

ECSC 2025 Finals

Warsaw 06.10.2025 - 09.10.2025

Handbook



Table of Contents

1.	Introduction	5
1.1.	Overview	5
1.2.	About ECSC.....	5
1.3.	About the organizers	6
1.3.1.	ENISA.....	6
1.3.2.	NASK – Państwowy Instytut Badawczy	6
2.	Travel, hotels and logistics	7
2.1.	Reaching Warsaw	7
2.1.1.	By air	7
2.1.2.	By train	7
2.2.	Hotels and bookings	7
2.3.	Public transport.....	8
2.4.	Emergency services.....	9
2.5.	Suggested activities	9
3.	The venue	9
4.	Event schedule & timetable.....	11
4.1.	October 6th (3 PM - 10 PM)	11
4.2.	October 7th (9 AM - 10 PM)	11
4.3.	October 8th (9 AM – 10 PM)	11
4.4.	October 9th (2 PM – 10 PM)	11
5.	Staff and roles	12
5.1.	The steering and executive committees.....	12
5.2.	The jury	13
5.3.	Watchdogs and angels.....	14
5.4.	Organizers and technical staff.....	15
5.5.	Teams	16
6.	Competition format, setup and general rules	17
6.1.	Setup.....	17
6.2.	Day 1 – Jeopardy	19
6.2.1.	Schedule.....	19

6.2.2.	Challenges.....	19
6.2.3.	Challenge hints.....	20
6.2.4.	Network setup.....	20
6.3.	Day 2 – Attack-Defence	21
6.3.1.	Schedule.....	21
6.3.2.	Game overview.....	21
6.3.3.	Flags	22
6.3.4.	Vulnbox.....	23
6.3.5.	Game network	24
6.3.6.	Cheatsheet	27
6.4.	Rules for team composition.....	27
6.5.	Communication	28
6.5.1.	Discord account.....	28
6.5.2.	Communication rules	29
6.6.	Technical and human behaviour	31
6.6.1.	Fair sportsmanship rules	31
6.6.2.	0-day Policy	32
6.7.	Data recording and retention.....	32
6.8.	Allowed/necessary tools and hardware equipment	32
6.8.1.	Software equipment.....	32
6.8.2.	Hardware equipment.....	32
6.9.	Penalties and complaints	33
7.	Scoring system	34
7.1.	Jeopardy scoring.....	34
7.2.	Attack-Defence scoring	35
7.3.	Aggregated scoring	40
8.	Challenge categories and distribution	40
9.	Platforms and API documentation.....	41
9.1.	Jeopardy platform	41
9.1.1.	/challenges	41
9.1.2.	/challenge/<challenge_name>.....	41

9.1.3.	/scoreboard	42
9.1.4.	/activity	42
9.1.5.	/profile.....	42
9.2.	Attack-Defence platform and APIs	42
9.2.1.	/api/v1/services : Fine-grained service info	43
9.2.2.	/api/v1/teams : Fine-grained team info	43
9.2.3.	/api/v1/score : Fine-grained scoring info	44
9.2.4.	/api/v1/attack_info : Fine-grained attack info	45
9.2.5.	/api/v1/current_round : current round time and id	45
9.2.6.	/api/v1/next_round : next round time and id.....	46
9.2.7.	/api/faustctf2024/teams.json : Attack Info (FaustCTF 2024)	46
9.2.8.	/api/saarctf2024/attack.json : Attack Info (SaarCTF 2024).....	46
9.2.9.	Parameters	48
9.2.10.	Scoreboard	48
9.2.11.	Contact, disclosure, bug bounty	48
9.3.	Discord server	49
9.3.1.	Authentication system.....	49
9.3.2.	Ticketing system	50
9.3.3.	Teams' manual	51
9.3.4.	Ticket creation and lifecycle.....	53
9.3.5.	Staff Manual	55
9.3.6.	Managing tickets.....	56
9.3.7.	Channels and Categories	57
9.3.8.	Discord Roles	59
9.3.9.	Role sets.....	60

1. Introduction

1.1. Overview

This document is intended to be a comprehensive and self-contained guide for ECSC2025 participants. The current version is a draft, subject to changes depending on feedback and possible organizational updates.

This document is based on the ECSC documentation including the Charter, the Rules, and the Code of Conduct, and all participants should be familiar with this documentation. Some parts of this document (e.g. communication rules, roles and structure of the document) are based on the CINI Cybersecurity National Lab ECSC 2024 Handbook.

1.2. About ECSC

The European Cybersecurity Challenge (ECSC) is an annual event happening at an international level, that brings together the most promising talents in the cybersecurity sector. The competition acts as a unique platform dedicated to sharing knowledge, ideas and skills through practical tests on topics representing the state of the art of cybersecurity. During the event, teams from participating countries compete in a series of Capture-The-Flag format challenges over two days to determine the best prepared nation.

Originating in 2014, the event is coordinated by ENISA, the European Union Agency for Cybersecurity. Each year a different host country is responsible for the organization.

For the 2025 edition, the event will be held in Warsaw, Poland, from October 6th to October 9th. It will be organized by Państwowy Instytut Badawczy NASK (Research and Academic Computer Network - National Research Institute) in collaboration with ENISA.

The event offers a unique opportunity for participants to interact with other young people from across all of Europe, as well as receive mentorship, broaden their skills and establish lasting connections in the cybersecurity field, benefiting their careers and becoming part of an extremely active community of young cybersecurity talents.

1.3. About the organizers

1.3.1. ENISA

The European Union Agency for Cybersecurity, ENISA, is an institution whose main goal is to achieve a high common level of cybersecurity across Europe. Established in 2004 and strengthened by the EU Cybersecurity Act, ENISA contributes to EU cyber policy, enhances the trustworthiness of ICT products, services and processes with cybersecurity certification schemes, cooperates with Member States and EU bodies, and helps Europe prepare for the cyber challenges of tomorrow. Through knowledge sharing, capacity building and awareness raising, the Agency works to keep Europe's society and citizens digitally secure.

1.3.2. NASK – Państwowy Instytut Badawczy

Państwowy Instytut Badawczy NASK (Research and Academic Computer Network - National Research Institute) is the Polish national research institute responsible for cybersecurity and digital infrastructure. 30 years ago, NASK introduced the Internet in Poland, now we operate three research divisions and 14 specialized departments dedicated to cybersecurity, artificial intelligence, and cloud computing.

Developing proprietary cybersecurity solutions, conducting security audits, providing support for businesses in the digitalization process, taking schools online, creating secure document management systems for government agencies and training public officials, as well as taking measures to combat online fraud, removing illegal content and spreading awareness about digital threats – the highlights of our daily job make Poland and Europe a safer space for everyone.

2. Travel, hotels and logistics

2.1. Reaching Warsaw

Useful information about travelling to Warsaw can be found on the Warsaw Tourism website:

<https://go2warsaw.pl/en>

The following sections briefly describe some of the most convenient options.

2.1.1. By air

Warsaw has two international airports: Warsaw Chopin Airport and Warsaw Modlin Airport. Warsaw Chopin Airport is the main airport and is located approximately 10 kilometers from the city center. Warsaw Modlin Airport is located around 40 kilometers from the city center. Both airports offer various transportation options including trains, buses, and taxis to reach the city center.

2.1.2. By train

Warsaw is a major railway hub with connections to many European cities. The main railway station is Warszawa Centralna, which is located in the city center. Other important stations include Warszawa Wschodnia and Warszawa Zachodnia. High-speed trains and international connections are available, making it easy to reach Warsaw by train.

2.2. Hotels and bookings

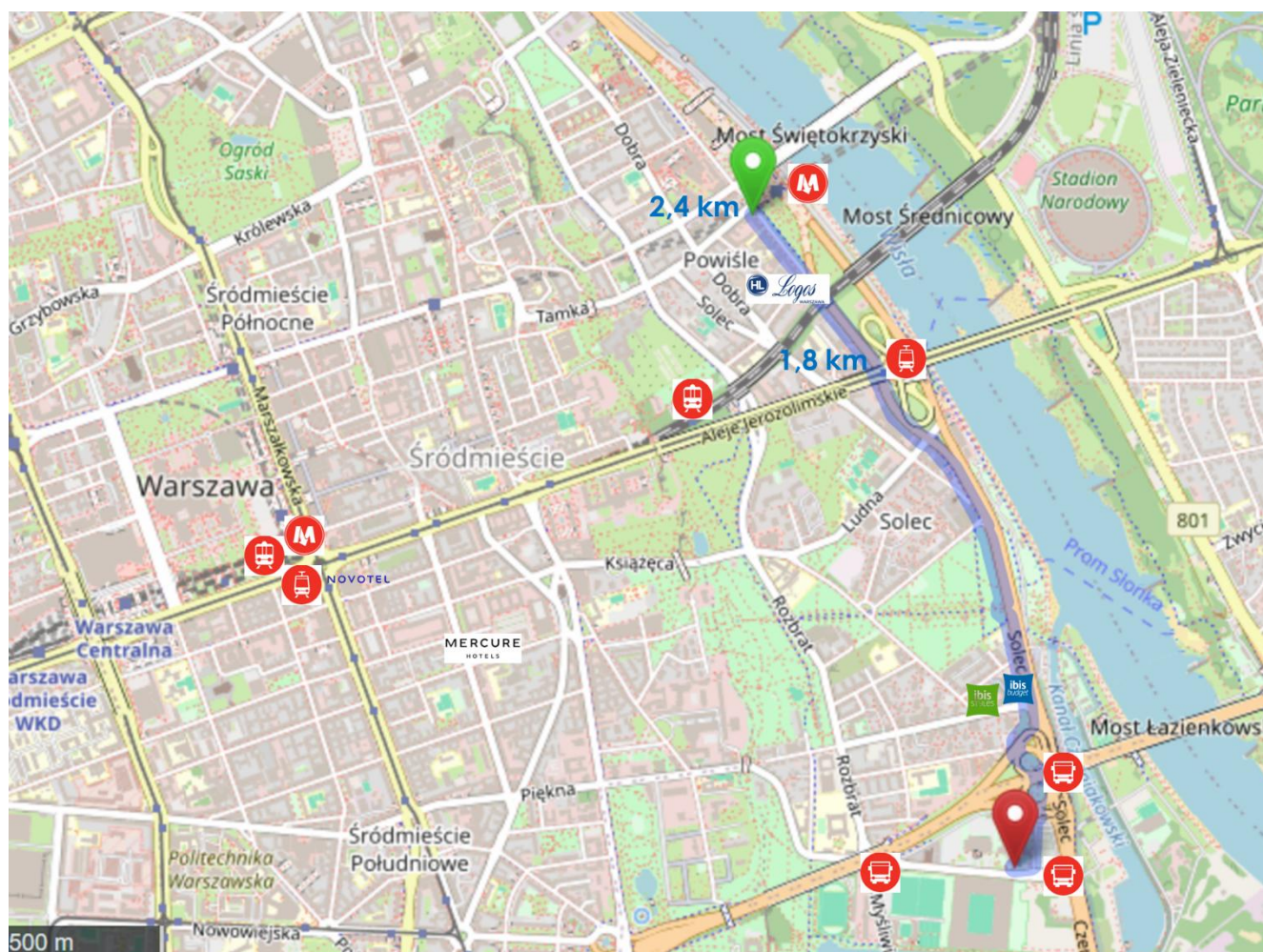
The recommended hotels have been chosen with regard to the price, quality and location to best suit your needs. They've been sorted by an estimated cost of the stay:

- LOGOS Warszawa (Wybrzeże Kościuszkowskie 31/33; 2 km from the venue)
- Ibis Budget Warszawa Centrum (Zagórna 1; 0.7 km from the venue)
- Ibis Styles Warszawa Centrum (Zagórna 1; 0.7 km from the venue)
- Ibis Warszawa Stare Miasto (Muranowska 2; 5 km from the venue)
- Mercure Warszawa Grand (Krucza 28; 2.1 km from the venue)
- Novotel Warszawa Centrum (Marszałkowska 94/98; 2.6 km from the venue)

2.3. Public transport

Public transport in Warsaw is managed by WTP (Warszawski Transport Publiczny) and provides extensive coverage of the city via metro, buses, and trams. Tickets can be obtained through various apps or purchased directly in the ticket machine on the bus/tram or before entering the metro stations. More information can be found on the website: <https://www.wtp.waw.pl/>

For your convenience, we provide a simplified map of the City Centre, including hotels and public transport stops that may be used to reach the venue (pinned in the lower-right corner).



2.4. Emergency services

The main number for emergency services in Poland is 112, which handles all emergencies and redirects to the most appropriate service. Information about other emergency numbers can be found on the Warsaw Tourism website.

2.5. Suggested activities

Warsaw offers a variety of activities and attractions for visitors. Some suggested activities include visiting the Old Town, exploring the Royal Castle, walking through Łazienki Park, and experiencing the vibrant cultural scene at the POLIN Museum of the History of Polish Jews. More information about activities and attractions can be found on the Warsaw Tourism website.

3. The venue

The Central Sports Centre Torwar is an indoor arena in Warsaw, Poland. It opened in 1953 and seats 4,824 people in the stands, with up to an additional 1,480 seats available to add on the floor, located at ul. Łazienkowska 6a in Warsaw.

The ECSC 2025 is a closed for public event and you must be registered as a team member or as a country delegation member to enter the venue.





4. Event schedule & timetable

(time zone: CEST)

4.1. October 6th (3 PM - 10 PM)

- Registration of participants (3 PM)
- Checking the team workstations (4 PM)
- Official opening (6 PM)
- Dinner & networking (8 PM)

4.2. October 7th (9 AM - 10 PM)

- Arrival of participants (9 AM)
- Competition - Jeopardy (10 AM - 6 PM)
- Dinner & networking (7 PM)

4.3. October 8th (9 AM - 10 PM)

- Arrival of participants (9 AM)
- Competition – Attack-Defence (10 AM - 6 PM)
- Dinner & networking (7 PM)

4.4. October 9th (2 PM - 10 PM)

- Closing ceremony, results announcement (2 PM)
- Official dinner (5 PM)
- Afterparty (7 PM)

A current schedule can be found on the ECSC 2025 official website at <https://ecsc2025.pl>.

5. Staff and roles

5.1. The steering and executive committees

The Steering Committee consists of representatives from the countries participating in the ECSC competition. It is the governing body of ECSC, ensuring the organization fulfills its mission and achieves its objectives. The Steering Committee oversees ECSC's operations, sets policies, makes major decisions, and is legally responsible for ECSC's activities. It establishes standing committees (such as the executive committee) to address specific issues.

The Executive Committee is established on behalf of the Steering Committee. The Steering Committee provides guidance and oversight to ECSC's Executive Board. Its responsibilities include setting the overall strategic direction of the organization, establishing policies and procedures, monitoring organizational performance, ensuring financial sustainability, accountability, transparency, and communicating with stakeholders. Together, the Steering Committee and Executive Board work to ensure the organization's success.

The steering committee can be identified by their **orange** badge.

5.2. The jury

Jury members are responsible for impartially resolving disputes and maintaining fairness in the competition. Unlike the Steering Committee, the jury is a small body (3-5 people) with high experience and diverse skill sets.

The main responsibilities of the jury include:

- Receiving and handling complaints from participants during the event.
- Coordinating with watchdog managers and watchdogs to address possible unfair behaviors during the competition.
- Making decisions in emergency situations (e.g., broken challenges, unfair advantages to some teams due to infrastructure issues, etc.).
- Acting as an impartial authority to control and evaluate decisions made by the organizers.

The jury can be identified by their **black** badge.

5.3. Watchdogs and angels

Watchdogs are expected to monitor the correct development of the competition in accordance with the rules, the code of conduct, and any jury or Steering Committee decisions. They are the primary point of contact for players during the competition (in addition to the ticketing system) in case of complaints and/or questions about the rules. Watchdogs report their observations to watchdog managers, who act as a direct interface with the jury and organizers to handle non-trivial situations.

Angels are non-technical collaborators needed to ensure a smooth event. They mainly handle logistical aspects such as registration, access control, and food and beverage administration.

Watchdogs are selected with a public call among CTF players, students and former ECSC players with past experiences in the CTF or large event management field.

Watchdogs can be identified during the event by a **red** badge and t-shirt, with “WATCHDOG” written on the back. Angels can be identified too, with their white badge and t-shirt, with “ANGEL” written on the back.

5.4. Organizers and technical staff

Organizers and technical staff serve as the core team responsible for running the event.

Organizers are in charge of venue logistics, which includes setting up several areas, coordinating catering, and managing the event schedule.

Technical staff oversee the competition infrastructure, including the creation and development of challenges, testing and deployment, as well as the overall format, rules, and organization of the competition.

During the event, they address the competition-related tickets from participants and assist the jury by providing all necessary technical materials for their investigations. Additionally, they manage communication with players and moderate the Discord server.

Organizers can be identified during the event by a **light purple** badge and t-shirt, and technical staff by a **green** badge and t-shirt.

5.5. Teams

A team is the entire set of people representing one of the countries playing in the ECSC. The main components of a team are the team players, which are the ones who actively play the competition games in the game arena, at the team's table.

Each team has at least one coach. The coaches do not take part in the competition challenges and there is no limitation on their age. The coaches are responsible for the well-being and behavior of their players.

The coaches should ensure that essential information coming from or intended for the team players is communicated, understood, and acted upon.

The coaches are physically separated from the team players during the competition. In case a team has multiple coaches, one of them must be selected as the main coach. The main coach is the primary point of reference for the team. The main coach can convey the team's requests to the jury and raise complaints on behalf of the team whenever it is necessary. In case the main coach is not available during the competition, an alternate main coach must be nominated temporarily among the other team's coaches.

Among its players, each team nominates a captain, which is the main interface for the team players inside the arena to the external world.

Players can be identified by a **light blue** badge, coaches by a **yellow**, and the rest of the delegations will wear a **light gray** one.

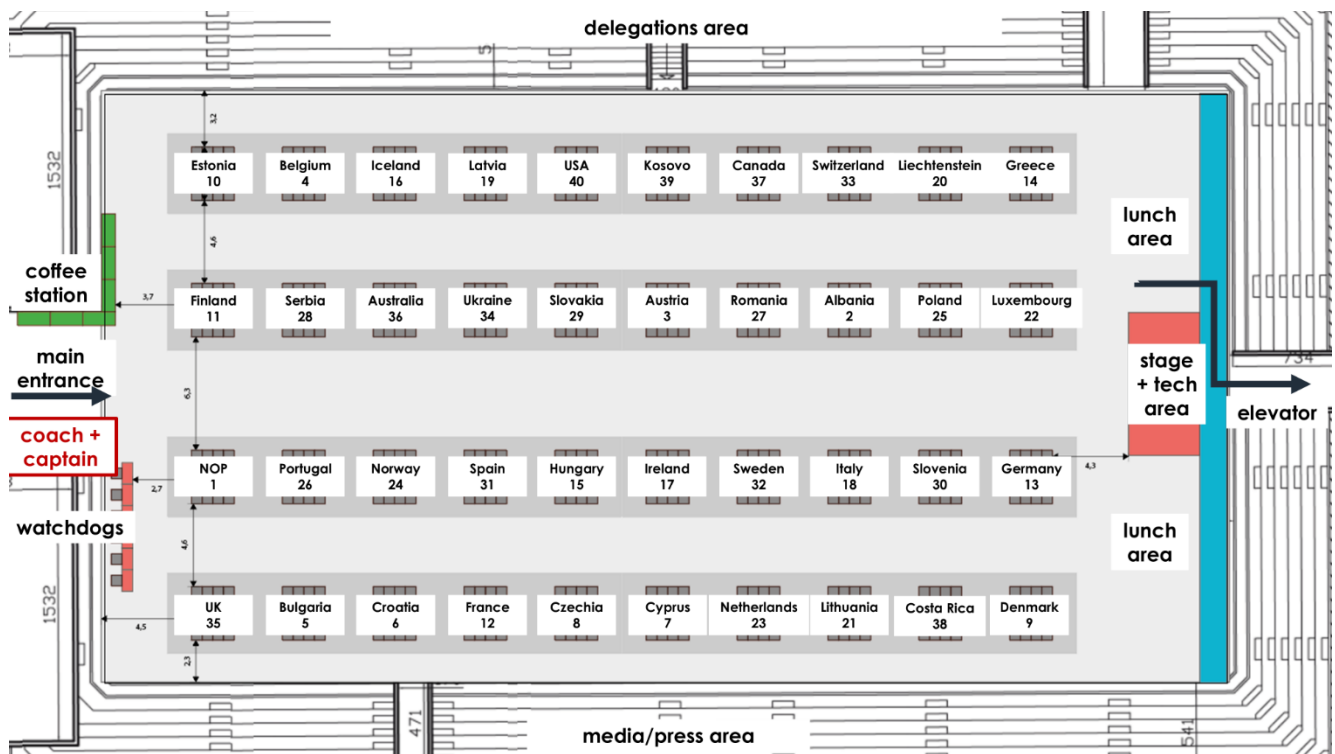
6. Competition format, setup and general rules

The event features two separate, 8-hour long competition days played on-site. The first day is the jeopardy CTF competition while the second is the attack-defence CTF competition.

6.1. Setup

The CTF infrastructure will be hosted in the cloud. The participants will connect to the cloud using infrastructure prepared by the organizers on-premises.

Each team has its own table in the game arena. The tables' assignment is as follows and it includes country name with its number in Attack-Defence infrastructure.



Each table has a switch, which has at least 12 Ethernet ports available (i.e., excluding ports already used by the organizers). Each table has at least 12 power sockets available (type E compatible), for a total of at least 4 kW power capacity per table.

Team tables may include additional hardware or network devices provided by the organizers for challenges or infrastructural support. This equipment, sourced from the venue, sponsors, or the organizers themselves, **must** be left on the table at the end of the competition, unless otherwise specified.

Teams can connect to their switches through ethernet cables to connect to the Internet. Players are expected to bring their own computer, ethernet cables (and adapters, if required). The wired infrastructure has a dedicated uplink of 20 Gbps and VMs and players have 1 Gbps access to the game network per table, but keep in mind that the overall capacity can vary depending on the type of traffic and the routes used, making it difficult to provide an exact estimation. Players are supposed to avoid generating excessive traffic; team-specific rate limits can be imposed to limit infrastructure disruptions.

6.2. Day 1 – Jeopardy

Day 1 of ECSC 2025 (7th October, 2025) is the Jeopardy competition!

While the competition will follow a typical, modern jeopardy-style CTF with dynamic scoring, the following subsections explore certain aspects of the competition in more detail.

We strongly encourage players, captains, and coaches to carefully read the sections below, but also to familiarize themselves with other jeopardy-style CTF competitions to better understand this format.

6.2.1. Schedule

Jeopardy competition lasts 8 hours, with an additional hour for writeup submission.

Time (CEST)	Event
10:00	Jeopardy competition starts. Challenges become accessible, flags and writeups can be submitted.
18:00	Jeopardy competition ends.
19:00	Writeup submission deadline.

In case of unforeseen circumstances the schedule can be altered. In such case all the teams shall be informed. In case this would affect the competition length, exact details of schedule change will be decided by the Jury.

6.2.2. Challenges

Day 1 Jeopardy competition will consist of 30 CTF challenges. A challenge is considered solved when a player successfully acquires a flag and submits it on the Jeopardy Platform during the competition.

Unless a challenge's description says otherwise, the flag format matches the following regular expression: `^ECSC{ . * }$`

Additionally, **players are required to upload a writeup for each solved challenge before the writeup submission deadline.** A working solver script or exploit is considered a writeup equivalent (must be delivered in a form that can be easily analyzed if needed).

Important: Failure to submit a valid writeup before the deadline may result in penalties for the team, as determined by the Jury.

All 30 challenges will be published at the start of the CTF or shortly thereafter in case of unexpected issues with individual challenges.

If, during the competition, a challenge is found to have issues, one of the following actions will be taken:

- In case a challenge is deemed fixable, it will be promptly repaired and re-deployed.
- In case a challenge is deemed unfixable, Jury shall decide whether to remove the challenge or — if feasible — replace it.
- In case a challenge has unintended solution(s), the organizers can prepare a new version without the unintended solution(s) and the Jury can decide to add this challenge to the competition's challenge set.

Given the above, it is possible for the competition to effectively end with less than or more than 30 challenges.

6.2.3. Challenge hints

While unlikely, challenge hints may be released during the competition. The Jury will make the final decision on whether a hint is issued and on its exact wording.

6.2.4. Network setup

No special network setup will be required. There are however other network considerations to discuss.

Most importantly, both **the Jeopardy platform and all Jeopardy CTF challenges will only be accessible from the local competition network (ethernet link)**. This means that challenges with server-side components will not accept inbound connections from hosts external to the local competition network unless otherwise specified.

At the same time selected challenges with server-side components will have internet access and will be able to connect to external hosts. As is typical for CTF competitions, at times players can be expected to operate external servers, be able to set up external domains (DNS), be able to acquire HTTPS certificates, and take similar actions.

6.3. Day 2 – Attack-Defence

The Attack-Defense CTF will take place on 8th of October, starting at 10:00 CEST and lasting 8 hours until 18:00 CEST.

Attack-Defense CTFs are a type of cybersecurity competition in which participating teams host services and attempt to exploit each other over a shared, private network. The goal of the game is to *earn points* by stealing secrets stored in your opponents' service instances, and to *avoid losing points*, by preventing your own secrets from being stolen and submitted, all the while keeping the services available and functioning. The team with the most points by the end wins.

IMPORTANT NOTE: Information in the Attack-Defence Wiki (<https://wiki.ad.ecsc2025.pl>) takes precedence over this Handbook. This applies to sections 6.3., 7.2., and 9.2.

6.3.1. Schedule

The schedule for the day of the Attack-Defense CTF:

Time (CEST)	Event
09:00	Players can download wireguard configuration to access the game network
09:30	Players can start their vulnbox using the platform
10:00	The Attack-Defense CTF officially begins: SSH access to vulnboxes is unblocked API endpoints are available to test connectivity Flag submission at 10.42.251.2:1337 accepts connections Scoreboard at 10.42.251.2 is available (empty) Team router and scoreboard respond to pings VPN Connection works within but not between teams
11:00	Network opens and teams can communicate with other vulnboxes
17:00	The scoreboard scores are frozen
18:00	The Attack-Defense CTF officially ends

6.3.2. Game overview

Each team is given root access to one cloud-hosted Linux-based virtual machine that exposes vulnerable services to other teams over a private virtual network.

Over the course of every round, lasting 60 seconds, so-called checkers store text snippets called flags in the services on each team's vulnbox and test their functionality to make sure they are working as intended. Extracting these flags from other teams' services and submitting them to a central flag submission each round to earn **ATK-points** is the primary goal of the game.

Note: flag stores

A checker may store multiple unique flags each round in distinct areas of a service, and there may be more than one intended vulnerability to reach each one.

To incentivize teams to keep their services available to other teams to exploit, a series of checks is performed each round against every service of every team by the organizers' checkers. These tests define the so-called *Service-Level Agreement* (SLA); the functionality required for a team to earn **SLA-points** each round.

Note: attack info

Checkers may provide hints for successfully stored flags to help guide exploits. In some cases, this info is crucial to exploiting the vulnerability at all. It can be retrieved via the attack api.
--

Each round a team receives **DEF-points** for every service. The amount of points earned is highest when the service is unexploited and decreases with the amount of other teams exploiting it.

These points combine to calculate the team score using the scoring formula (see section 7.2.).

6.3.3. Flags

Each flag is matched by the regular expression `/^ECSC\[A-Za-z0-9-\_]{32}\}$/`

Checkers retrieve flags from the previous **5** rounds in addition to the current round to enable exploits which take longer than a single round to complete. The SLA penalty for missing any of the flags in this *retention period* incentivizes teams to keep them available for capture.

Flags will award points upon submission for 5 rounds including the round they were deployed in.

Each flag consists of a prefix and suffix, that wrap a base64-encoded¹ payload with the following format:

- 2 bytes: round id

- 2 bytes: team id
- 2 bytes: service id
- 2 bytes: flagstore id
- 16 bytes: SHA256-HMAC (of first 8 bytes)

Players can submit stolen flags by sending them line-delimited in a plain TCP connection to 10.42.251.2 on port 1337. This must be done via the game network, since the source ip is used to determine the submitting team.

For each line, in the order that they are received, the flag submission will return one of the following results on a new line:

- [OK]: The flag is valid and was accepted
- [ERR] Invalid format: The flag is malformed
- [ERR] Invalid flag: The signature of the flag is incorrect
- [ERR] Expired: The flag was submitted after the retention period
- [ERR] Already submitted: The flag has already been submitted by this team
- [ERR] Can't submit flag from NOP team: The flag submitted is from NOP
- [ERR] This is your own flag: The flag is from the submitting team

6.3.4. Vulnbox

Cloud-hosted vulnboxes and exploiter VMs are provided by the organizers to all teams.

The team router provides access to the internet via a public IPV4 and IPv6 address, as well as to the game network through WireGuard. The vulnbox and exploiter are not reachable from the internet via their public IPv4 or IPv6 address.

Vulnboxes and exploiters run on OVHCloud infrastructure as **C3-32** instances in Warsaw's WAW1 data center. With **16 cores, 32 GB of ram and 400 GB of storage** each, the vulnbox and exploiter should have enough resources to run the services, exploit their vulnerabilities, and handle exploit as well as checker traffic.

Self-hosting services is not officially supported, and carries a latency and bandwidth penalty.

The Wireguard connection from the vulnbox to the router is established inside a cloud-provider network, which allows for a faster connection than the public interface a self-hosted vulnbox would have to proxy traffic over.

Each vulnbox is configured to accept SSH keys by the organizers in addition to those configured by the teams via the platform. Teams are free to remove these SSH keys, however doing so limits the amount of support and automated fixes the organizers can provide.

Vulnboxes and exploiters are assigned an IPv4 address in the Game Network and the Team Cloud Network.

Vulnboxes are provisioned with an iptables firewall that drops all traffic except:

- Ingress connections from the game network to ports exposed by docker
- Ingress connections to SSH (port 22)
- Established connections
- Egress connections

Additionally, the vulnbox is firewalled at the cloud-provider level from external access to ensure all game-related traffic runs through the router and misconduct can be accounted for.

Players may start their vulnbox **30 minutes** before the CTF begins through the platform, but SSH access will be prevented through the team router until the CTF has officially started.

6.3.5. Game network

The A/D CTF takes place in the game network. Players connect to this network through a WireGuard tunnel. WireGuard configuration files will be made available one hour before the game starts via the platform. They can be used with standard WireGuard tooling such as wg-quick.

Every WireGuard configuration file allows exactly one host to connect to the game network. Trying to use the same configuration on multiple hosts simultaneously will likely make the connection unstable for all hosts using that configuration. We provide three configuration files to every player, and 10 additional configs per team, which are only accessible to captains and players promoted to *Technician* on the platform.

The VPN endpoint will be accessible via IPv4 for the infrastructure demo on 28.09.2025, and most likely only via IPv6 for the final CTF.

The configuration files contain credentials and information about the VPN endpoint. **Do not share any of this information with anyone outside your team.** This includes the endpoint information - it is different for every team. Modifying the configuration files should not be necessary.

The VPN connection is only used to access the game network. Most importantly, your vulnbox, your team members, other teams' vulnboxes, the flag submission, attack info, and the scoreboard. Your devices **can not** use the VPN connection to access the internet.

The game network uses the address range: $10.42.0.0/16$.

Every team has its own subnet, the *team network*: $10.42.<TEAM>.0/24$.

Each team's *vulnbox* gets the ip $10.42.<TEAM>.2$.

Every team network has a *gateway* $10.42.<TEAM>.254$, controlled by the infrastructure.

The NOP (**non**-playing) team is assigned the team id **1**. Therefore, the NOP vulnbox is available at the ip $10.42.1.2$.

The gameserver has the ip address $10.42.251.2$, and hosts the scoreboard (port **80**), the flag submission (port **1337**) and the attack api (port **8080**).

Connecting to vulnboxes is only possible through the VPN, including your own team's vulnbox. Vulnboxes can access the internet, but they do not have a designated public ip. They essentially sit behind a NAT. This is done to ensure all interactions can be properly observed by the organizers and any misconduct can be accounted for. It also minimizes opportunities for "opsec fails" and attacks that are likely against the rules of conduct.

Note that the overall bandwidth for each team's in- and outbound VPN traffic is capped at 1.8gbit/s. Keep this bandwidth budget in mind if you intend to extract pcaps from your vulnbox to another host.

Each team's router, exploiter and vulnbox are also connected to an internal cloud network without WireGuard. Connectivity within this network is mostly unrestricted.

This network has the address range: $10.43.<TEAM>.0/24$

The vulnbox gets the ip $10.43.<TEAM>.2$

The exploiter gets the ip $10.43.<TEAM>.3$

The gateway gets the ip $10.43.<TEAM>.254$

These addresses of other teams are not reachable via the game network.

Connections originating from outside your teams network are anonymized to prevent checker fingerprinting.

All connections from checkers and other teams will appear to originate from 10.42.<TEAM>.254 – your team's *gateway*.

You should consider the following when encountering network issues:

- **Packets with IP header options are rejected** since they are likely used unintentionally and are easily fingerprintable. The game network will reply with ICMP type Destination unreachable (3) and code Administratively Prohibited (13).
- **TCP headers are normalized** to prevent teams from telling apart checkers from exploiters via patterns in TCP options / flags usage.
- **TTLs of packets entering team networks are capped at 32** such that traffic to vulnboxes arrives with the same TTL regardless from where it was sent.
- **TTLs are not decremented in our router mesh** to prevent traceroute-ing of router topology.
- **We artificially introduce latency and degrade performance** to prevent fingerprinting based on network conditions / response time of external exploiters.
- **MTU negotiations are dropped** to prevent using cached PMTU values to keep a single host fingerprinted.
- **TCP MSS** is set to a fixed value (MTU - 40) to prevent fingerprinting and potentially causing high packet rates.

Despite all of this, it might still be advisable to run exploits from the vulnbox if you suspect your traffic is being fingerprinted.

While traffic within one team is almost unrestricted, traffic *between* teams is heavily restricted. The only allowed traffic is:

- TCP connections, but only to vulnboxes
- ICMP traffic of type Echo Request (8), Echo Reply (0) and Destination unreachable (3) with codes 0, 1, 2, 3, 10, and 13.

Note that attacking other teams hosts other than the vulnbox is generally prohibited.

Additionally, the game firewall blocks connections to other teams' vulnboxes in the port range 8000 to 9000 (exclusive). Using this port range for tooling deployed on the vulnbox ensures that it can not be easily accessed by other teams - even if the host firewall would allow it.

Finally, the network drops malformed packets, packets with invalid checksums, and packets that can't be attributed to connections. These firewall rules are designed to protect teams and

infrastructure from malicious traffic that does not target the services. You generally won't receive ICMP messages for prohibited traffic, it will just be discarded.

We impose a bandwidth limit on the sum of traffic between any two distinct teams. This limit is **at least 20mbit/s**.

Note that the bandwidth achievable for a single connection between teams may be significantly less than **20mbit/s** due to how our traffic anonymization policies affect TCP throughput.

We also impose a bandwidth limit of **at least 20mbit/s** on connections to the gameserver per team.

Teams should monitor the traffic generated by their exploits to avoid becoming fingerprintable based on latency introduced by hitting the bandwidth limit.

6.3.6. Cheatsheet

A quick-reference for the most important information during the Attack-Defence CTF.

Key	Value
Platform	https://platform.ad.ecsc2025.pl
Public Scoreboard	https://scoreboard.ad.ecsc2025.pl
Internal Scoreboard	http://10.42.251.2
Attack API	http://10.42.251.2:8080
Game Network	10.42.0.0/16
Team Network	10.42.X.0/24
Team Cloud Network	10.43.X.0/24
Vulnbox IP	10.42.X.2 (game network), 10.43.X.2 (team cloud network)
Gateway IP	10.42.X.254 (game network), 10.43.X.254 (team cloud network)
Flag submission	nc 10.42.251.2 1337

6.4. Rules for team composition

Each team is composed of a maximum of 10 (ten) and a minimum of 5 (five) players. Each participant must belong to one of the following groups:

- Senior, born from January 1st, 2000, to December 31st, 2004,
- Junior, born from January 1st, 2005, to December 31st, 2011.

People who do not fall in this age range are not allowed to participate in the ECSC 2025. Each team can contain up to 5 (five) senior players; no limitations are applied on the number of junior participants.

Reserve players and players substitutions are not allowed.

6.5. Communication

Invitation link to the server:

<https://discord.gg/whGW9GKeVh>

The link above does not expire, but in case of issues please contact the staff.

As in previous years, ECSC will use the Discord chat platform. Most communication on Day 1 (Jeopardy) and Day 2 (Attack-Defense) is expected to take place on the designated Discord server.

Note: Use of the provided Discord server is strongly recommended and is required for certain actions, such as filing formal complaints during the competition.

The Discord server setup, the authentication and the ticketing system is outlined in section 9.3. of this document.

Please note that this Handbook is published prior to the competition. As such, changes may be introduced.

6.5.1. Discord account

To use the Discord platform, it is required to have or create a Discord account. The platform can be accessed using one of the following methods:

- using a modern web browser through <https://discord.com/> (Discord accounts can also be created here).
- using a desktop client available at <https://discord.com/download>,
- or using a mobile client available at <https://discord.com/download>.

Important documents:

- <https://discord.com/privacy> – Discord Privacy Policy
- <https://discord.com/terms> – Discord's Terms of Service

6.5.2. Communication rules

Effective communication between players, coaches and the organization is crucial for smooth functioning of the competition and the overall event. To ensure clarity and efficiency, it is essential to distinguish between three main communication categories:

- competition-related communication, which pertains to the gameplay and technical aspects (e.g., game rules and format, technical issues affecting gameplay, formal complaints or disputes regarding scoring and rankings, reporting possible disruptive behaviour from other players within the game, strategic play, ...).
- event-related communication, which covers logistics, and any other non-competition matters (e.g., issues with meal schedules and dietary needs, health and safety concerns, general inquiries about event logistics).
- emergencies, which can cover health and safety issues, but also inappropriate behaviour, bullying, discrimination, illegal activities and every other scenario described in the Code of Conduct.

For all the competition-related communications, teams can always choose between one of the following options:

- A team captain, a main coach or a steering committee member opens a ticket to report a formal complaint, by using the dedicated ticket bot button in the Discord server. Every formal complaint must be initially reported through the dedicated ticket bot button to be considered by the jury. Other communication means are not considered valid.
- A player or a coach opens a proper ticket on the Discord server explaining the issue (e.g., a challenge seems to be not responding, or they need a clarification about the rules), by clicking on the corresponding ticket bot button in the Discord server.
- The team captain requests a meeting with a team coach or vice versa, by writing a message in the dedicated channel on the Discord server, pinging the role watchdog-manager.
 - o Meetings must be performed in one of the predefined spots (which will be clearly indicated in the game arena); after a meeting is requested, a watchdog

- handles the meeting request and accompanies the player to the predefined spot.
- The meeting must always be supervised by at least one watchdog and in English.
- Communication must be limited to non-technical topics. Some examples are:
 - Suggesting priorities
 - Reporting challenge status (but not technical details or hints)
 - Game strategy
 - Raise complaints about unfair behaviour from other teams
- Each team can request up to 8 meetings during each competition day.
- The maximum duration of a meeting is 3 minutes.
- The team captain speaks directly with a watchdog inside the game arena; in this case, it will be the watchdog's job to decide whether the request should be handled via a ticket through the Discord server or if it needs to be addressed urgently and immediately. Watchdogs can report urgent or serious issues directly to watchdog managers and/or organizers. Particularly serious situations will be handled case by case.

For all the event-related communications, people can always choose between one of the following options:

- A participant opens a proper ticket on the Discord server explaining the issue (e.g., a logistical issue in the venue), by clicking on the corresponding ticket bot button in the Discord server.
- The team captain writes a message on the coaches - captains Discord channel (#captain-coaches-comms), explaining what they need (e.g., they would like some more snacks).
- A player speaks directly with a watchdog inside the game arena; in this case, it will be the watchdog's job to decide whether the request should be handled via the Discord channel or if it needs to be addressed urgently and immediately (e.g. a player is not feeling well and needs medical assistance).

As regards emergencies, people should always feel free to communicate any serious issue in whatever way may work best for them. Some examples could be:

- Speak with a watchdog in the game arena or someone from the staff around the venue.

- Open an emergency ticket on the Discord server.
- Directly write on the emergencies dedicated Discord channel
- Generally, follow the communication guidelines described in the Code of Conduct.

6.6. Technical and human behaviour

Be nice to each other. ECSC is a competitive event, but also an opportunity for teams to learn from each other and have fun. Players should help foster that environment.

6.6.1. Fair sportsmanship rules

It is vital that everyone enjoys the event and leaves with a satisfying experience. Therefore, the **following are disallowed**, up to the penalty of a ban (as decided by the Jury), beyond those defined by common law (and common sense):

- Any communication about the CTF challenges (e.g. sharing flags, challenge details or solutions) with people outside of the player's own team (understood as players+captain) during the competition itself.
- Giving or accepting assistance from anyone outside the own team and organizers for issues related to the competition is strictly prohibited.
- Interfering with other teams' efforts, e.g. by DoSing, vandalizing public resources (like Wikipedia articles) related to a challenge solution, or similar actions.
- Flag hoarding — restraining from submitting flags after solves, and then submitting them at once near the end of the competition.
- Gathering and / or taking advantage of insider information relating to the competition is prohibited.
- Attacking the infrastructure or other teams is strictly prohibited. All of the targets will be hosted under challenge specific domain, unless otherwise specified in the challenge description.
- Attempting to elicit unintended behavior in any devices or services not designated as challenges for the competition running in the game network is strictly forbidden.
- Physically interacting or digitally tampering with the physical infrastructure provided by the organizers without permission is prohibited.
- Any action with the effect of creating excessive load for the contest or team infrastructure is prohibited, even if such actions are in the interest of the competition.
- Any action with the effect of intentionally making a service unavailable or non-functioning when interacted with by players or the checkers is prohibited.
- The deployment of fake flags is prohibited.

Any unfair behaviour with respect to the competition or the other players is forbidden, even if not explicitly described in the rules above; the organizers, the jury and any other relevant authority reserve the right to evaluate each case independently.

When in doubt, please ask (file a Discord ticket).

If you encounter any infrastructure or platform issues, please report them via a ticket on the Discord server and refrain from disclosing them publicly.

6.6.2. 0-day Policy

The organizers may disclose any vulnerabilities (including 0-days) used during the competition to the relevant upstream vendor, but they will put their best effort to credit the original finder and coordinate the disclosure process. Note that none of the challenges require using 0-day class vulnerabilities against third-party code.

6.7. Data recording and retention

Please note that various activities, including Discord messages across all channels, competition network traffic, Jeopardy platform usage, and entire Attack-Defence network traffic are **recorded and logged**. It will be accessed in case of a suspected rule violation.

Recorded data will be deleted within 30 days after the conclusion of ECSC2025, except for data fragments that may need to be retained for ongoing investigations, if any. Such data will be removed when no longer needed.

6.8. Allowed/necessary tools and hardware equipment

6.8.1. Software equipment

Every challenge proposed during both competition days will be solvable by using just open source or freely available software. Participants are free to use any software tool they want, including commercial ones, but the organizers will try to ensure they will not give any significant advantage to discourage their use.

6.8.2. Hardware equipment

Players are allowed to bring a basic hardware equipment setup: one laptop each, mouses, keyboards, power/ethernet/data cables, adapters, external drives, phones, headphones/headsets. The team is allowed to bring up to:

- 1 additional computer (max 300 W total output power as indicated on the charger, e.g., laptop, mini-PC, Raspberry Pi).
- 1 externally powered monitor, in total.
- 10 self-contained monitors (without external power source, e.g., USB monitors) or tablets in total.
- Recommended: 1 additional computer for hardware challenges (you will get separate instructions before the competition).

No power extenders are allowed except for the ones provided by the organizers.

Cloud resources are not considered “hardware equipment”; therefore, there is no limitation on them. If something is not mentioned in the list above, then teams must request it. This applies for electronics as well as bulky objects which may annoy other teams in the arena. If teams are unsure about something, then it is recommended to request it.

Each team is required to bring their own ethernet cables to connect their devices to the access switch they have on their table. Teams must also make sure they have all the adapters they need (e.g., in case no ethernet port is present on a laptop). For the hardware challenge you will need two unoccupied USB A and/or USB C ports. An externally powered USB hub is also acceptable.

Each team is required to bring all the power adapters they need. Each table has at least 12 sockets available (type E compatible).

The hardware equipment necessary to solve the hardware challenges is provided by the organizers; **players are not allowed to use any additional tool** to solve the hardware challenges apart from their laptops and phones or tablets.

Additional equipment must be requested in advance for the approval of the jury.

In case of players’ hardware failure, teams can request to substitute a device. The request must be reported to a watchdog, which can either accept it or raise it to the jury, depending on how difficult it is to determine the best solution. If the request is accepted, the team coaches can bring a new device to the players and bring the old device outside of the arena.

6.9. Penalties and complaints

Individuals or teams breaking one or more rules can receive a warning or a penalty, depending on the seriousness of the situation. Possible penalties apply to rule infringements both during the final event (from the time a team arrives until they depart) and online, on the Discord platform, at any time.

Penalties will be decided by the jury in accordance with the Steering Committee, in the form of:

- Time or point penalties for the team in one or both the competition days, or in the aggregated scoreboard.
- Lock on one or more challenges if the team is performing disruptive actions towards them (e.g. physical tampering on the hardware badge).
- Temporary or permanent exclusion of one or more team members from the competition.
- In extreme cases, complete disqualification of the team from one or both competition days.

7. Scoring system

7.1. Jeopardy scoring

The Jeopardy category will use *dynamic scoring*. Each challenge is worth 500 points at the beginning of the competition, regardless of category or difficulty. The challenge's worth decreases with more solves according to the following formula (Python):

```
int(500 * (30 / (29 + max(s, 1))) ** 3)
```

where **s** is the number of solves. The total number of points of each team is the sum of the *current* challenge worth (points) for all challenges solved by a given team. Specifically, it is NOT the sum of points the challenge had at the time of the solve — each team receives the exact same amount of points for a given challenge.

Example point values:

Challenge solve count	Points
0	500
1	500
5	343
10	227
15	158
25	85
33	56

"First bloods" (the first solution to a challenge) will be recognized and celebrated, but will not award additional points beyond eternal glory.

7.2. Attack-Defence scoring

In Jeopardy CTFs, dynamic scoring is used to infer the difficulty of a challenge based on the number of teams that can solve it. This scoring formula applies the same concept to A/D.

In effect, each round is treated as a Jeopardy CTF with the following *challenges*:

- For each flag you capture, you receive **ATK points** based on the number of teams that capture that flag.
- For each service and each flag store, you receive **DEF points** for each actively exploiting team that did not capture your flag, proportional to the number of teams whose flags that team did capture.

Additionally, you gain a **fixed amount of SLA points** for each deployed flag that is still *valid* (submittable for points) and retrievable from the service, as long as the checker status is SUCCESS or RECOVERING.

The total score is the sum of the **ATK**, **DEF** and **SLA** components.

The checker returns one of the following results for each service:

- SUCCESS if all flags could be successfully deployed and retrieved, and functionality checks were successful.
- RECOVERING if all checks for the current round succeed, but flags from the past 4 rounds are missing.
- MUMBLE if any functionality checks for the current round failed.
- OFFLINE if the checker failed to establish a connection to the service.
- INTERNAL_ERROR if an internal error occurred. **Please notify us with context in a ticket.**

The following is an edited snippet which calculates the final scoreboard from a series of rounds.

An *approximate* implementation may be evaluated against real CTF data using our simulator (<https://github.com/attacking-lab/scoring-playground>).

However, the following sections provide the most accurate description of the actual implementation. Please reference it in questions about scoring behavior. We try to include real code snippets from the game engine where possible to help with understanding.

Dynamic scaling is applied to each *challenge* by using the ECSC 2025 Jeopardy formula. We use a lower base value to account for the fact that the A/D CTF will have more dynamically weighted *challenges*, as we do not want scores to become too large.

```
def jeopardy(teams: int, base: int = 10):  
    return int(base * (30 / (29 + max(teams, 1))) ** 3)
```

The value of a challenge is close to base when the number of solving teams is low and close to zero when the number of solving teams is high. In this case, the value will drop down to 8% of base at 40 teams.

To distribute **ATK points**, the gameserver increases the number of submissions of all valid flags, based on the number of flags submitted each round, and updates their value accordingly. The value of a flag is calculated dynamically based on the number of teams who were able to capture it.

```
def attack(num_submissions: int):  
    return jeopardy(num_submissions)
```

The scores of teams who have captured flags previously are updated to reflect the decreased value of those flags by new submissions.

Since an attacker can exploit at most all active teams, which are neither themselves nor NOP (`max_victims`), the maximum gain from an exploit is `base * max_victims`.

To calculate **DEF points**, the gameserver updates the amount of captures of every flag which is still valid. For each team, the points gained from defending against a specific attacker are calculated dynamically based on the number of teams that were (or were not) exploited by them in that flag store in that round. This is meant to reflect that some exploits may be much more difficult to defend against than others and rewards teams that can construct solid defenses.

This value is scaled by the number of active teams (excluding the attacking team and NOP), and divided by the number of attackers for that flag store. Defense points are only awarded for *active* attackers, i.e., those teams that submit at least one flag from that flag store and round. If no teams are exploited, no teams receive DEF points.

```
def defense(max_victims: int, num_victims: int,
```

```
        num_attackers: int, exploited: bool):
    if exploited or num_victims == 0:
        return 0
    return jeopardy(max_victims - num_victims) * max_victims /
num_attackers
```

The defense points are scaled so that an ideal patch that blocks all active attackers does not gain more points than the attacker would from the exploit itself. When a patch blocks all attackers, it gains `defense(..) * num_attackers` or `jeopardy(max_victims - num_victims) * max_victims / num_attackers * num_attackers` points, which is at most `base * max_victims` (the same as for attack points but significantly harder to achieve).

To ensure that deleting flags in your own service is not a viable strategy for earning DEF points, we award the defense points for a flag spread across all rounds for which this flag must be retained. If a flag is unavailable in a specific round, no defense points are awarded for that flag in that round. Intuitively, this reflects the idea that defense points should be gained for successful defending; if no flags are at risk, no reward is earned.

```
def defense_scaled(live_tick: int, flag_tick: int,
                  checker_status: dict[int, str], ...):
    pts = 0
    max_tick = max(live_tick + 1, flag_tick + flag_rounds_valid)
    for tick in range(flag_tick, max_tick):
        if checker_status[tick] not in {"SUCCESS", "RECOVERING"}:
            continue
        if flag_avail_in[tick][flag_tick, team, service, flagstore]:
            pts += defense(...) / flag_rounds_valid
    return pts
```

Since defense points are recalculated for rounds in which still-valid flags were deployed, `max_tick` eventually reaches `flag_tick + flag_rounds_valid`.

At the end of the game, some flags need to be retained for fewer rounds. This means that protecting these flags earns proportionally fewer points over time, as there was also less time for other teams to capture them. However, the total number of flags you need to protect (and thus the defense points that can be earned in each round) does not change at the end of the game.

Additional ATK points are awarded to each attacking team based on the DEF points that other teams earn from defending against their attacks (more if the exploit is harder to defend against).

This prevents scenarios where defenders gain more DEF points than the attacker can gain ATK points simply because the attacking team started attacking (in other words, failed exploit attempts do not affect the score difference between teams).

To determine **SLA points**, the gameserver calculates the ratio between the number of valid flags retrievable from a service and the number of rounds a flag is valid for.

```
def sla(checker_status: str, flags_avail: int, base: int = 10):
    if checker_status == "SUCCESS":
        return base
    elif checker_status == "RECOVERING":
        return base * flags_avail / flag_rounds_valid
    else:
        return 0
```

This means that at the start of the CTF, SLA points ramp up from zero to base over the first five rounds, as the validity period is five rounds long.

- Since the capture count of each stored flag determines its worth, attackers are rewarded based on how difficult it is to exploit each specific team.
- The same goes for defense; a patch is rewarded based on the number of other teams that could not defend against the exploiting team. If a vulnerability is harder to patch or a specific exploit is harder to defend against, successfully doing so earns more defense points.
- Not attacking a team effectively gives that team defense points. Thus, there is an additional incentive to attack as many teams as possible beyond attack points. Teams will need to decide if the points gained from not attacking a team offset the expected loss of having the exploit stolen from the attack traffic.
- The maximum points gained from defending a flag store are never more than an exploiting team stands to gain.

After the Jeopardy and A/D CTF, teams' scores in these two categories are combined to yield a single final scoreboard, which should fairly represent the skill demonstrated in both contests.

Our goal in using the same dynamic scaling as the Jeopardy CTF is that the final team scores of both scoreboards approximate player skill with points in the same way, and thus allow for fairer merging.

Since defense points are awarded for not being attacked, and NOP team is never awarded defense points, NOP team score represents a team which only managed to keep their services

up, without exploiting anyone or defending against any exploits (by not receiving DEF points for any of them).

To avoid SLA points from inflating `ad_score` before merging and thus devaluing Jeopardy challenges, we subtract the NOP team score in the final scoreboard.

`ad_score = max(0, ad_score_unadj - nop_score)`

Q: Why is our team *losing* defense / attack points?

A: Teams may appear to *lose* defense or attack points when the value of the attacks they defended against or the flags they submitted decreases. This calculation is retroactive, as flags may be submitted up to 4 rounds *after* the round in which they are deployed.

Q: Why are the defense points not zero in a round our service status is neither SUCCESS or RECOVERING?

A: Most likely, a team was attacking your service before it went down and submitted (at least some of) those flags in the round before it went down. These flags are only considered in the next round, and you are then awarded defense points for defending against this exploit from the previous round retroactively. Crucially, you do not gain defense points for any flag stores not retrievable in the round in which your service was down.

Q: Why are the defense points zero in the first round after our service is available if we didn't get exploited?

A: The defense points may be zero in this round because no other team is attacking the flag store yet. Attackers typically rely on [attack info](#), which is only released in the subsequent round. If your service was unavailable for multiple rounds before this one, then no team will have attack info to attack the service with in the first round it is available.

7.3. Aggregated scoring

The aggregated scoring is computed by combining teams' scores and positions in the two competition days as described by the following formula; for each team:

$$\text{aggregated_score} = \text{jeopardy_score} + \text{ad_normalized_score}$$

where `jeopardy_score` is the team's score at the end of the jeopardy competition, while `ad_normalized_score` is the team's score at the end of the attack/defense competition, normalized on the same scale as the jeopardy one as follows:

$$\text{ad_normalized_score} = \text{ad_score} * (\text{jeopardy_winner_score} / \text{ad_winner_score})$$

where `ad_score` is the team's score at the end of the attack-defense competition, while `jeopardy_winner_score` and `ad_winner_score` are the scores of the teams who got first place during, respectively, the jeopardy and attack-defense competitions, considering only official teams.

After the score for each team is computed, two separate scoreboards will be created for official and guest teams.

8. Challenge categories and distribution

The ECSC 2025 challenge categories will be:

- **web** – vulnerabilities in web (internet) applications,
- **re** – reverse engineering,
- **hardware** – vulnerabilities in hardware, protocols, and embedded software,
- **crypto** – cryptography,
- **pwn** – vulnerabilities in applications and code,
- **forensic** – digital forensics,
- **stegano** – steganography,
- **misc** – other challenges not fitting into the above categories.

The platform at <https://hack.cert.pl> hosts challenges from the Polish national qualifiers. You can use it as a reference and for examples of topics that may appear in each category.

There will be **30 challenges** in Jeopardy style CTF and **5 services** in Attack-Defence style CTF.

The Jeopardy challenges distribution is as follows:

- a maximum of 5 challenges in each of web, re, crypto, pwn, forensic and misc categories,
- a maximum of 3 challenges in each of hardware and stegano categories.

There will be at least 4 easy and at least 4 hard challenges. This tries to ensure that every team should be able to solve at least a few challenges, while most of the teams will not be able to solve all challenges within the competition timeframe.

Since it is difficult to accurately predict the actual difficulty of each challenge, we do not use predefined difficulty indicators. Instead, we use dynamic scoring to reflect the real difficulty level of each challenge.

9. Platforms and API documentation

9.1. Jeopardy platform

The Jeopardy platform will serve as the central hub for all participants during the competition. It provides access to the challenges and displays the live scoreboard, allowing players to track their progress throughout the first day.

During the competition the Jeopardy platform will be available at: <https://ecsc-board.pl/>

Teams are strongly encouraged to register on the platform during the Setup Day. The ECSC2025 platform uses a per-team account approach. This means that **only one account per team must be registered** (e.g. by the team's captain) and it should be shared by all players of the given team. Secure password distribution within the team is left as an exercise to the players.

The following subsections describe various subpages of the Jeopardy Platform.

9.1.1. /challenges

List of all available challenges, including selected information about the challenge, such as:

- Name of the challenge.
- Current worth in points.
- Status of the challenge.
- Category of the challenge.

9.1.2. /challenge/<challenge_name>

This page provides all details of the given challenge, including its description, current solve count and point value.

It also provides following functionality:

- Flag submission form (use it to score points).
- Uploading write-ups.

Note: Flag submission will be rate limited to 20 flags per minute per team. Submissions exceeding this limit will result in an error message.

9.1.3. /scoreboard

This page shows current team rankings and lists solved challenges for each team.

Scoreboard by default shows only official teams, but has a feature to show all teams, i.e. official teams and guest teams.

Unlike in previous years, the scoreboard will **not** be frozen at the end of Day 1 this year. This said, please be mindful that the scoreboard might not be final and that submitting writeups for solved challenges is required.

9.1.4. /activity

Shows history of all challenge solves.

9.1.5. /profile

This page allows teams to change their password.

9.2. Attack-Defence platform and APIs

The platform at <https://platform.ad.ecsc2025.pl> is used for distributing Wireguard configs, collecting player SSH keys, and giving teams control over their cloud-hosted vulnbox instance.

All players are registered ahead of time on the platform, WireGuard configs may be downloaded 1 hour before the CTF begins.

With their accounts, players may submit SSH keys, which will later be deployed onto the vulnbox, as well as download WireGuard configs for accessing the game network through their team-specific router.

We provide a documented API on the gameserver (at 10.42.251.2 on port 8080), which returns filterable game-related information for use by team infrastructure.

You may try the api live at <https://api.demo.ad.ecsc2025.pl>.

9.2.1. /api/v1/services : Fine-grained service info

This endpoint returns service information, by default for all services. Data for a specific service may be queried through the use of URL parameters.

Example API usage:

```
curl http://10.42.251.2:8080/api/v1/services
```

```
{
  # service id : service info
  "0": {
    "name": "fooserv",
    "flagstores": 2,
  },
  ..
}
```

```
curl http://10.42.251.2:8080/api/v1/services?service=fooserv
```

```
{
  "name": "fooserv",
  "flagstores": 2,
}
```

9.2.2. /api/v1/teams : Fine-grained team info

This endpoint returns team information, by default for all teams. Data for a specific team may be queried through the use of URL parameters.

Example API usage:

```
curl http://10.42.251.2:8080/api/v1/teams
```

```
{
  ..,
  # team id : team info
  "2": {
    "name": "Team Europe",
    "affiliation": "Team Europe",
    "logo": "/static/109381717838108471.png"
  },
  ..
}
```

```
curl http://10.42.251.2:8080/api/v1/teams?team=2
```

```
{
  "name": "Team Europe",
  "affiliation": "Team Europe",
  "logo": "/static/109381717838108471.png"
}
```

9.2.3. /api/v1/score : Fine-grained scoring info

This endpoint returns scoring related information, by default for the current round. Data for a specific round, team and service may be queried through the use of URL parameters.

Example API usage:

```
curl http://10.42.251.2:8080/api/v1/score
```

```
{ # for current round id
  # team id : team info
  "12": {
    # service name : service info
    "fooserv": {
      "checker": "SUCCESS",
      "total": 632.7,
      "components": {
        "attack": 432.7,
        "defense": 0,
        "sla": 200.0
      },
      "flags_gained": 43,
      "flags_lost": 0
    },
    ..
  },
  ..
}
```

```
curl http://10.42.251.2:8080/api/v1/score?team=12&service=fooserv
```

```
{
  "checker": "SUCCESS",
  "total": 632.7,
  "components": {
    "attack": 432.7,
    "defense": 0,
    "sla": 200.0
  },
  "flags_gained": 43,
```

```
    "flags_lost": 0
}
```

9.2.4. /api/v1/attack_info : Fine-grained attack info

This endpoint returns *attack info* for services, by default for the current round. Data for a specific round, team, service or flagstore may be queried through the use of URL parameters.

Example API usage:

```
curl http://10.42.251.2:8080/api/v1/attack_info
```

```
{ # for current round id
  # team id : team info
  "12": {
    # service name : service info
    "fooserv": {
      # flagstore id : attack info
      "0": "target is 10cd9l7rt3",
      "1": null
    },
    ..
  },
  ..
}
```

```
curl http://10.42.251.2:8080/api/v1/attack_info?service=fooserv
```

```
{
  "12": {
    "0": "target is 10cd9l7rt3",
    "1": null
  },
  ..
}
```

9.2.5. /api/v1/current_round : current round time and id

This endpoint returns the current round id and start time as an ISO-8601 UTC timestamp with second precision. If the game has not started, this endpoint will return round 0.

Example API usage:

```
curl http://10.42.251.2:8080/api/v1/current_round
```

```
{
```

```
"round": 4,  
"time": "2025-09-15T13:58:21"  
}
```

9.2.6. /api/v1/next_round : next round time and id

This endpoint waits for the current round to complete before returning the new round start time as an ISO-8601 UTC timestamp with second precision. The connection may timeout if the new round has not started after two rounds worth of time.

Example API usage:

```
curl http://10.42.251.2:8080/api/v1/next_round
```

```
{  
  "round": 5,  
  "time": "2025-09-15T13:59:00"  
}
```

9.2.7. /api/faustctf2024/teams.json : Attack Info (FaustCTF 2024)

This endpoint returns attack info in the FaustCTF 2024 / teams.json format.

Example API usage:

```
curl http://10.42.251.2:8080/api/faustctf2024/teams.json
```

```
{  
  "teams": [  
    # team ids  
    123, 456, 789,  
    ..  
  ],  
  "flag_ids": {  
    # service name : service info  
    "service1": {  
      # team id : attack infos for validity period  
      "123": ["abc123", "def456"],  
      "124": ["xxx", "yyy"],  
      ..  
    },  
    ..  
  }  
}
```

9.2.8. /api/saarctf2024/attack.json : Attack Info (SaarCTF 2024)

This endpoint returns attack info in the SaarCTF 2024 /attack.json format.

Example API usage:

```
curl http://10.42.251.2:8080/api/saarctf2024/attack.json
```

```
{
  "teams": [
    # team infos
    {
      "id": 1,
      "name": "NOP",
      "ip": "10.42.1.2"
    },
    ..
  ],
  "flag_ids": {
    # service name : service info
    "fooserv": {
      # team vulnbox ip : team info
      "10.42.1.2": {
        # round id : attack info
        "123": ["info_flag1", "info_flag2"]
        ..
      },
      ..
    },
    "barserv": {
      "10.42.1.2": {
        "123": "info_single"
      },
      ..
    }
  }
}
```

The API returns attack info generated *at the start* of the round specified by the request. Scoring data returned by the api is the state of team scores *at the start* of the round specified in the request.

Note: Round schedule drift
Even though <i>successful</i> rounds are guaranteed to stay aligned with the round interval of 60 seconds , <i>single rounds may be cancelled in rare cases</i> , such as when the gameserver needs

to be restarted to address an infrastructure issue. To keep players in sync with the round schedule despite this albeit rare possibility, please use the `/api/v1/next_round` endpoint.

9.2.9. Parameters

The first played round of the CTF has the id **1**.

Player team ids start at **2**, since id **1** is reserved for the NOP Team.

Service ids are indexed starting at **1**.

Flagstore ids are indexed starting at **0**.

9.2.10. Scoreboard

Any APIs made available through the scoreboard host at `10.42.251.2` on port `80` exist solely for enabling the client-side functionality of the scoreboard. No guarantees are made for the availability or contents of these APIs.

9.2.11. Contact, disclosure, bug bounty

Please send information about vulnerabilities you find in the platform directly to the organizers via the following address: ecsc2025+bugs@attacking-lab.com.

If possible, encrypt your message with the key corresponding to the following fingerprint:

1B83 6E52 A23C 80B1 28CB 53BF 69BD EDEF 0AEC 0D6C

We want to reward players that find and report bugs in our platform.

If you report a bug before the competition and help us triage, we will do our best to reward that show of sportsmanship. Prizes TBD.

For support during the CTF, please open a ticket via the ticket bot in the ECSC 2025 Discord server.

In the unlikely case that Discord is unavailable, please send an email to the following address: ecsc2025+support@attacking-lab.com.

9.3. Discord server

9.3.1. Authentication system

By following the instructions in this section, users will be able to authenticate and gain access to proper Discord roles and channels.

The interface of the authentication system is provided in the form of a Discord channel **#auth-token** within the category of the same name.

This channel is accessible to anyone who has access to the Discord Server (ECSC2025).

1. The user should click the blue “🔑 Authenticate” button.

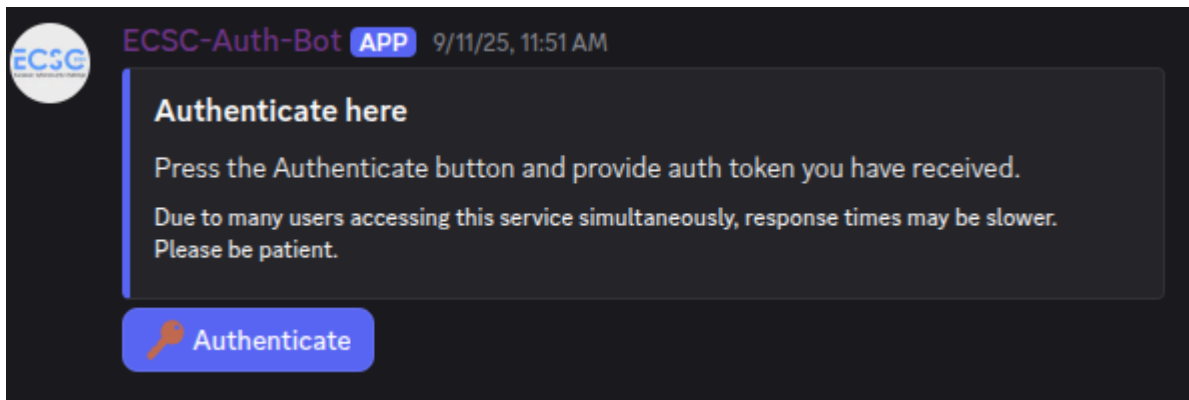


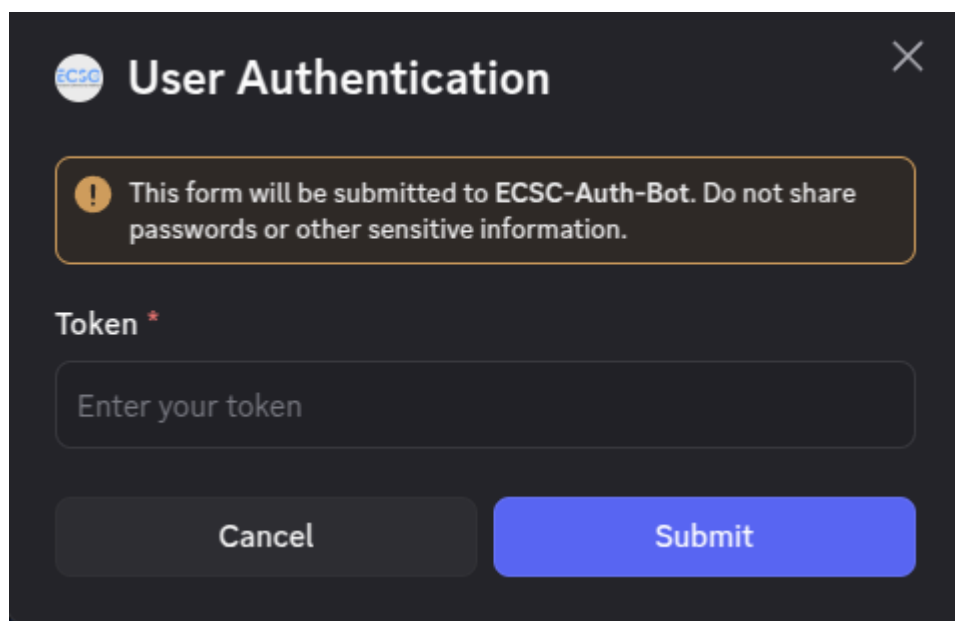
Figure - Token authentication message

2. The User Authentication form will appear and the user should enter the token they have received from their team's Point of Contact or from organizers.

An example token looks like this:

`e85e78efeb9d6c684e64d5c0cca2c90397149be266a3b9be4f2455e5d272cefb`

After entering the token in the form, press the blue “Submit” button.



The image shows a dark-themed modal window titled "User Authentication" with a close button (X) in the top right corner. Inside the modal, there is a warning box with an exclamation mark icon and the text: "This form will be submitted to ECSC-Auth-Bot. Do not share passwords or other sensitive information." Below this, the label "Token *" is followed by a text input field containing the placeholder text "Enter your token". At the bottom of the modal, there are two buttons: a grey "Cancel" button and a blue "Submit" button.

Figure - Token authentication modal form

3. On successful authentication a message similar to the one below will appear.

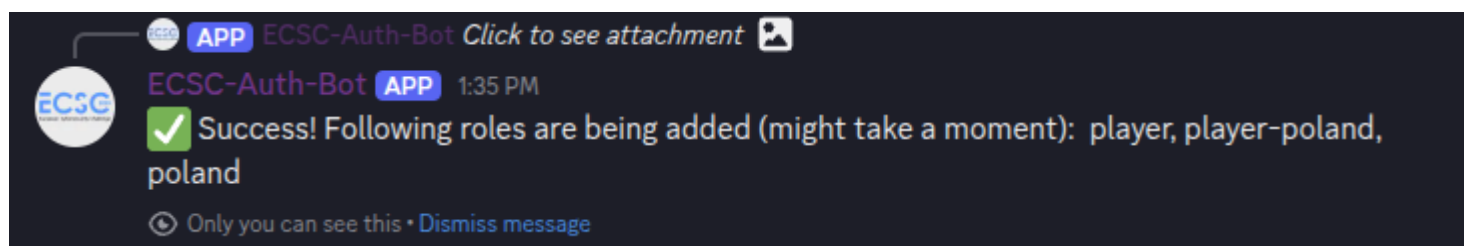


Figure - Success message

4. In case the token is invalid, please check for typos and check with your team's Point of Contact. If needed, please contact the organizers.

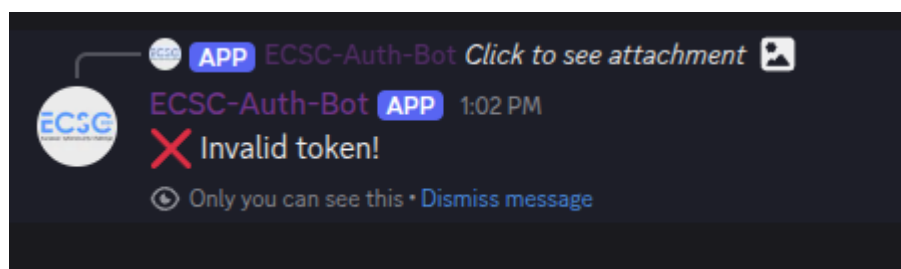


Figure - Failure message

In the case the bot is not responding or the “🔑 Authenticate” button is not working, please re-try in a couple of minutes. In case of further problems, please contact the organizers.

9.3.2. Ticketing system

The ticketing system enables participants to submit support requests to the organizers. It may be used for filing formal complaints, reporting issues related to the stability or functionality of challenges and services, and addressing general or platform-related matters.

For all technical docs visit:

- <https://docs.tickettool.xyz/>
- <https://tickettool.xyz/>

This section explains how to use the system from both the participants' and the support staff's perspectives.

9.3.3. Teams' manual

The ticketing system is implemented using the "Ticket Tool Discord Bot", which provides an efficient method for managing user requests during the competition.

This section of the manual describes the usage of the Ticket Tool bot for participants of the competition, including players, captains, and coaches.

The interface of the ticketing system is provided in the form of a Discord channel **#ticket** within the **📁 Tickets** category.

The interface is visible only to already authenticated users (see [How to authenticate](#)).

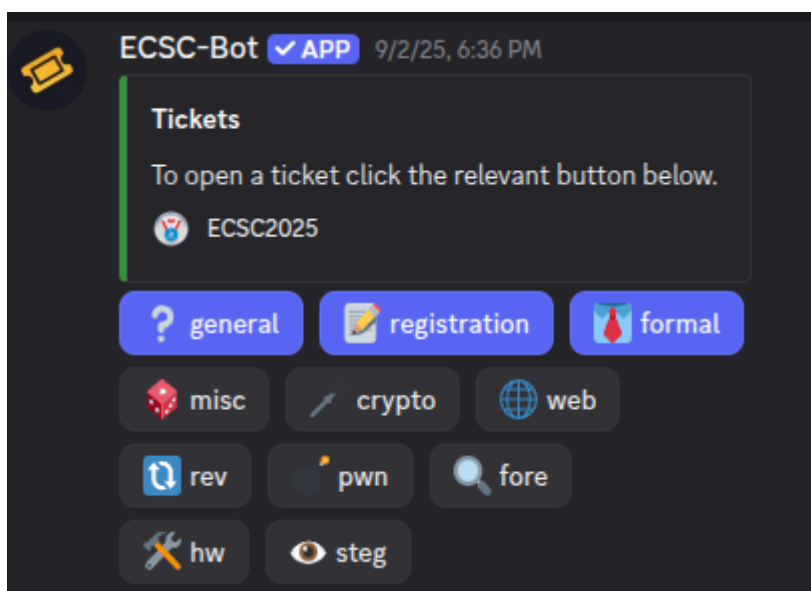


Figure - General Ticket Panel

The main ticket panel contains several buttons, each corresponding to a specific ticket category. The categories are briefly described below.

Tickets General category:

1. **general** - general questions, that may not fit any other category;
2. **registration** - account platform questions;
3. **formal** - related to complaints, policies questions.

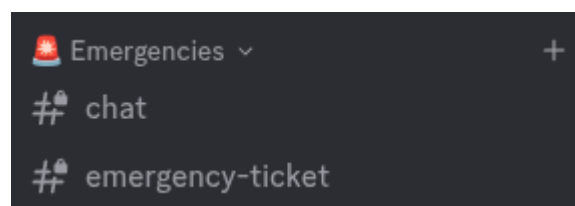
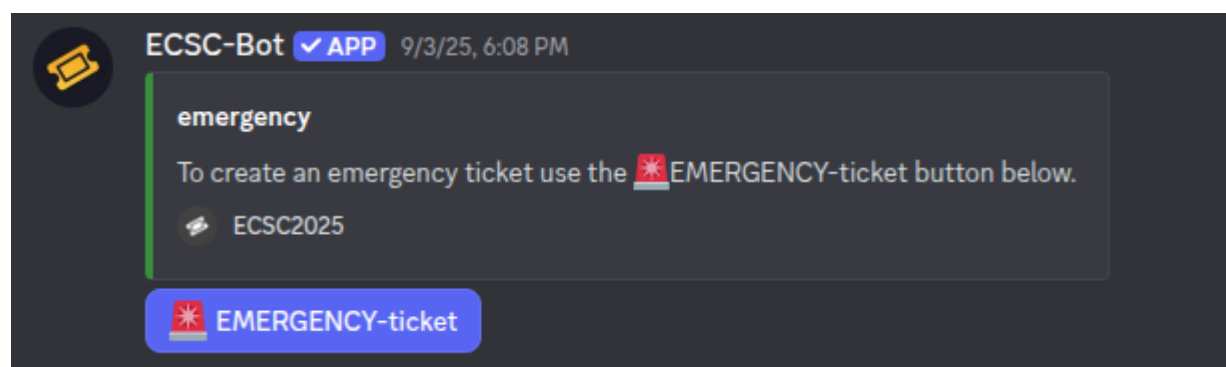
Tickets Competition category:

4. **misc** - “miscellaneous” challenges;
5. **crypto** - “cryptography” challenges;
6. **web** - “web security” challenges;
7. **rev** - “reverse engineering” challenges;
8. **pwn** - “binary exploitation” challenges;
9. **fore** - “forensics” challenges;
10. **hw** - “hardware” challenges;
11. **steg** - “steganography” challenges.

Emergencies category (special category):

12. **emergency** - the type of a ticket related to “emergency” questions — in case of urgent emergencies, please first contact staff on site!

Emergencies category is for emergency tickets only and its panel is located in **#emergency-ticket** channel:



9.3.4. Ticket creation and lifecycle

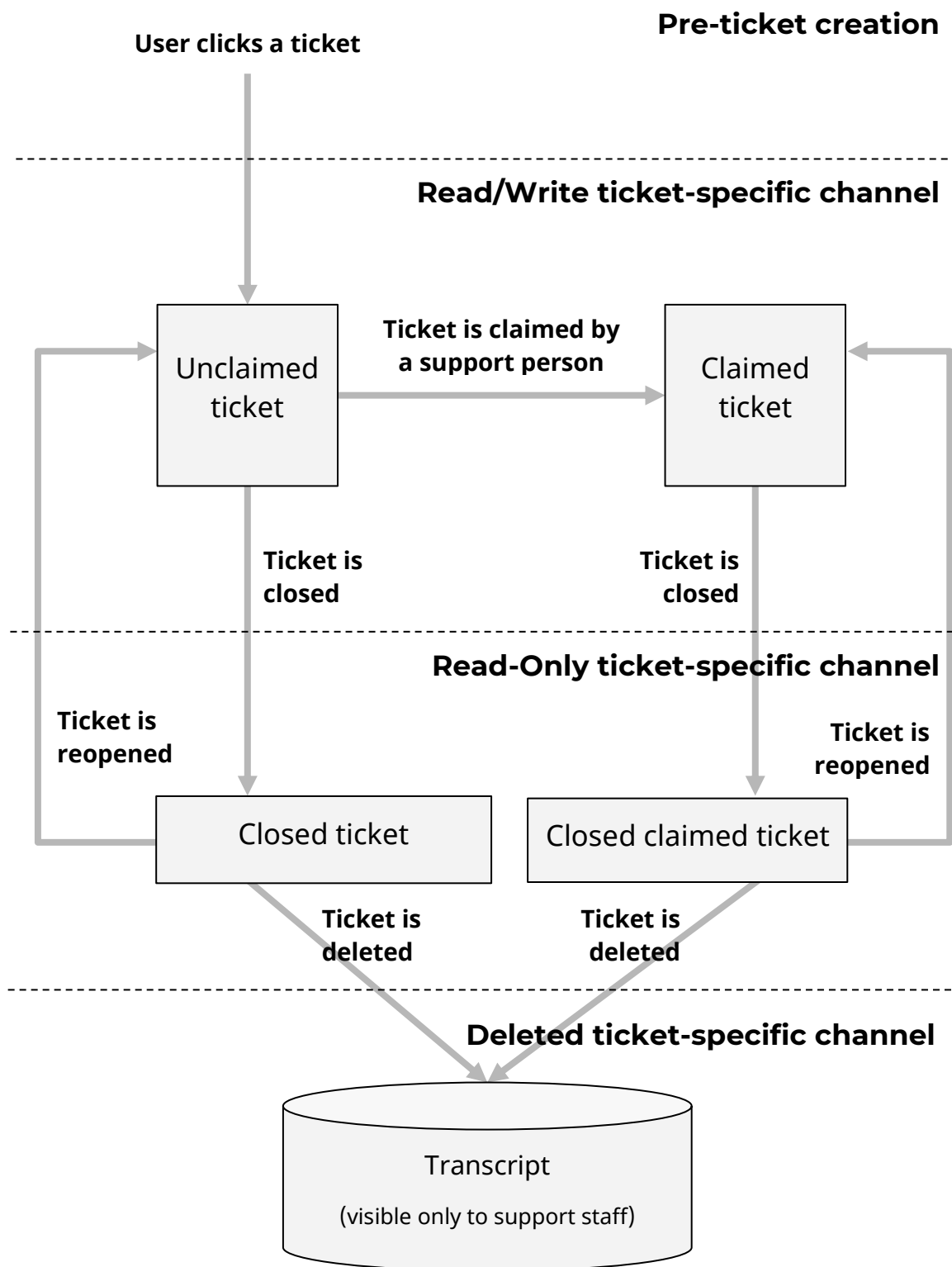


Figure - Ticket lifecycle

The following guideline is the same for any type of tickets (**emergency included**).

1. The user should click the button for the type of a ticket that they want to create/open, e.g., “ ? **general**” button. On a successful creation, this message should appear:

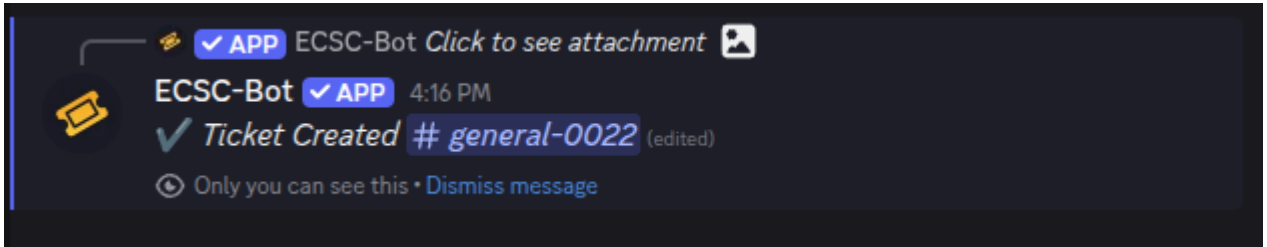


Figure - Successful ticket creation message

2. Newly created ticket — **#general-0022** in this case — is actually a Discord channel that the user will see in the **Tickets-General** category upon creation. This temporary ticket channel is only visible to the ticket owner which is the user in this case and the support team.
3. The ticket channel will contain a “Welcome message”. The user can write to the ticket channel before or after the support claims the ticket. The user should describe the problem in detail once the ticket channel is created.

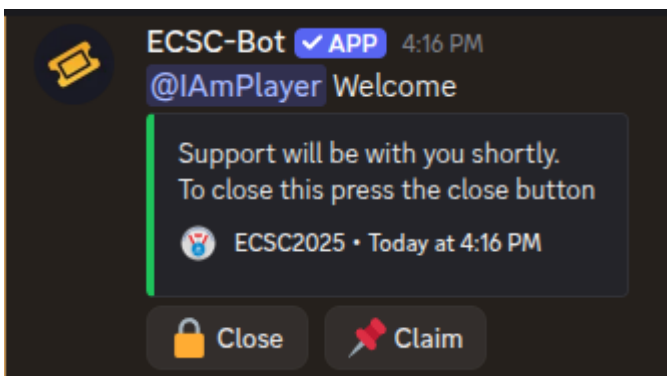


Figure - Ticket channel message

4. The user has an option to “  Close” the ticketing and so does the support team. The user might want to close the ticket in case its opening was not intentional (e.g. a misclick) or the issue was resolved. While “  Claim” option is only accessible by the support team. When the user clicks on the “Close” button, the close confirmation message should display.

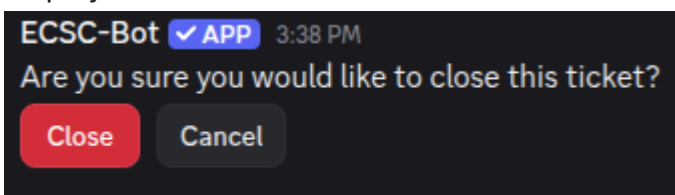


Figure - Confirm ticket closing message

- By clicking the red “Close” button the ticket will be closed and the channel will be renamed with “-closed” in the same category where the opened ticket channel was. The post closing message is below:

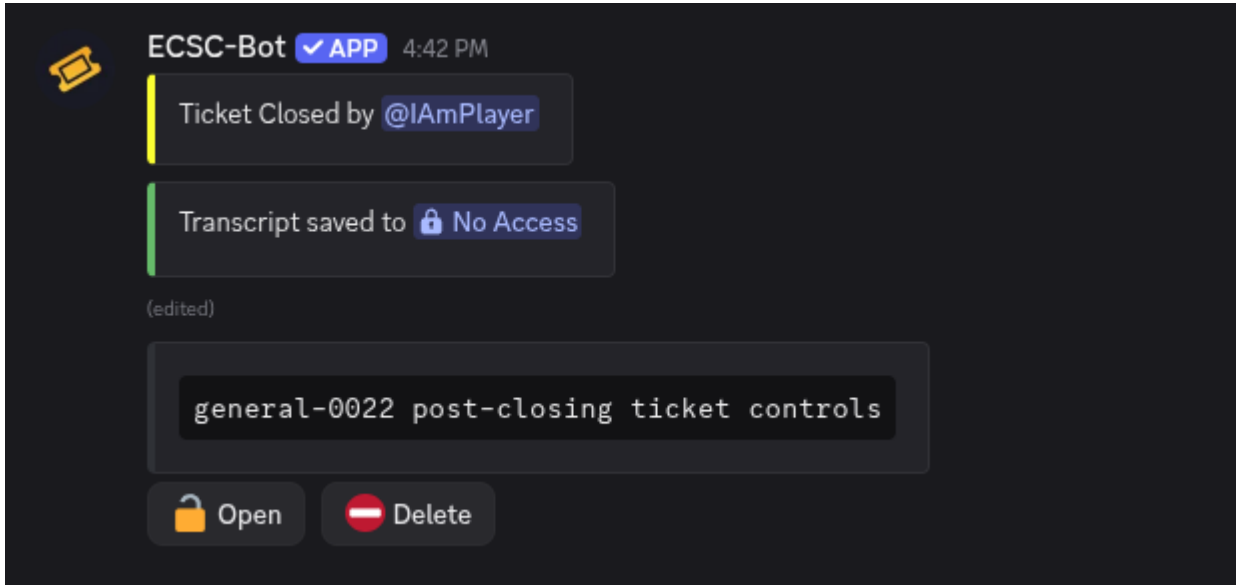


Figure - Post closing ticket message

- The user can also delete the ticket by clicking the “⊖ Delete” button. This means that the user can no longer see a deleted ticket, though the support team still has access to the recorded transcript of the conversation. This message should display on delete action:

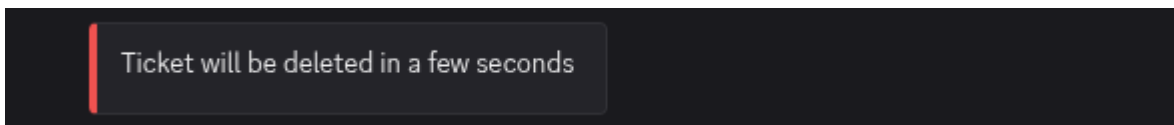


Figure - Delete ticket message

9.3.5. Staff Manual

This section of the manual describes the usage of the Ticket Tool bot for the **support team** of the competition, providing guidance on how to manage users' tickets.

When the user creates a ticket, it will be opened as a channel under either one of the categories, such as: “**Tickets-General**”, “**Tickets-Competition**” or “**Emergencies**”, depending on the type of a ticket.

Each ticket follows the naming convention “<type of ticket>_<number of ticket>” (e.g. *general-0001*).

9.3.6. Managing tickets

The following guideline is the same for any type of tickets (emergency included).

1. The support team member can click on the channel of the desired ticket that they want to resolve. Support member will see the two buttons: “🔒 Close” and “📌 Claim” with a Support message and mention of the user who created the ticket — e.g., “@IAmPlayer” in the screenshot below.

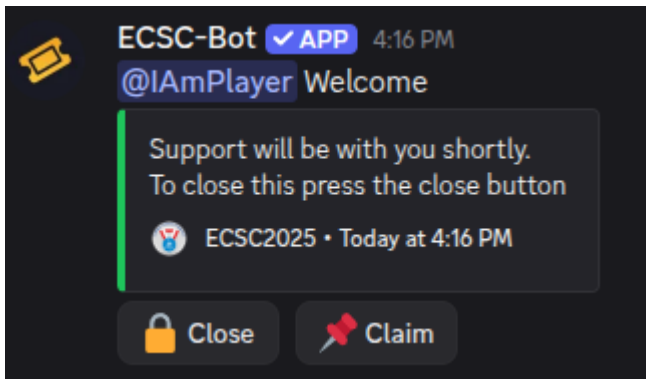


Figure - Opened ticket

2. The support team member can claim the ticket by clicking the “📌 Claim” button – once claimed the ticket is assigned to that support member, which means this member will be managing this ticket e.g., “@staff🦄”, though any other support members are still able to see that ticket and interact with it. Message that appears is below:

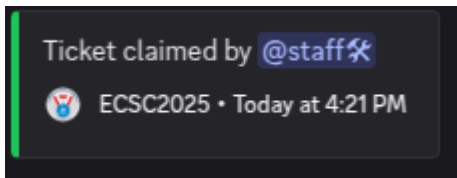


Figure - Claimed ticket message

3. The support team can close the ticket by clicking the “🔒 Close” button. The closed ticket channel will be renamed with “closed” being added to its name. The closing message is displayed as:

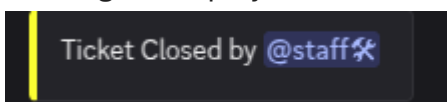


Figure - Closed ticket message

4. After closing the ticket the post-closing message will appear with two buttons: “🔒 Open” and “🗑 Delete”, where the player or support team can either re-open the closed ticket or delete the closed ticket. The support team can access the transcript of the closed/deleted ticket.

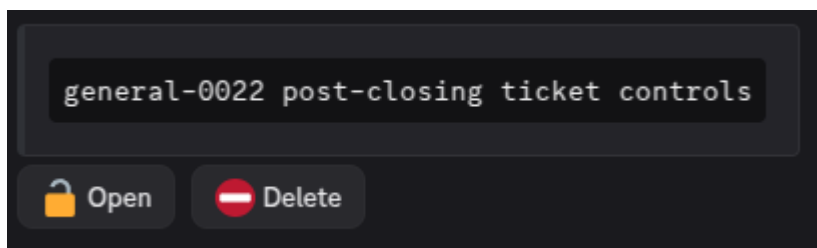


Figure - Post closing ticket message

- By clicking on the “🚫 Delete” button, the ticket will be deleted and the message will appear:

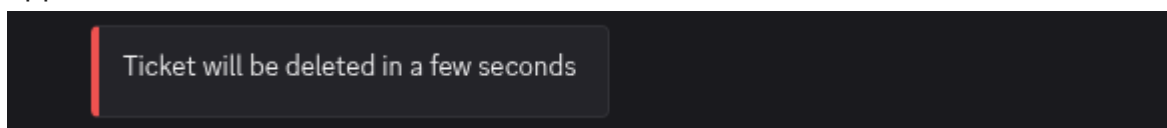


Figure - Delete ticket message

- The closed ticket can also be re-opened by clicking the “🔓 Open” button, this will re-create the ticket and the message will appear:

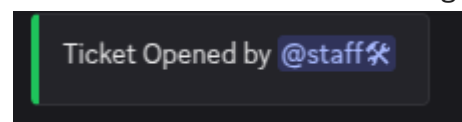


Figure - Re-open ticket message

9.3.7. Channels and Categories

This section describes the organization of the ECSC2025 Discord Server categories and its channels.

All channels are textual.

Notes:

- Various organizer roles (e.g. staff, Discord admins, ENISA) have read-write access to every channel. Other selected organizer roles might have broader access as well.
- The jury has at least read access on every channel.
- There are multiple announcement channels that will be used for various purposes – please monitor all of them.

The channels and categories are:

- Top channels without a category
 - o rules
- ECSC-Players
 - o 📢-players-announcement

- *Country (all country-specific channels)*
- Captains-Coaches
 - 📢-coaches-announcement
 - captains-coaches-comms
- Steering-Committee
 - 📢-sc-announcement
 - general
- Staff
 - general
- Watchdogs
 - general
- Angels
 - general
- Jury
 - general
- Authors (of challenges)
 - general
- ECSC-Public
 - 📢-announcements
 - social-feeds
 - general
 - memes
 - random
 - mvm
 - joins
- ECSC-Day1-Jeopardy
 - 📢-announcement-jeopardy
 - general
- ECSC-Day2-AD
 - 📢-announcement-ad
 - general
- Tickets, Tickets-General, Tickets-Jury, Tickets-Competition (see ticketing system section 9.3.3.)
 - transcript-logs
 - ticket
- Emergencies
 - general
 - emergency-ticket

- emergency-logs
- Auth-Token (see the ticketing system section 9.3.3.)
 - auth-token

9.3.8. Discord Roles

This section is describing all important Discord roles that exist on ECSC2025 Discord Server.

- player
- *player-country*
- country
- captain
- *captain-country*
- coach
- *coach-country*
- main-coach
- *main-coach-country*
- steering-committee
- jury
- watchdog
- watchdog-manager
- angel
- angel-manager
- ticket-manager
- staff
- admin
- ENISA
- author

Roles like: captain, coach, main-coach, *country*, ticket-manager can be mapped to multiple role sets – they are sort of role “tags” .

For more information, see Role sets section.

Please note that a Staff member might have multiple roles, including multiple extra case-specific roles if needed.

9.3.9. Role sets

This part is explaining the role sets for the roles in the way they are organized on ECSC2025 Discord Server.

Role set name	Discord roles
Captain	captain, player, player-country, country, captain-country
Coach	coach, coach-country, country
Main Coach	main-coach, coach, main-coach-country, country
Player	player, player-country, country
Watchdog Manager	watchdog-manager, watchdog
Angel Manager	angel-manager, angel