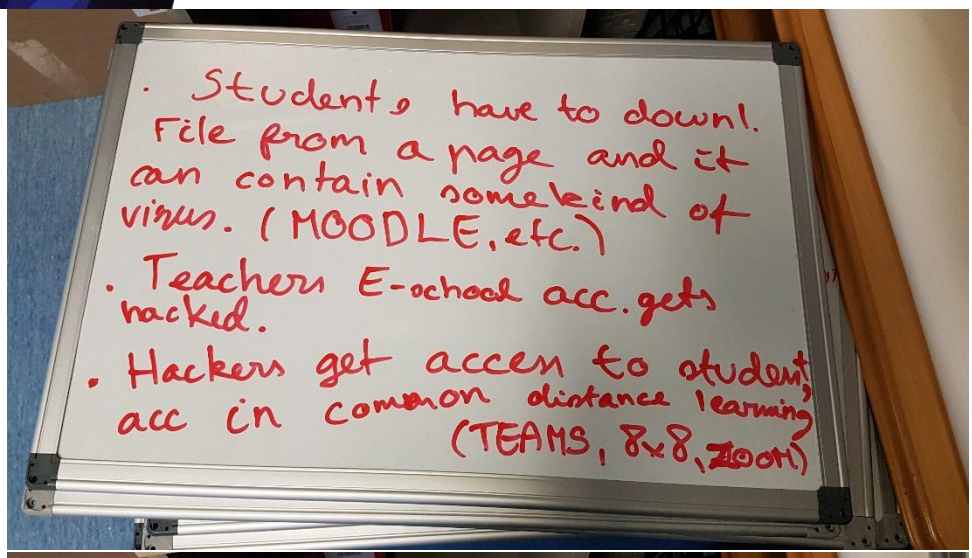
Stage I - Brainstorming

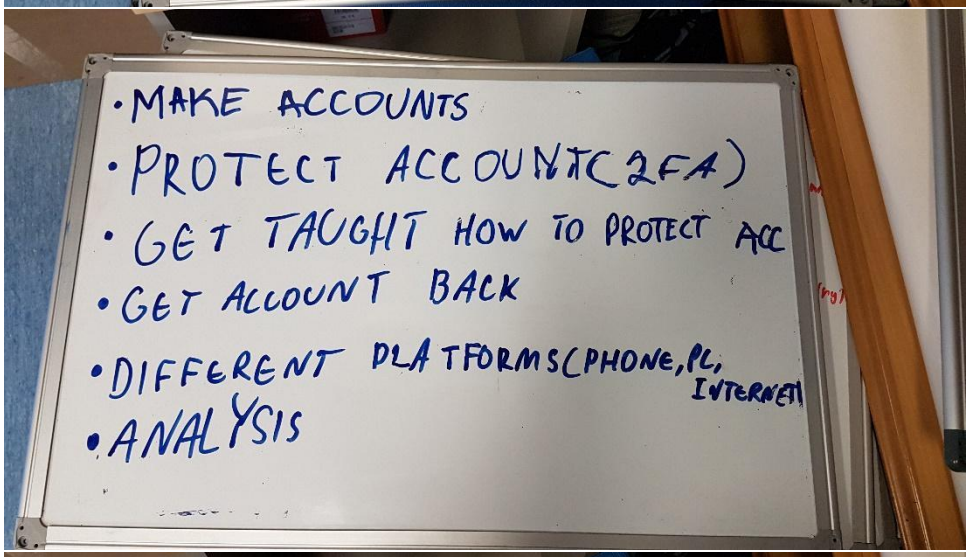
Discussing the plot took place on 20.09.21 in Estonia. We decided it will be a 1-day storyline that starts from the previous evening and the end is the grand finale of the science fair.

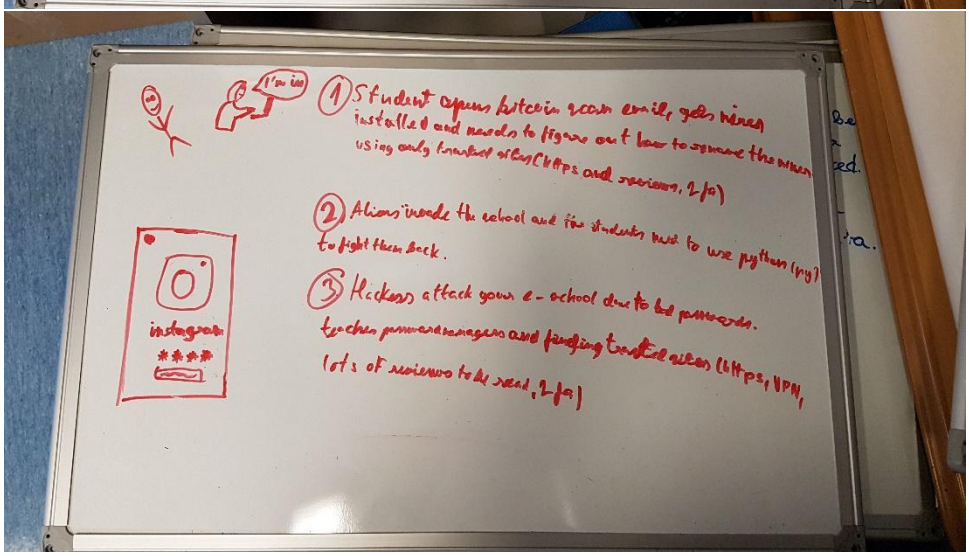
Student's impact:

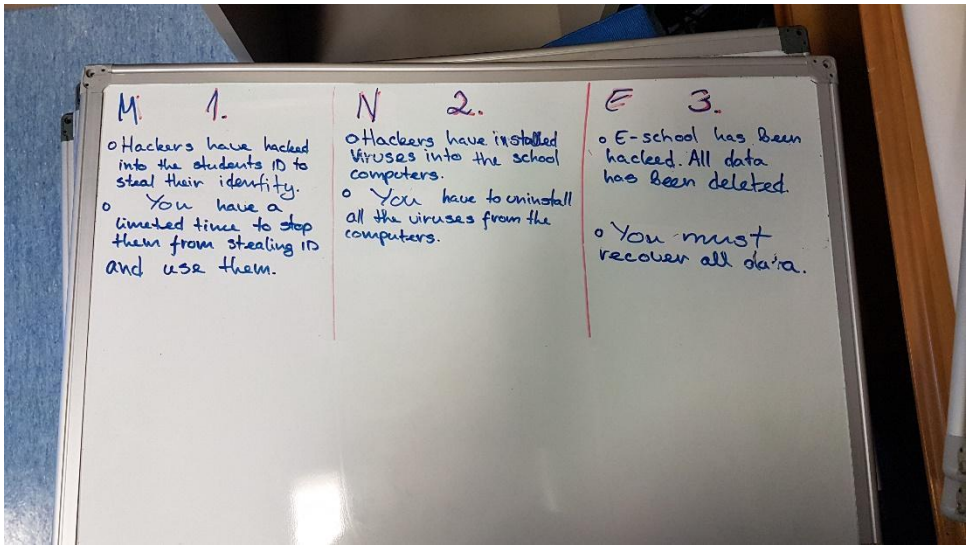
Students' insights were related to their everyday use of technologies like e-school (intranet), social media, e-mails, and Teams (cloud that they use). The environment where events were happening was home, in the school garden, cafeteria, corridor, etc.

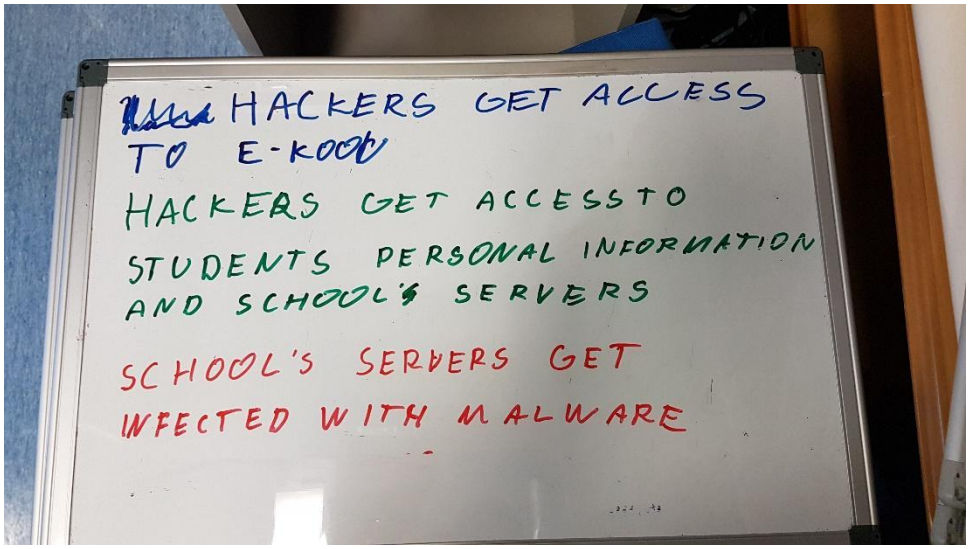
Co-funded by the
Erasmus+ Programme
of the European Union


· Students have to downl. File from a page and it can contain some kind of virus. (MOODLE, etc.)
· Teachers E-school acc. gets hacked.
· Hackers get accen to student acc in common distance learning (TEAMS, 8x8, ZOOM)

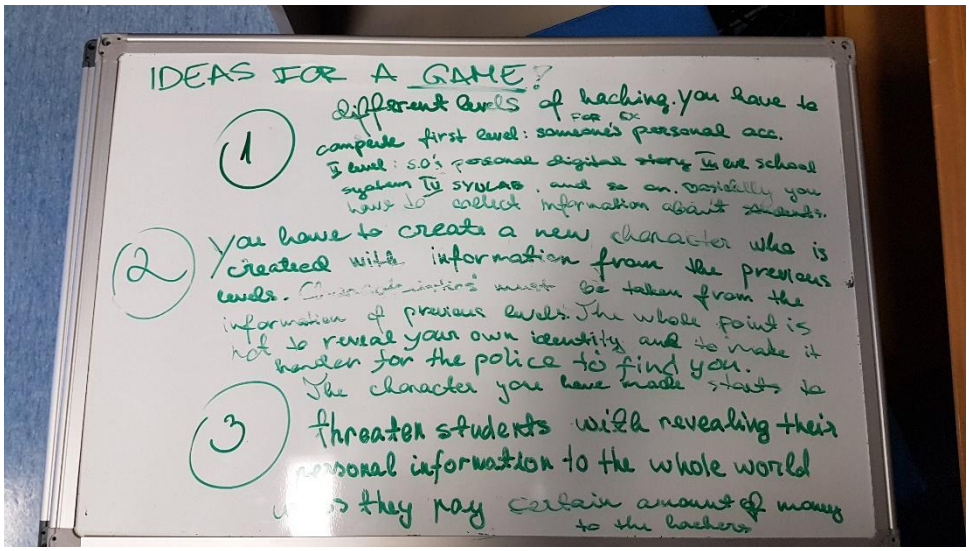

· MAKE ACCOUNTS
· PROTECT ACCOUNTS(2FA)
· GET TAUGHT HOW TO PROTECT ACC
· GET ACCOUNT BACK
· DIFFERENT PLATFORMS(PHONE, PC, INTERNET)
· ANALYSIS

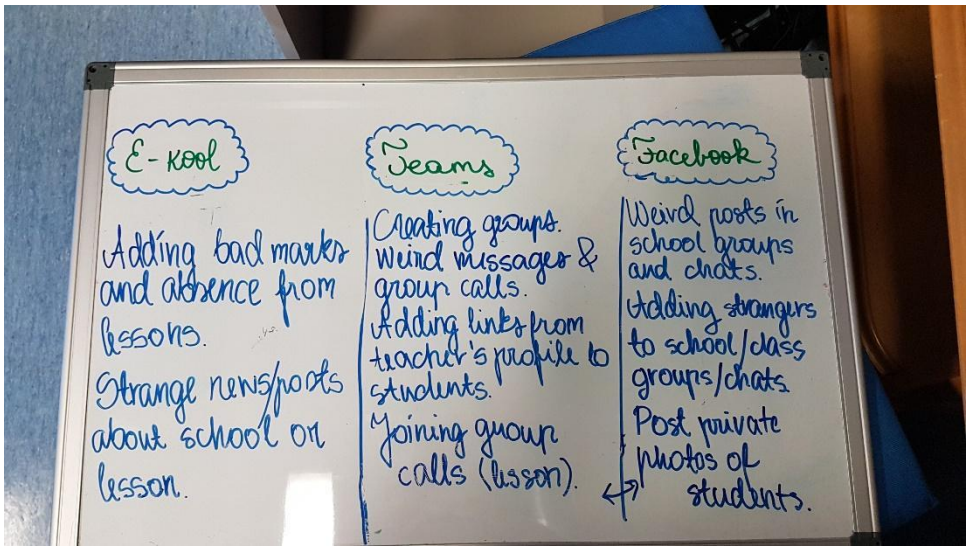

instagram
****





Co-funded by the Erasmus+ Programme of the European Union



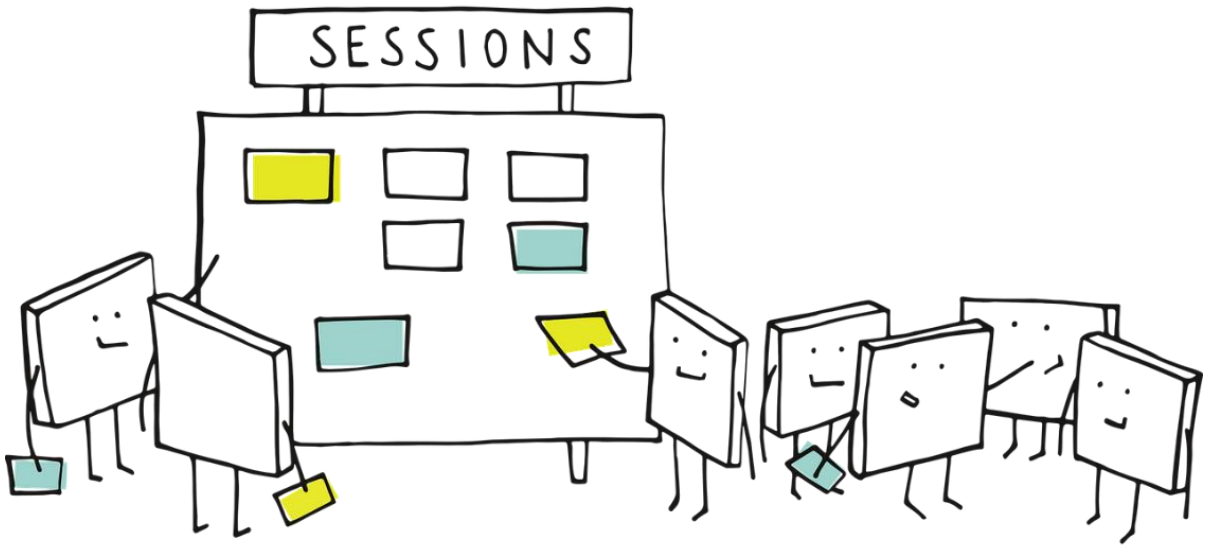


Co-funded by the Erasmus+ Programme of the European Union

Stage II – Building the Scenario

Spy's at school is one-day cybersecurity gameplay with 5 events and several roadblocks. The game starts from the previous evening and the end is the grand finale of the science fair. Many teams of participants will want to achieve the highest goal – to win it or sabotage it. You are a part of the team of red (the villain side who wants to sabotage the event to win it themselves) or blue (the defender side from the student union, who helps the school to keep the integrity of the fair competition and helps the other teams). The game can be played in two levels – beginner mode and advanced mode.

A white player is a teacher mode. The teacher has access to see the current state, restore the task, and give hints or other things that are needed.



Co-funded by the
Erasmus+ Programme
of the European Union

Event 1 Description

ROADBLOCK – Social Engineering

Description for users/pupils: It's late afternoon at 19:00. The blue team members receive an email from their teacher. This email contains two links:

- The first link is regarding the final plan for the science fair
- The second one is regarding a list of prizes

Find out if the email is valid and check the links, to whether they are safe.

The extended description for moderator/teacher: Within the email folder, the players will find other emails from the teacher that are valid. So, they will have something to compare with and find out if the last email is real and whether there is something wrong with the links.

Learning goal: Understand how fake emails are created and sent, and how to recognize spoofed sender addresses and malicious links.

RED TEAM:

Tasks:

1. Prepare the fake email and change the sender data to appear valid.
2. Put two links into an email, where one is real and the second is spoofed (will land on the site which is not in the text of the visible link).

Beginner level: Playing with a playbook or reaching the goal with one command execution.

1. Execute the command: prepare_fake_email. Check the result by opening the ready email.
2. Execute the command: place_maliciousfile. Check the result by looking into the directory file stored.
3. Execute the command: send_fake_email. Check the result by reading the log files regarding the sent email.

Advanced level: Playing without a playbook and doing all the steps needed without automation. The names of the tools are not playing any role in the preparation of the phase – we will put them into another file when ready.

1. Place the malicious file into the required directory.
2. Prepare the text file with a fake email code.
3. Use the Sendmail tool to configure and send a fake email.

Co-funded by the Erasmus+ Programme of the European Union

Questions to track progress:

1. True or false: It is impossible to spoof the sender's email address.
2. Paste text: What is the address of the email server for sending emails?
3. Single-choice: Choose the name of the HTML tag to fake an anchor text in the email body.

BLUE TEAM:

Tasks:

1. Check whether the sender's data (name, e-mail) are valid teacher data.
2. Check whether the links are genuine and safe, so one can download the data safely.

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Open the application check_email. See the results.
2. Open the application: link_checker. See the results.

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of the tools do not play any role in the preparation of the phase – we will put them into another file when ready.

1. Read the last email received from the teacher, as well as emails received in the past. Try to point out the differences between them.
2. Open the email with extended details to check the sender's validity.
3. Extract the real links hidden in the email's body and identify the malicious ones.

Questions to track progress:

1. True or false: The email was sent by the teacher.
2. Paste text: Put the anchor of a legitimate link into the box.
3. Single-choice: Choose the filename that the malicious link tries to download when being clicked.

Follow-up questions – what did you learn about the exercise?

1. What are the typical attributes of phishing emails?
2. What will be more secure to use: enriched (HTML-based) emails or plain-text emails?
3. Are attachments the only way to get your system infected?

Co-funded by the Erasmus+ Programme of the European Union

4. Should you click on a link without being sure where it redirects you to? Why?

5. How to check the real email address of the sender?

Exercise time:

- Beginner mode: 15 minutes
- Advanced: 30 minutes

Event 2 Description

ROADBLOCK – Malware

Description for users/pupils: It's 8:00 in the morning. Blue team members receive an email with an attached file. They are demanded to fill it out and send it back as soon as possible. The file is regarding a topic discussed some time ago – visiting the EU Space center. Upon the opening of the attachment, all project files get encrypted. How did this happen? Is it possible to avoid the infection? Are there backups available?

The extended description for moderator/teacher: The attached file will encrypt some files within the specific folder. There is another file with backup, which remains safe. Players need to find out how to access the backups. By analyzing the malware itself, they will also be able to reveal why the backups were not encrypted.

Learning goal: Understand how malicious attachments are created and how to back up critical files from potential corruption. Obtain a basic understanding of the malware behavior and perform basic reverse engineering.

RED TEAM:

Tasks:

1. Prepare the malicious attachment with encryption code.

2. Try to avoid detection from the antivirus by obfuscating the malicious code.

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Execute the command: prepare_malicious_attachment. Check the result by opening the log file.

2. Execute the command: obfuscate_malicious_file. Check the result by examining the logs and the directory file stored.

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of the tools are not playing any role in the preparation of this phase – we will put them into another file when ready.

Co-funded by the Erasmus+ Programme of the European Union

1. Create a malicious script for the targeted system.
2. Include the script into legitimate or look-like legitimate documents.
3. Perform obfuscation of the initial malicious code.

Questions to track progress:

1. Single-choice: What is the language of your malicious code?
2. True or false: Is it possible to hide the malicious code into a legitimate file without being detected by an antivirus?
3. Paste text: What is the SHA-1 value of the final malicious file version?

BLUE TEAM:

Tasks:

1. Is the attached file infected with a virus?
2. Open the file and examine its content.
3. Find out what happened upon the opening of the file. Try to apply backups.

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Open the application: check_attachments. See the results in the logs.
2. Open the application: backup_rollout. Follow the instructions to get the backups.

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of the tools do not play any role in the preparation of this phase – we will put them into another file when ready.

1. Apply different ways to check whether the attachment is infected.
2. Open the attached file and check for possible consequences.
3. Identify and extract the real links hidden in the email's body. Which one of them is malicious?

Questions to track progress:

1. True or false: The attachment is free of viruses.
2. Single-choice: Choose the file type that is not affected by the malicious file.
3. Paste text: Put the SHA-1 value of the malicious attachment.

Follow-up questions – what did you learn about the exercise?

Co-funded by the Erasmus+ Programme of the European Union

1. What are the typical attributes of malicious attachments?
2. Can the malicious code always be detected? If not, why?
3. Will there be consequences if you save the attachment but don't execute it?
4. How can you validate the authenticity of the attached file?
5. Are there any safe ways to check the attachments?

Exercise time:

- Beginner mode: 15 minutes
- Advanced: 30 minutes

Event 3 Description

ROADBLOCK – Network security

Description for users/pupils: It's lunchtime and everybody is offered the opportunity to connect to the school's Wi-Fi. Sometimes though, the internet is very slow due to the high number of pupils accessing it. In search of faster connectivity, the blue team members find a better Access Point. As the access point is free and requires no password, they hurry to upload the file projects to the project competition server. Later they notice that even after having uploaded the files with no errors, they do not appear on the server.

The extended description for moderator/teacher: There is a network, which is created by the threat actor to trick people into connecting to it. Since it is manipulated by the threat actor, it is impossible to ensure the security of the data and expected outcomes of the activity performed in such a network. People are presented with a fake website, an evil copy of the original one, which can trick them into uploading their files, putting credentials, or other data.

Learning goal: Understand how network traffic is manipulated and what are the consequences of using an untrustworthy access point for browsing.

RED TEAM:

Tasks:

1. Create a copy of a legal project competition website and start it on the fake web server.
2. Redirect all the network users from legal websites to this malicious twin.
3. Push the users to upload their project files into the malicious server.

Co-funded by the Erasmus+ Programme of the European Union

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Execute the command: prepare_malicious_website. Check the results by opening the fake server with the webpage.

2. Execute the command: prepare_redirect. Check the results by looking into the logs and the iptables.

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of the tools do not play any role in the preparation of this phase – we will put them into another file when ready.

1. Create a copy of the legal project competition website.

2. Start a new web server and place the malicious website files there.

3. Create new rules for the iptables to redirect the users from the genuine legal website to the malicious one.

4. Adjust the files of the malicious website to demand user credentials. Save the uploaded files on the malicious server.

Questions to track progress:

1. Single-choice: What language is the malicious website based on?

2. True or false: The threat actor workstation serves a MITM (Man-in-the-middle) attacker.

3. Paste text: What is the address of a malicious website?

BLUE TEAM:

Tasks:

1. Check whether the visited website is legal and upload the files to the server.

2. Access and connect to the secure network.

3. Find out what are the differences between legal websites and malicious twins.

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Open the application: compare_websites. See the results in the logs.

2. Open the application: back_to_secure_network. See the logs.

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of the tools do not play any role in the preparation of the phase – we will put them into another file when ready.

Co-funded by the Erasmus+ Programme of the European Union

1. Use the malicious website to provide the required information and upload files.
2. Assess different features of a legal and malicious website.
3. Change settings in Windows to turn to the secure network.

Questions to track progress:

1. True or false: Both legal and malicious websites reside uploaded files.
2. Single-choice: What changed while moving to the secure network?
3. Paste text: Find the flags of the real website within the index page metadata.

Follow-up questions – what did you learn about the exercise?

1. What are the typical attributes of fake websites? Is it always possible to recognize them?
2. Why is it not secure to type your credentials within a non-secure environment (public networks)?
3. Would you as a user still be safe if you were using a VPN?
4. Does the SSL (lock) sign-in web browser guarantee that the website is fully secure? Give explanations.
5. What are some other threats to the local networks?

Exercise time:

- Beginner mode: 15 minutes
- Advanced: 30 minutes

Event 4 Description

ROADBLOCK – Credentials and authentication process

Description for users/pupils: It's 14:00, right after the lunch break. One of the teams is discussing their website project and instantly working on their laptops. They are doing this in the hall, where there is a security camera behind them. The website they are working on is not that well secured... And the camera behind them does not seem secure either...

The extended description for moderator/teacher: The mentioned, website as well as the camera, are both residing on the same network. They also have default credentials to protect their access, which is very easily guessable or can be easily found by the use of brute forcing.

Learning goal: Understand the role of the authentication process in security and potential threats to unencrypted communications and weak passwords.

Co-funded by the Erasmus+ Programme of the European Union

RED TEAM:

Task:

1. Identify the security camera's portal and access it by using the credentials guessed or identified from network traffic.

2. Find out the website's address by using the information extracted from the camera.

3. Use brute-forcing to access the administrative portal of the group's project website.

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Execute the command: extract_weak_password. Check the results by opening the log file and tracking the events and the results that lead to the credentials leakage.

2. Open the leaked security camera picture with the website address.

3. Execute the command: bruteforce_website_admin to guess the admin credentials to the website and check the log file.

4. Enter the website using the identified credentials and observe the potential risk landscape for the owner.

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of tools do not play any role in the preparation of this phase – we will put them into another file when ready.

1. Use captured traffic files to analyze and find the credentials of the camera.

2. Use the found credentials to access the camera and extract the captured frame with the website address.

3. Apply brute-force attack on the identified website, using the provided wordlist to guess the admin credentials.

4. Enter the website using found credentials and observe the potential risk landscape for the owner.

Questions to track progress:

1. Single-choice: What is the simplest authentication process used on many websites?

2. True or false: The passwords can only be cracked online.

3. Paste text: What is the flag stored on the website's settings?

BLUE TEAM:

Tasks:

Co-funded by the Erasmus+ Programme of the European Union

1. Check if your password transmission is not affected by a simple MITM attack.

2. Create a secure configuration for the security camera.

3. Replace the weak website credentials with strong ones and protect them from password guessing/cracking.

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Open the application: analyze_password_transmission. See the result in the logs.

2. Open the application: back_to_secure_network. See the logs.

3. Use the password-safe tool to create safe passwords.

4. Go to the log change of the website to see how it was protected against password guessing.

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of tools do not play any role in the preparation of this phase – we will put them into another file when ready.

1. Using the network analysis tool to check whether the network conditions are secure.

2. Change the configuration of the security camera to secure its exposure and vulnerability by default.

3. Apply different passwords for the website and check how "strong" they are against brute-forcing.

4. Change the configuration of the website to enhance the protection against password guessing/cracking.

Questions to track progress:

1. True or false: A password of 6 characters is stronger than a password of 4 characters long.

2. Single-choice: What kind of attack was performed to obtain the credentials of the security camera?

3. Paste text: Which of the proposed passwords for the website is theoretically the strongest?

Follow-up questions – what did you learn about the exercise?

1. Why the Man-in-the-middle attack is one of the easiest and most effective to perform? How can it be detected?

2. What are the attributes of strong credentials?

3. Apart from the use of strong credentials, what are some additional measures for protecting against password guessing attacks?

4. What is a hash and how it is keeping your password secure?

Co-funded by the Erasmus+ Programme of the European Union

5. How to ensure the security of your online accounts?

Exercise time:

- Beginner mode: 15 minutes
- Advanced: 30 minutes

Event 5

ROADBLOCK – OWASP TOP application security risks

Description for users/pupils: It's 16:00 and the fair is starting. The grading is made by the teachers and judges online, by accessing the school's cloud. They have already accessed the school's cloud and they are filling a table there with all the insights. It's blind judging, so no one has a full picture of the grading process. Schools' administrator, however, sees something suspicious – grades are changing, while no one is grading the teams online.

The extended description for moderator/teacher: There is a web application for grading the teams, which shows the points of each team.

Learning goal: Understand the extended security threats to web applications and the most typical attack vectors as well as protection mechanisms.

RED TEAM:

Tasks:

1. Find hidden services on the web application and enumerate them.
2. Analyze the information found to prepare the attack on the scoring server.
3. Use security flaws in the scoring server to get control over it.
4. Find the scoring table and change the scores there to win the competition.

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Execute the command: enumerate_services. Check the results by opening the log file.
2. Read through the log file of the security assessment tool and find out which vulnerabilities look exploitable.
3. To access the scoring server, execute the command: get_control.
4. To change the scores on the scoring server and win the competition, execute: the win_competition command.

Co-funded by the Erasmus+ Programme of the European Union

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of tools do not play any role in the preparation of the phase – we will put them into another file when ready.

1. Use the available tools to assess the scoring server security and identify the hidden services.
2. Based on the information found, check the possible ways to get to the scoring server.
3. Access the scoring server using the exploitable flaw.
4. Enumerate the server for privilege escalation. Use one of the available flows to receive unrestricted access to the server resources.
5. Find the file containing the scores and change the score of your team to win the competition.

Questions to track progress:

1. Single-choice: What is the port of exploitable service on a web server?
2. True or false: Only one website can be run on the web server at a time.
3. Paste text: What is the directory name where the scoring file is stored?

BLUE TEAM:

Tasks:

1. Check the attack vector used to hack the scoring server.
2. Find the traces of the attackers.
3. Make changes to the server's configuration to protect it.

Beginner level: Playing with the playbook or reaching the goal with one command execution.

1. Open the application: analyze_attack. Check the extracted logs to understand what happened with the scoring server.
2. Open the log files of the server and find out the time and source of the attack.
3. To identify unnecessary applications running on the Webserver, open the application: running_services.
4. Close unnecessary services and applications.

Advanced level: Playing without the playbook and doing all the required steps without automation. The names of tools do not play any role in the preparation of the phase – we will put them into another file when ready.

Co-funded by the Erasmus+ Programme of the European Union

1. Use the security log application to find out what happened with the web server before, during, and after the attack.

2. Access the scoring server with one of the remote access tools and find valuable traces to identify the attackers.

3. Use the administration tools to identify unnecessary services and applications running on the scoring server.

4. Make appropriate changes to secure the server.

Questions to track progress:

1. True or false: It is impossible to have two identical websites on the same server.

2. Single-choice: What is the vulnerable service running on the server?

3. Paste text: Paste the string left by attackers in the form of comment inside the log files.

Follow-up questions – what did you learn about the exercise?

1. What is OWASP and how is it used to assess the security of the online resources?

2. What are the best practices to secure online resources and web applications?

3. Why are log files important to security?

4. What are the typical traces left by threat actors after the attack?

5. What is the difference between admin rights and user rights on the system?

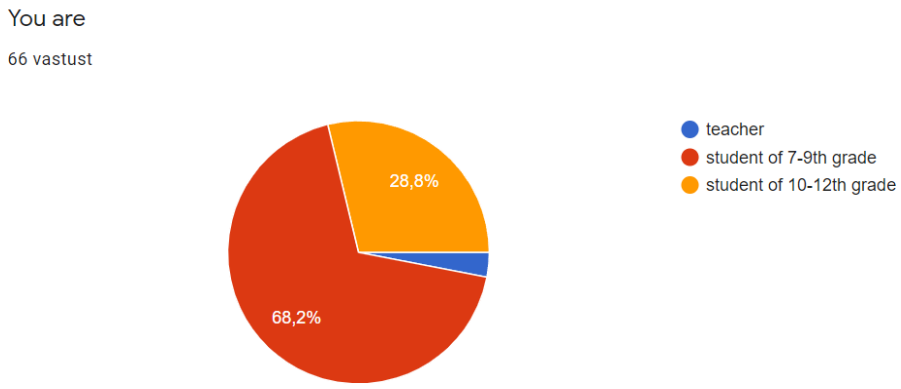
Exercise time:

- Beginner mode: 20 minutes
- Advanced: 40 minutes

Co-funded by the Erasmus+ Programme of the European Union

Stage III – Results of students piloting suggestions

School-s feedback on the scenario was taken in 3 weeks and 66 participants were giving their insight. Mainly students from 7-9th grade and 10-12th grade. 71,2% from Estonia, 25,8% from Poland and 3% from Germany. Students who answered the survey were common, had no IT specialty, and not aware of cyber security topics.

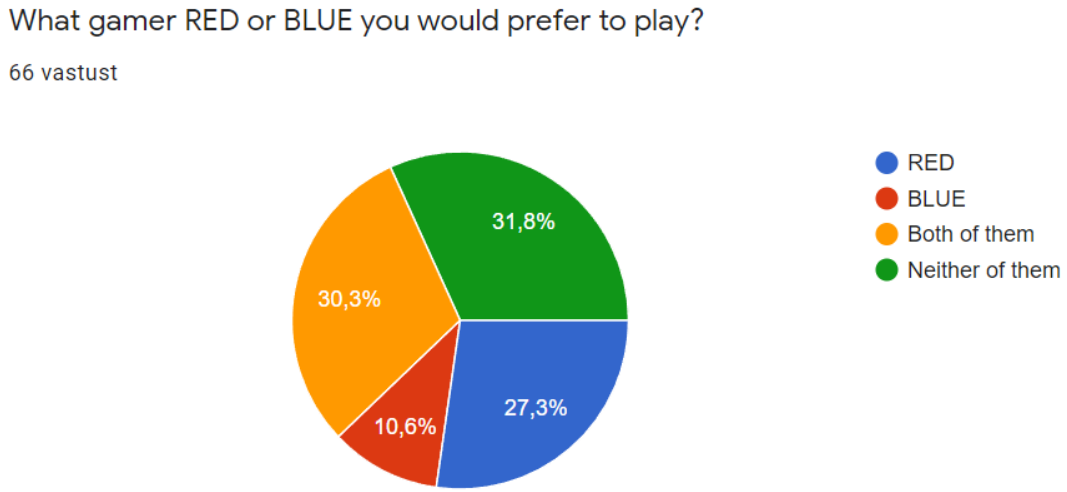


Analyzed topics and students' interest to learn about the issues:

- Social engineering - 44% (don't know/don't understand 9%)
- Malware 45% (don't know/don't understand 12%)
- Network Security 59% (don't know/don't understand 9%)
- Credentials and authentication 58% (don't know/don't understand 11%)
- OWASP 49% (don't know/don't understand 11%)

We also asked about what assignments needed more explanation than others: students from that showcase that half to ⅓ understand cybersecurity terminology better than others, but it does not mean they can perform solo in the game. Also how BLUE (defender) behaves is slightly more understandable than RED (attacker).

Co-funded by the Erasmus+ Programme of the European Union



Results:

- Most students do not know anything about IT and cyber security. Therefore
    - students need to be educated on the importance of these topics and how they benefit from knowing about these issues. This means that the playbook should also have tutorials about the topics, terminology, and maybe even videos and cases. Also how playing an educational game can benefit they're learning should be explained more.
    - The game must have 2 levels where the first level is guided and supported and the second level is for the actual testing of what has been learned and showcasing the skills and knowledge.
- Warning - if one would use the game without any preparation then students get afraid, feel misunderstood, and lost. This means the importance of prep for the teachers is vital.

How to play the game (guide, hints, and badge/ranking system)

Game guides will be provided inside the gaming environment (help reading material or guide, hints) and an introductory video (screen video) of how to use the gaming environment will also be provided. The features that the platform is designed to have will serve the purpose of assisting the pupils during the game execution as well as make the experience more exciting and enjoyable.

Hint system gives 3-level hints:

- Level 0: exercise description
- Hint level 1: keywords or comments on how to start solving the exercise
- Hint level 2: overall description of how to solve the exercise
- Hint level 3: walkthrough step-by-step of how to solve the exercise

The ranking system provides teachers to develop their strategies on how students are evaluated. The system gives every student (student team) feedback:

- time used to play the section
- hints taken
- exercises steps solved and time
- average exercise solving time and results comparison (analysis)
- feedback questions asked correctly/not correct

Co-funded by the Erasmus+ Programme of the European Union

- diploma or certificate (possible to download)

The badge system correlates to ranking: quickest solver, most correct answers to the puzzles, most excellent in exercises, fewer hints taken by the team, etc., and it's decided by the teacher to give them out.

The analysis information, in the end, is downloadable for the student. Teachers can download analyses of the players (whole group analysis and single team view).

Guide system inside the game:

- The guide should look like - a PDF guideline with pictures and videos. Read more at O2 and O5
- The hint system should be embedded in the game - read more on O4.
- Ranking system – read more on O4

Co-funded by the Erasmus+ Programme of the European Union