Harmonizing People, Processes, and Technology for Robust Cybersecurity

D6.1 Cyber Awareness Platform v1

Document Summary Information

Grant Agreement No	101158555	**Acronym**	CYberSynchrony
Full Title	Harmonizing People, Processes, and Technology for Robust Cybersecurity		
Start Date	01/06/2024	**Duration**	36 months
Project URL	https://www.cybersynchrony.eu/		
Deliverable	D6.1 Cyber Awareness Platform v1		
Work Package	WP6 – Cybersecurity Consciousness: Empowering Society Through Awareness (CYBERWISE and Cybersecurity Standardization and Compliance Reinforcement (CYBERGOPLUS)		
Contractual due date	31/12/2025	**Actual submission date**	22/12/2025
Type	OTHER (Report on the specs)	**Dissemination Level**	PU
Lead Beneficiary	MAG		
Responsible Author	Konstantinos Andrikos (MAG)		
Contributions from	Vasilis Keleris (MAG), Evangelos Michos (IANUS), Demetris Antoniou (IANUS), Eleni Keliri (eBOS), Hassan Yeganeh (eBOS), Laskarina Mavropoulou (MER), Andrea Purificato (MER), Leandros Maglaras (TRUST), Dimitris Koutras (FP), Vasileios Roussakis (KETYAK), George Karavokiros (BYTE)		

Funded by the European Union

ECCC

Funded by the European Union under Grant Agreement no. 101158555 and supported by the European Cybersecurity Competence Centre (ECCC). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Cybersecurity Competence Centre. Neither the European Union nor the granting authority can be held responsible for them.

Revision history

Version	Issue Date	Changes	Contributor(s)
0.1	10.10.2025	Initial Deliverable structure	MAG
0.2	22.10.2025	Input in Chapters 1, 4, 5	MAG, IANUS, TRUST
0.3	03.11.2025	Input in Chapter 3, 4	IANUS, eBOS
0.4	23.11.2025	Input in Chapters 2, 3, 5, 6	MAG, IANUS, MER, BYTE FP, KETYAK
0.5	03.12.2025	Final draft ready for internal review	All
0.6	15.12.2025	Peer review	Dimitris Koutras (FP), Laskarina Mavropoulou (MER)
0.7	17.12.2025	Revision of deliverable and submission for QA review	Konstantinos Andrikos (MAG)
0.8	19.12.2025	Final quality check	Eleni Keliri (eBOS)
0.9	22.12.2025	Addressing comments	Konstantinos Andrikos (MAG)
1.0	22.12.2025	Final version ready for submission	Violeta Vasileva (MAG)

Disclaimer

The content of the document herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the CYberSynchrony consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the CYberSynchrony consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the CYberSynchrony consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

© CYberSynchrony Consortium This Deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.

[1] According to CYberSynchrony's Quality Assurance Process for reports:

- 3 months before Deliverable's Due Date: Table of Contents ready to be reviewed by WP leader/ Technical Manager/ Quality Manager
- 2 months before Deliverable's Due Date: 50% should be complete. Review by the Quality Manager (& Technical Manager for software).
- 1 month before Deliverable's Due Date:100% should be complete. Review by the Quality Manager and Peer Reviewers.
- M: Sent to PC for Submission to the EC after addressing all comments by Quality Manager and Peer Reviewers

Table of Contents

1 Executive Summary 9
2 Introduction 10
2.1 Mapping CYberSynchrony Outputs 10
2.2 Linkage to other project outputs 11
2.3 Deliverable Structure 11
3 Design Principles for the Cyber Awareness Platform 12
3.1 Learning objectives 12
3.2 Platform Design 12
3.3 Accessibility, privacy and harmonization 13
3.3.1 Accessibility and Inclusivity 13
3.3.2 Privacy by Design and GDPR Compliance 14
3.3.3 Harmonization of Standards 14
4 Platform Architecture and Integration 15
4.1 Deployment architecture 15
4.2 Flow of information 16
4.3 Module features 16
4.3.1 CYBERRA – Cyber Threat Intelligence and Customized Risk Analysis 16
4.3.2 CYRESCUE - Cybersecurity Resilience and Incident Response Enhancement 17
4.3.3 CYBRITE - Proactive Cyber Risk Monitoring, and Incident Response 17
4.3.4 CROSS-CORE - Cross-Border Cyber Incident Cooperation & Response 17
4.3.5 CYBERWISE - Cybersecurity Education for All 18
4.3.6 CYBERGOPLUS - Cybersecurity Governance and Compliance 18
4.4 Components at a glance 18
4.5 Identity & Access Management with Keycloak 20
4.5.1 Centralized Authentication 20
4.5.2 Role-Based Access Control (RBAC) 21
4.5.3 Integration With CYberSynchrony Modules 21
4.5.4 Attribute-Based Personalization 22
4.5.5 Security and Privacy Considerations 23
4.6 Technology stack 23
4.7 Deployment models 23
4.7.1 Prerequisites 24
4.7.2 Node Preparation — 01-prepare-node.sh 24
4.7.3 Control Plane Initialization — 02-init-control-plane.sh 26
4.7.4 Calico Network Installation (CNI) — 03-calico.sh 28
4.7.5 Ingress Controller — 04-ingress-nginx.sh 29
4.7.6 cert-manager Core — 05-cert-manager-core.sh 30
4.7.7 GitLab Agent (KAS) — 06-install-gitlab-agent.sh 32
4.7.8 Linking the Cluster to GitLab.com 33
4.8 Execution Order (Summary) and Cluster Validation Plan 35
5 User/Cyber Awareness Content 37
5.1 Content pipeline overview 37
5.2 Training plan outline 38
5.2.1 Core Methodology and Structure 38
5.2.2 Training Plan Content Integration and Objectives 38
5.2.3 Measurement and Effectiveness 39
5.3 Real word social engineering 39
5.3.1 Introduction and Strategic Objectives 39

© CYberSynchorny, 2024

5.3.2 Methodology 40
5.3.3 Next steps and future work 42
5.4 Integration of gamified cybersecurity 43
5.4.1 Cybergate's Learning Methodology 43
5.4.2 Alignment with CYBERWISE Objectives 43
5.4.3 Technical Integration 44
5.5 Integration of cybersecurity policy toolkit 44
5.6 Sector-Specific Cybersecurity Harmonization 45
5.6.1 The Harmonization Model 45
5.6.2 Application of Harmonised Principles to Training 45
5.6.3 Technical Implementation of Harmonization 46
5.7 Integration of compliance monitoring 46
5.7.1 Role of the Cyber Awareness Platform in Compliance Monitoring 46
5.7.2 Data Flows and Integration Interfaces 47
5.7.3 Alignment with CYBERGOPLUS Compliance Domains 47
5.7.4 Contribution to Real-Time Oversight 47
5.7.5 Architectural Positioning 48
5.8 Training Module: Practical Software Vulnerability Assessment using SNYK 48
5.8.1 Introduction and Strategic Objectives 48
5.8.2 Core Learning Objectives and Expected Outcomes 48
5.8.3 Pedagogical Methodology and Content Delivery 49
5.8.4 Assessment Strategy and Implementation Planning 49
6 Conclusions 51
7 References 52
Annex I: Environment variable reference (per script) 53
Annex II: Sample Manifests & Snippets 54

List of Figures

Figure 1. Deployment Architecture 15
Figure 2. Single Sign-On via Keycloak 20
Figure 3. Platform's Client scopes 21
Figure 4. CYberSynchrony modules 22
Figure 5. Example of CYBRITE module environment 22
Figure 6. Sample of prompts with env overrides. 25
Figure 7. Sample kubeadm-config.yaml (generated). 27
Figure 8. Sample of quick verify in control plane. 27
Figure 9. Sample of Quick Verify in CNI. 29
Figure 10. Sample of Quick verify in Ingress Controller. 30
Figure 11. Example of ClusterIssuer (prod). 31
Figure 12. Quick verify of cert-manager. 32
Figure 13. Scheme outlining the procedure of linking the cluster to Gitlab.com. 34
Figure 14. Samples of "commit agent configuration", "minimal configuration", "GitOps" (optional), "Private Registry & ImagePullSecret" (optional), and "reference the workloads". 35

© CYberSynchorny, 2024

Figure 15. Gophish campaign design internal system 40
Figure 16. Gophish statistics about the emails sent during the campaign (sent, opened, clicked, submitted data, reported 40
Figure 17. Phishing Email from the 1st campaign 41
Figure 18. The loading webpage where the victim is asked to add their information 41
Figure 19. The victim is asked to add username and password, but this information is not captured in Gophish 42
Figure 20. Typical adversarial attack lifecycle 43

List of Tables

Table 1: Adherence to CYberSynchrony's GA 10
Table 2: Links to other Project Outputs/Work 11
Table 3. Possible issues and troubleshooting recommended actions. 36

© CYberSynchorny, 2024

Acronyms and abbreviations

Abbreviation / Term	Description
GA	Grant Agreement
WP	Work Package
PC	Project Coordinator
EC	European Commission
ECCC	European Cybersecurity Competence Centre
CYBERRA	Cyber Threat Intelligence and Customized Risk Analysis
CYBRITE	Proactive Cyber Risk Monitoring and Incident Response
CROSS-CORE	Cross-Border Cyber Incident Cooperation & Response
CYBERGOPLUS	Cybersecurity Governance and Compliance
CYBERWISE	Cybersecurity Consciousness: Empowering Society Through Awareness
CYRESCUE	Cybersecurity Resilience and Incident Response Enhancement
APT	Advanced Persistent Threat
KAS	GitLab Kubernetes Agent
SOC	Security Operations Centre
IoC	Indicator of Compromise
PII	Personally Identifiable Information
BYOD	Bring Your Own Device
NIS2	Directive on Measures for a High Common Level of Cybersecurity
GDPR	General Data Protection Regulation
CNI	Container Network Interface
DNS	Domain Name System
TLS	Transport Layer Security
ACME	Automated Certificate Management Environment
UI	User Interface
API	Application Programming Interface
JSON	JavaScript Object Notation
XML	Extensible Markup Language

© CYberSynchorny, 2024

IAM	Identity and Access Management
RBAC	Role-Based Access Control
SSO	Single Sign-On
DCO	Defensive Cyber Operations
OWASP	Open Web Application Security Project
QA	Quality Assurance
DevSecOps	Development, Security, and Operations
WCAG	Web Content Accessibility Guidelines
ARIA	Accessible Rich Internet Applications
REST	Representational State Transfer
CI/CD	Continuous Integration / Continuous Delivery
URL	Uniform Resource Locator
ID	Identifier
ML	Machine Learning
UEBA	User and Entity Behaviour Analytics
IDS	Intrusion Detection System
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
SIEM	Security Information and Event Management
XDR	Extended Detection and Response
HER	Electronic Health Record
CSPA	Cyber Security Posture Assessment
PTA	Purple Team Assessment
SBS	Security Breach Simulation
ASA	Attack Surface Assessment

© CYberSynchorny, 2024

HRM	Human Risk Management
ECSF	European Cybersecurity Skills Framework

© CYberSynchorny, 2024

1 Executive Summary

Deliverable D6.1, titled “Cyber Awareness Platform v1”, presents the design and initial implementation of the **CYberSynchrony Cyber Awareness Platform**. The platform represents the human-centric dimension of CYberSynchrony, complementing the technical modules (CYBERRA, CYRESCUE, CYBRITE, CROSS-CORE) with a dedicated environment that promotes cyber awareness, behavioral change, and compliance readiness across sectors.

The objective of D6.1 is to define the **architecture, design principles, and integration framework** for a unified learning and awareness platform that seamlessly connects with the broader CYberSynchrony ecosystem. The work focuses on building the Cyber Awareness Platform as part of the overarching CYberSynchrony interface, ensuring interoperability through **Keycloak-based identity and access management**, containerized deployment, and REST API communication with the other modules. This integration allows users to access personalized awareness content, training modules, phishing simulations, and compliance dashboards through a single secure entry point.

The design of the platform follows a **user-centric, scalable, and privacy-preserving approach**. It adopts principles of inclusiveness, modularity, and harmonization, aligning with the project’s goal of “Harmonizing People, Processes, and Technology for Robust Cybersecurity.” The platform architecture enables both organizational and individual users to engage in cyber awareness activities while maintaining alignment with EU cybersecurity directives such as NIS2 [1] and GDPR [2].

Version 1 establishes the foundational framework for the Cyber Awareness Platform, including:

- Definition of the platform’s learning objectives and design principles
- Initial system architecture and technology stack aligned with the D8.1 CYberSynchrony Integration Blueprint [3]
- Implementation of secure authentication and role-based access control using Keycloak [4]
- Integration pathway between the learning environment, gamified modules, and policy/compliance tools to be delivered in D6.2–D6.8.

The future deliverable, D6.5 “Cyber Awareness Platform v2”, will extend this work with fully operational LMS functionalities, gamified content, real-time compliance monitoring, and data-driven behavioral analytics. Collectively, these developments will empower end-users, public institutions, and enterprises to strengthen their cybersecurity culture and resilience through continuous learning and engagement.

© CYberSynchorny, 2024

2 Introduction

This chapter introduces the context, objectives, and positioning of the current deliverable within the overall CYberSynchrony project. It explains how D6.1 contributes to the project's mission of harmonizing people, processes, and technology for robust cybersecurity. The chapter provides an overview of how this deliverable aligns with the Grant Agreement outputs and Work Package 6 objectives, highlights its interconnections with other project activities, and presents the structure of the document. Together, these elements establish the foundation for understanding the purpose and scope of this deliverable within the broader CYberSynchrony ecosystem.

2.1 Mapping CYberSynchrony Outputs

This section presents the CYberSynchrony's Grant Agreement (GA) commitments, as extracted from the formal deliverable and task description, in respect to their outputs and work to be performed.

Table 1: Adherence to CYberSynchrony's GA

Component no	Component title	Respective Document Chapter(s)	Justification
DELIVERABLE			
D6.1	Cyber Awareness Platform v1	All	Defines the initial design, architecture, content framework and stack of the Cyber Awareness Platform
TASKS			
T6.1	CYberSynchrony Platform and Pervasive Cybersecurity Awareness Platform	Chapters 3, 4, 5	Design principles, architecture, content pipeline and operation
T6.2	Proactive Social Engineering Assessments	Chapter 5 (Section 5.3)	Planning, execution and analysis of the simulated phishing campaign
T6.3	CYberSynchrony training planning	Chapter 5 (Sections 5.1 and 5.2)	Content pipeline and initial training plan
T6.4	Immersive Gamified Cybersecurity	Chapter 5 (Sections 5.4 and 5.8)	Gamified learning integration and experiential training modules
T6.7	Real-time Compliance Oversight	Chapter 5 (Section 5.7) and Chapter 4 (Section 4.3.6)	Introduces compliance monitoring integration and CYBERGOPLUS governance features.

© CYberSynchorny, 2024

2.2 Linkage to other project outputs

Table 2: Links to other Project Outputs/Work

WP Number	Task Number	Deliverable Number related	Content
WP2	T2.1 - T2.4	D2.1, D2.2	Threat landscape and user requirements shaping the platform's learning and design
WP3	T3.3, T3.4	D3.1 - D3.4	Risk-analysis inputs supporting contextual awareness content
WP4	T4.1	D4.1, D4.2	Monitoring tools referenced in the architecture as data sources
WP6	T6.2–T6.5	D6.2, D6.3, D6.4	Gamified training elements, social-engineering assessments and training-planning insights; Policy-toolkit elements; Compliance-monitoring insights
WP8	T8.1	D8.1	Integration blueprint defining how CYBERWISE and CYBERGOPLUS are positioned in the overall system architecture.

2.3 Deliverable Structure

This document is organised into the following chapters:

- **Chapter 1: Executive Summary**
  Provides a high-level overview of the Cyber Awareness Platform, its objectives within CYberSynchrony, and the main outcomes of this first version.
- **Chapter 2: Introduction**
  Defines the purpose and scope of D6.1, maps the content to project commitments, and outlines the structure of the document.
- **Chapter 3: Design Principles for the Cyber Awareness Platform**
  Presents the core design principles, learning objectives, and key considerations for accessibility, privacy, and harmonization.
- **Chapter 4: Platform Architecture and Integration**
  Describes the deployment architecture, information flows, module features, identity and access management, technology stack, and deployment models.
- **Chapter 5: User/Cyber Awareness Content**
  Summarizes the content pipeline, training plan, social engineering assessment, gamified components, policy toolkit integration, and initial SNYK-based training module.
- **Chapter 6: Conclusions**
  Highlights the main achievements of this first platform version and outlines directions for further development.
- **Chapter 7: References**
  Lists the bibliographic and project sources cited in the deliverable.

© CYberSynchorny, 2024

3 Design Principles for the Cyber Awareness Platform

The Cyber Awareness Platform represents the human-centric layer of the CYberSynchrony ecosystem, complementing its technical cybersecurity capabilities with structured awareness, training, behavioural reinforcement and compliance enablement. Its design principles reflect the project's vision of harmonizing people, processes and technologies to improve resilience across sectors.

Informed by the multi-actor threat landscape and end-users needs identified via D2.1 [5], the platform focuses on addressing human-factor vulnerabilities through continuous, context-aware learning.

3.1 Learning objectives

The Cyber Awareness Platform aims to achieve the following learning objectives:

- **Strengthen foundational cyber hygiene**: Enable users to adopt safe digital practices, recognise social engineering attempts and avoid behaviours that elevate organisational risk, addressing common human-factor vulnerabilities highlighted in the threat analysis.
- **Increase awareness of sector-specific risks**: Ensure that users understand the threat patterns most relevant to their operational domain (e.g., healthcare data breaches, public sector APT targeting, academic IP theft), as detailed in the multi-sector analysis of D2.1 [5].
- **Improve recognition and reporting of security incidents**: Equip users with the ability to identify suspicious activity and follow appropriate reporting or escalation procedures, complementing the detection and response capabilities summarised in the module descriptions of this deliverable.
- **Support internalisation of organisational policies and compliance requirements**: Promote understanding of cybersecurity policies, acceptable use rules and regulatory obligations that govern user behaviour, reinforcing the compliance-related features described in CYBERGOPLUS and related components.
- **Foster preparedness for coordinated and cross-border cybersecurity response**: Introduce users to principles relevant to coordinated incident handling and secure information sharing, reflecting the collaboration mechanisms supported by the ecosystem's cross-border components.
- **Encourage sustained engagement through interactive and immersive learning**: Reinforce behavioural change through simulations, gamified exercises and scenario-based training, as outlined in the platform's training and gamification sections.

Together these learning objectives provide a structured foundation for resilient cybersecurity mindset across all user groups engaging with the Cyber Awareness Platform.

3.2 Platform Design

The design of the Cyber Awareness Platform follows a modular and user-centric approach that aligns with the overall architecture and operational model of the CYberSynchrony ecosystem. It draws on the system-level interfaces, identity management framework, and module interconnections described in the technical sections of this deliverable, which detail how services such as monitoring, threat intelligence, compliance and human-risk analytics are integrated into a cohesive environment.

Here are the platform's design principles:

- **Modularity and Interoperability**: The platform operates as a distinct yet fully interoperable component of CYberSynchrony. Its awareness, behavioural analytics and compliance features integrate seamlessly with other modules through shared interfaces and identity controls, consistent with the deployment and data-flow architecture presented in the following chapters of this deliverable.
- **User-Centric and Role-Adaptive Experience**: The platform provides personalised learning pathways tailored to user roles, responsibilities and risk exposure. Role-based access control and

© CYberSynchorny, 2024

authentication mechanisms ensure that each user engages with relevant content and tools while maintaining the platform's security model.

- **Integration of Behavioural and Technical Insights**: Awareness content is informed by insights derived from monitoring, detection and intelligence components. By aligning learning with real or simulated threat patterns, the platform enables context-aware and dynamically updated training experiences.
- **Adaptivity Through Human Risk Management**: The platform incorporates behavioural metrics and performance indicators to adjust learning difficulty, frequency and content relevance. This supports a shift from static training to adaptive learning that reflects individual and organisational risk profiles.
- **Engagement Through Interactive and Scenario-Driven Features**: Simulation-based exercises, gamified components and scenario-driven activities enhance user engagement and reinforce practical understanding. These methods complement traditional instructional content by promoting experiential learning.
- **Sector-Sensitive Flexibility and Harmonisation**: The platform accommodates sector-specific needs and regulatory drivers, drawing on the multi-sector analysis presented in earlier project work [5]. This ensures that training and guidance align with domain-specific requirements without compromising interoperability across sectors.

These design principles provide a structured foundation for delivering an awareness platform that is technically integrated, behaviourally informed and adaptable to diverse organisational and sectoral contexts.

3.3 Accessibility, privacy and harmonization

The Cyber Awareness Platform is designed to serve as a pervasive and inclusive tool for cybersecurity education, targeting a diverse demographic ranging from technical professionals to general citizens. To achieve widespread adoption and compliance with European values, the platform's architecture and deployment are governed by three foundational non-functional pillars: accessibility for all user groups, rigorous data privacy protections, and harmonization of standards across sectors.

3.3.1 Accessibility and Inclusivity

In alignment with the Non-Functional Requirements defined in D2.2 [6], specifically NFR13 and NFR14, the platform prioritizes an inclusive user experience that removes barriers to entry for users with disabilities or language constraints.

- **Compliance with Accessibility Standards:** The User Interface (UI) is engineered to comply with the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA [7]. The frontend framework (developed in React/Vue.js) integrates semantic HTML, ARIA (Accessible Rich Internet Applications) [8] labels, and high-contrast visual modes. This ensures compatibility with assistive technologies such as screen readers, fulfilling the requirement to support users with disabilities identified in the end-user analysis.
- **Multi-Language Support:** To cater to the cross-border nature of the CYberSynchrony consortium and the European digital single market, the platform architecture supports dynamic localization. All educational content, navigation menus, and system notifications are stored in externalized resource files (JSON/XML), allowing for the seamless switching of languages. The initial deployment includes support for the primary consortium languages (English, Greek, Italian) to ensure users can engage with complex cybersecurity concepts in their native language.
- **Adaptive Learning Interfaces:** Recognizing the varied technical expertise of the target audience, the platform offers adaptive interfaces. The content delivery mechanism adjusts based on the user's profile, presenting simplified visualizations for general users while offering detailed technical logs for advanced practitioners.

© CYberSynchorny, 2024

3.3.2 Privacy by Design and GDPR Compliance

As the central repository for user training data and performance metrics, the platform processes Personally Identifiable Information (PII). Consequently, it operates under a strict "Privacy by Design" framework derived from the Data and Ethics Management task (T1.4) and the Data Protection Policy (Tool 10) defined in D6.3 [9].

- **Data Minimization and Identity Management:** The platform minimizes PII storage by offloading authentication to the **Keycloak Identity Manager**. The platform itself does not store user passwords or biometric data. Instead, it relies on OpenID Connect (OIDC) tokens for session management, ensuring that sensitive authentication credentials remain isolated within the secure identity provider.
- **Encryption and Data Security:** All data at rest, including user progress logs and assessment scores, is encrypted using AES-256 algorithms within the PostgreSQL database. Data in transit is protected via TLS 1.2 or higher, enforcing the "Secure Communication Protocols" requirement (NS4) defined in the Network Security Policy.
- **User Rights Management:** The platform includes self-service features that allow users to exercise their GDPR rights. This includes functionality for users to export their training history (Right to Data Portability) and to request account deletion (Right to be Forgotten), which triggers a cascading deletion of their data across the connected modules.

3.3.3 Harmonization of Standards

The platform acts as the unification point for the project's diverse training and policy outputs, ensuring consistency across the different sectors involved (Healthcare, Public Administration, Academia).

- **Sector-Specific Adaptation:** While the core platform remains uniform, the content delivery engine adapts to the specific regulatory needs of each sector. For example, healthcare users are presented with training modules emphasizing patient data privacy (HIPAA/GDPR), while public administration users focus on the integrity of public records. This capability is supported by the Sector-Specific Cybersecurity Harmonization task (T6.6).
- **Unified Competency Framework:** To ensure that the skills acquired are recognized across Member States, the educational curriculum is harmonized with the **European Cybersecurity Skills Framework (ECSF)** [10]. The platform maps specific training modules and gamified missions to ECSF skill profiles, allowing organizations to benchmark their workforce capabilities against a standardized European metric.
- **Policy Alignment:** The platform serves as the dissemination vehicle for the harmonized policies developed in D6.3 [9]. It ensures that users across all pilot sites are trained on the same versions of the "Acceptable Use Policy" (Tool 11) and "Incident Management Policy" (Tool 14), fostering a consistent security culture regardless of the organization's size or domain.

© CYberSynchorny, 2024

4 Platform Architecture and Integration

4.1 Deployment architecture

The CYberSynchrony platform is built on a modular, cloud-native architecture designed for interoperability, scalability, and layered cybersecurity defense. The system is structured around six core functional modules - **CYBRITE, CYBERRA, CYRESCUE, CROSS-CORE, CYBERWISE**, and **CYBERGOPLUS** - each representing a distinct layer of cybersecurity capability, from threat monitoring and intelligence analysis to incident response, human awareness, and compliance governance. The architecture combines **cloud-based services** (for heavy processing, correlation and storage) with on-premise data collection agents, which results in a clear integration between local infrastructure and centralized analytical functions. More detailed information on how the requirements gathered from the workshops and end-users were mapped to the actual functional and non-functional requirements that guided the way towards the project's architecture can be found in [5] and [6].

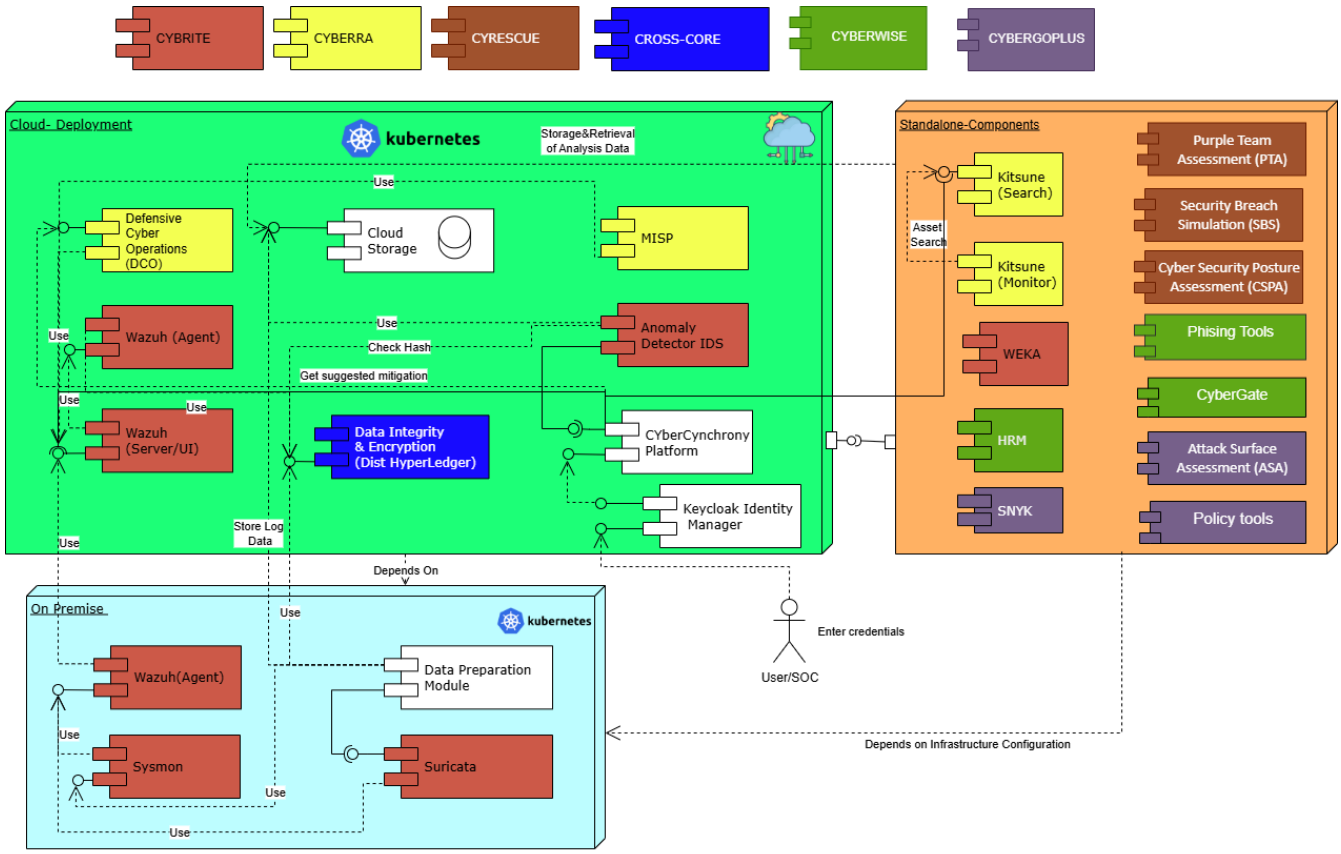


Figure 1. Deployment Architecture

Regarding the deployment architecture, each component within the system - whether operational (e.g., Wazuh, Suricata, MISP) or analytical (e.g., WEKA, HRM, Snyk) - connects through standardized interfaces (REST APIs, JSON, STIX/TAXII) under a unified identity and access management layer powered by Keycloak. The use of Kubernetes for container orchestration ensures scalability and resilience across deployments, while BYTE's Data Integrity and Encryption (Hyperledger) guarantee end-to-end data authenticity, immutability and encryption through its Blockchain-based approach.

Now, if we move on a higher level and see the situation more abstractly, the following takes place: At the top, we have the **CYberSynchrony platform**. This is the **"control center" where users log in, see dashboards, and**

access all the tools. It also acts as a catalogue where components can be found and connected. To make sure everything is secure, **Keycloak** manages who can log in, what they can see, and handles secure communication. Underneath the platform, there are 6 modules. One for **monitoring**, one for **risk assessment**, one for **responding to incidents**, one for **training people**, and so on. Each toolbox can be used separately or together, depending on what a situation requires. Next to the modules, there are **data repositories**. These are **safe storage spaces** that hold both threat intelligence (global knowledge about attacks) and system logs (records of what happened in your own network). All the data stored here is encrypted (locked) so nobody can tamper it or steal it. Beneath everything sits in the **Data Integrity and Encryption layer**. This layer makes sure that all data is trustworthy and secure. It uses **Blockchain** to guarantee that no record can be secretly changed, and it uses encryption to keep sensitive information safe when it's being stored or moved around. Finally, there are the (on-premises) **data collection agents**. These are like sensors installed inside each organization's IT environment (e.g. Wazuh, Sysmon). They **gather logs and security data** from servers, devices, and networks, then the **Data Preparation Module** cleans and formats this information before sending it to the platform. Once the data reaches the cloud, the **Data Integrity (Hyperledger)** component applies a digital fingerprint (hash) and encryption to guarantee that it remains secure and tamper-proof.

4.2 Flow of information

The data flow within CYberSynchrony follows a secure, multi-stage pipeline that begins at the organizational level and culminates in cross-module analysis and response.

1. The process starts when a SOC analyst logs into the platform through the CYberSynchrony UI, authenticated via Keycloak, which enforces role-based access control and multifactor authentication.
2. Once logged in, on-premise agents - primarily Wazuh and Suricata, assisted by Sysmon - continuously collect endpoint, system, and network telemetry.
3. The Data Preparation Module (BYTE) cleans, normalizes, and formats this data locally before forwarding it to the cloud repository.
4. Before the data is stored, it passes through the Data Integrity and Encryption layer (Hyperledger), where it is hashed, encrypted, and anchored in a distributed ledger to ensure immutability.
5. In the cloud environment, Wazuh correlates security events and Anomaly Detector IDS (and Weka) applies AI/ML analysis to identify abnormal patterns.
6. MISP enriches these findings with global threat intelligence, while Kitsune Monitoring and Search provide dark web and entity-based context.
7. The Defensive Cyber Operations (DCO) module then uses these inputs to trigger automated playbooks, suggesting or executing response actions such as host isolation or process termination.
8. Meanwhile, SOC analysts can launch on-demand components and tools, according to the features he needs (each feature belongs to a specific module).
9. All results feed back into the CYberSynchrony dashboard, where the user views live alerts, anomaly trends, compliance reports, and integrity validations.
10. The verified data and forensic artifacts can be shared securely across borders via CROSS-CORE, which uses blockchain anchoring and encrypted channels to preserve the chain of custody.

Ultimately, we strongly believe that CYberSynchrony will establish a continuous, verifiable cycle of detection, intelligence, response, and governance - transforming raw telemetry into actionable cybersecurity insight.

4.3 Module features

4.3.1 CYBERRA – Cyber Threat Intelligence and Customized Risk Analysis

The **CYBERRA module** dashboard serves as the central intelligence and analysis hub for monitoring evolving cyber threats, risk exposures, and correlated attack data. The dashboard displays a **Threat Intelligence View** powered by **MISP**, where analysts can see flagged Indicators of Compromise (IPs, domains, and file hashes), active global or sector-specific threat campaigns, and cross-references showing which alerts from match known IoCs. A dedicated **Risk Identification panel**, driven by **Kitsune Monitoring**, highlights current risks such as

external exposures, leaked credentials, or dark web mentions related to the organization's assets. The **Enrichment Layer** provides context for each alert, tagging it with additional intelligence such as "linked to ransomware X campaign" or "confirmed by EU threat feed," and presenting a **timeline view** that tracks when a threat was first observed globally and when it appeared locally in the monitored environment. The dashboard also allows analysts to **run customized risk assessments and simulations** using data from CSPA, PTA, or SBS tools, visualizing the results as comparative heatmaps or severity scores. In essence, the CYBERRA dashboard consolidates global intelligence, local detection data, and analytical insights into a single operational view-giving SOC teams the ability to identify, prioritize, and act on cyber threats with full situational awareness.

4.3.2 CYRESCUE - Cybersecurity Resilience and Incident Response Enhancement

The **CYRESCUE module** manages incident response and resilience-building activities, enabling SOC teams to analyze, mitigate, and recover from security incidents efficiently. It integrates **Defensive Cyber Operations (DCO) playbooks, CSPA, PTA**, and **SBS** tools to coordinate automated or semi-automated reactions and verify system resilience after incidents or simulated breaches. The dashboard features a **Response Actions panel**, showing recommended actions from the DCO engine awaiting analyst approval, along with a "decision tree" visualization that tracks which mitigation paths were taken. The **Incident Response** section aggregates inputs from CSPA, PTA, and past incident reports to display ongoing cases, response timelines, and containment status. A Breach Simulation view lists executed simulations such as ransomware or privilege escalation tests, alongside visual summaries showing system performance and recovery effectiveness. This interface allows analysts to approve or deny automated actions, monitor incident lifecycles, and strengthen resilience based on data-driven feedback from both real and simulated scenarios.

4.3.3 CYBRITE - Proactive Cyber Risk Monitoring, and Incident Response

The **CYBRITE module** functions as the central **detection and monitoring layer** of the platform, responsible for continuously identifying, classifying, and correlating security events across endpoints and networks. Using **Wazuh, Suricata**, and **Sysmon**, it gathers and analyzes real-time telemetry to detect suspicious activities, while the **Anomaly Detector IDS** and **WEKA** apply **AI/ML-based analytics** to uncover behavioural deviations and insider threats. The dashboard presents a consolidated Live Alerts view, displaying Wazuh correlation results, Suricata intrusion alerts, and Sysmon activity such as suspicious processes or unauthorized connections. It also includes AI-based anomaly detections, providing immediate insight into unusual traffic or entity behaviour. A Summary Metrics panel shows the number of active alerts, flagged anomalies, and data volume processed, alongside system "health status" indicators confirming agent connectivity and pipeline synchronization. Through this unified view, analysts can monitor ongoing incidents, investigate anomalies, and verify that all monitoring agents and data flows remain fully operational.

4.3.4 CROSS-CORE - Cross-Border Cyber Incident Cooperation & Response

The **CROSS-CORE module** enables secure collaboration between Member States, national CERTs, and critical infrastructure operators during large-scale or cross-border incidents. Its foundation lies in blockchain-backed data integrity verification and encrypted data sharing through the Secure Data Exchange Platform, ensuring authenticity and compliance in all information exchanges. The dashboard includes an Evidence Exchange view listing incident packages shared with partners, each with blockchain reference IDs confirming integrity and a "green/red" validation status for tamper checks. The Incoming Data Panel displays alerts, IoCs, and forensic logs received from external sources, with validation results clearly marked. A Cross-Border Collaboration Status panel tracks active joint investigations and shared response efforts, while the Communication Log documents secure messages exchanged between SOCs and CSIRTs. A Compliance Indicator highlights whether all exchanged data adheres to GDPR and NIS2 rules. Together, these views give operators a complete situational overview of ongoing collaborations, verified evidence chains, and international coordination status.

© CYberSynchorny, 2024

4.3.5 CYBERWISE - Cybersecurity Education for All

The **CYBERWISE module** strengthens the human layer of cybersecurity by focusing on awareness, training, and behavior change. It integrates phishing and social engineering simulation tools, **Human Risk Management (HRM) analytics**, and **gamified learning platforms like CyberGate**, providing interactive, scenario-based education aligned with real-world attack tactics. The dashboard presents an **Awareness Campaigns & Training** view, listing all active awareness initiatives and tracking completion rates, participation progress, and overall engagement metrics. The **Phishing & Social Engineering Simulation** Results section summarizes simulation outcomes such as click rates, report rates, and time-to-detect-offering a heatmap of vulnerable departments or users. The **Human-Centered Training** panel highlights behavioral risk trends detected by HRM and recommends tailored training for high-risk employees. Finally, a **Training Library** provides access to e-learning modules, quizzes, and refresher policies, along with recommendations for users who repeatedly fail simulations. This environment allows organizations to manage awareness campaigns, measure behavioral improvement, and foster a culture of cybersecurity accountability.

4.3.6 CYBERGOPLUS - Cybersecurity Governance and Compliance

The **CYBERGOPLUS module supports governance, auditing, and regulatory compliance** by linking technical security data with policy-level management and reporting. It integrates **Attack Surface Assessment (ASA), Snyk, and internal Policy Tools** to automate compliance tracking against NIS2, GDPR, and ISO 27001 frameworks. The dashboard provides a **Compliance Overview**, visualizing the organization's adherence levels via charts, timelines, and department-based filters for open versus resolved issues. The ASA Results panel lists external-facing assets, open ports, and misconfigurations with risk classifications and vulnerability context (CVE/CWE mapping). The Snyk Vulnerability Scan view shows detected code and dependency flaws across repositories, grouped by severity and remediation progress. A **Policy & Governance panel** tracks active cybersecurity policies, alerts on expired or violated rules, and offers an Evidence Collection view with audit logs and compliance reports. This consolidated interface empowers managers to assess infrastructure security, monitor compliance posture, and generate audit-ready documentation for regulators or internal review.

4.4 Components at a glance

No	Component	Partner	Description
1	Keycloak Identity Manager	MAG	Centralized **authentication and authorization**, enabling **secure access control** for applications. It **manages user identities, supports single sign-on (SSO)**. (*used in UI/UX dashboard*)
2	Anomaly Detector IDS	MAG	**AI based network anomaly detectors and classifications**. This component analysis log and network data **identifies anomalies and classifies them based on the data patterns to possible known attacks**. Uses AI to **constantly monitor network activity and flag anything unusual**, helping to catch cyber threats early.
3	Wazuh	IANUS	Open-source platform providing XDR and SIEM capabilities, covering **endpoint security, cloud protection, vulnerability assessment and incident response**. A platform that **watches over computers and networks, spotting security problems and responding quickly to keep systems safe**, whether on-site or in the cloud.

© CYberSynchorny, 2024

4	MISP	IANUS	**Open-source threat intelligence platform to enhance threat prediction capabilities**, enabling proactive threat detection and monitoring. A tool that helps collect, analyze, and share information about cyber threats. It uses AI to predict attacks and alerts organizations in advance.
5	Suricata	IANUS	Continuously **monitors network traffic, providing alerts and logs that are crucial for identifying and mitigating cyber threats** in real-time.
6	SNYK	TRUST	Real-time static application security testing (SAST) tool for **detecting vulnerabilities in software**. **Ensures software security and aligns with cybersecurity standards**. A security tool that checks software code for weaknesses before it's launched.
7	Human RM (HRM)	TRUST	**Risk assessment and security management package that focuses on human factors like attackers' profiles, awareness levels and behavioural characteristics**. Focuses on the human side of cybersecurity by checking how employees handle security and offering training to improve their awareness.
8	Blockchain Integrity Validator (Data Integrity and Encryption)	BYTE	Distributed ledger integrity checker with enhanced with Encryption methods. **Makes sure that important data hasn't been tampered with, using blockchain to keep records safe and trustworthy.**
9	Cyber Security Posture Assessment (CSPA)	FP	**Evaluates an organization's cybersecurity posture** using industry benchmarks to enhance security infrastructure and mitigate risks.
10	Attack Surface Assessment (ASA)	FP	**Discovers, evaluates and mitigates the exposure of internet-connected assets** to reduce the organization's attack surface.
11	Purple Team Assessment (PTA)	FP	**Enhances an organization's ability to respond to incidents** by aligning technical defences with the MITRE ATT&CK framework.
12	Security Breach Simulation (SBS)	FP	Conducts **ransomware vulnerability assessments to identify and address weaknesses** in an organization's infrastructure.
13	Defensive Cyber Operations (DCO) (Response Playbooks)	FP	Provides **live cyberattack simulations and defensive training for cybersecurity teams**. This component deploys **automated playbooks for real-time, standardized responses to cyber threats**, minimizing impact and ensuring continuous protection.
14	Kitsune Risk	MER	Customised **risk analysis and early warning notifications** for network exposures and cyber threats. Tailored to enhance interoperability.

© CYberSynchorny, 2024

15	Kitsune Monitoring	MER	Proactive **cyber risk monitoring and incident response** using intelligence sources to analyse threat behaviours and vulnerabilities.
16	WEKA	TRUST	This component integrates the WEKA tool to **analyse and model entity behaviour (UEBA)**, enhancing threat detection by identifying anomalies and patterns. **An ML tool that studies network traffic and figures out if something suspicious is happening.**

4.5 Identity & Access Management with Keycloak

The Identity and Access Management (IAM) component ensures secure, consistent and role- appropriate access to all platform functionalities. The IAM layer is implemented through **Keycloak**, an open-source identity and access management system selected for its native support for federation, role-based access control, fine-grained policy enforcement, and interoperability with modern authentication standards (OIDC, OAuth2, SAML).

Keycloak serves as the **single authentication and authorization provider** for the CYberSynchrony platform, enabling users to access the Cyber Awareness Platform, gamified training modules, compliance dashboards, and other system components through a unified identity. This ensures a seamless user experience and supports strong security controls across the entire ecosystem.

4.5.1 Centralized Authentication

The platform uses Keycloak to authenticate all users through a centralized login process. This approach provides:

- **Single Sign-On (SSO)** across all CYberSynchrony modules.
- **Support for federated identities**, including organizational directories and external identity providers where applicable.
- **Standards-based authentication**, enabling integration with all platform services via OIDC and OAuth2.

Authentication workflows ensure that sensitive components, such as phishing simulation dashboards, policy-compliance tools, or training analytics, are only accessible to validated and trusted users.

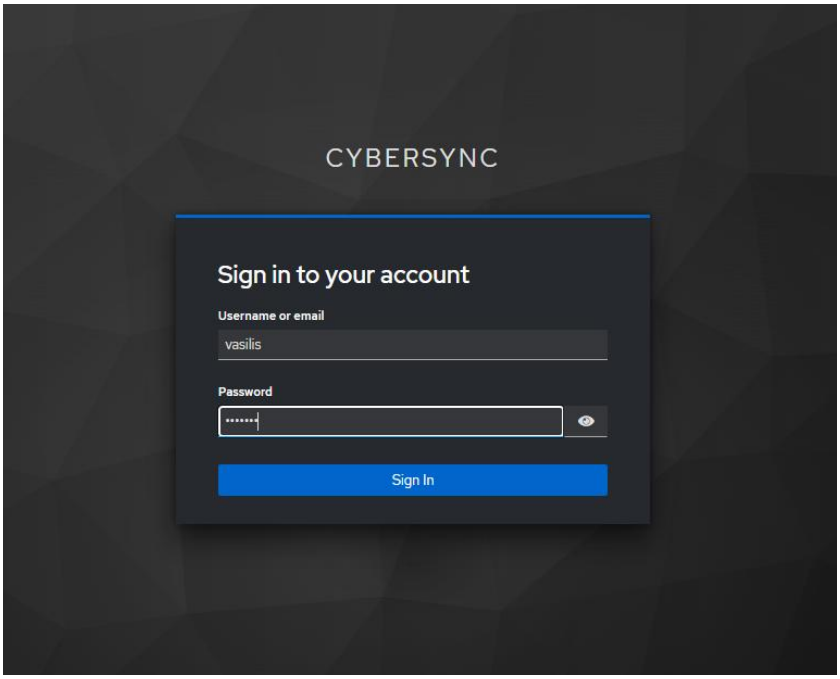


Figure 2. Single Sign-On via Keycloak

© CYberSynchorny, 2024

4.5.2 Role-Based Access Control (RBAC)

Keycloak manages fine-grained authorization policies, enabling differentiated access based on user roles. Examples include:

- **General users**: access to awareness content, gamified modules, and training materials.
- **Managers or administrators**: visibility into training progress, team-level analytics, and compliance readiness indicators.
- **Platform operators**: access to system configuration, monitoring dashboards, and module interconnections.

Role mappings are enforced at login, allowing downstream services to receive user attributes and permissions through secure tokens.

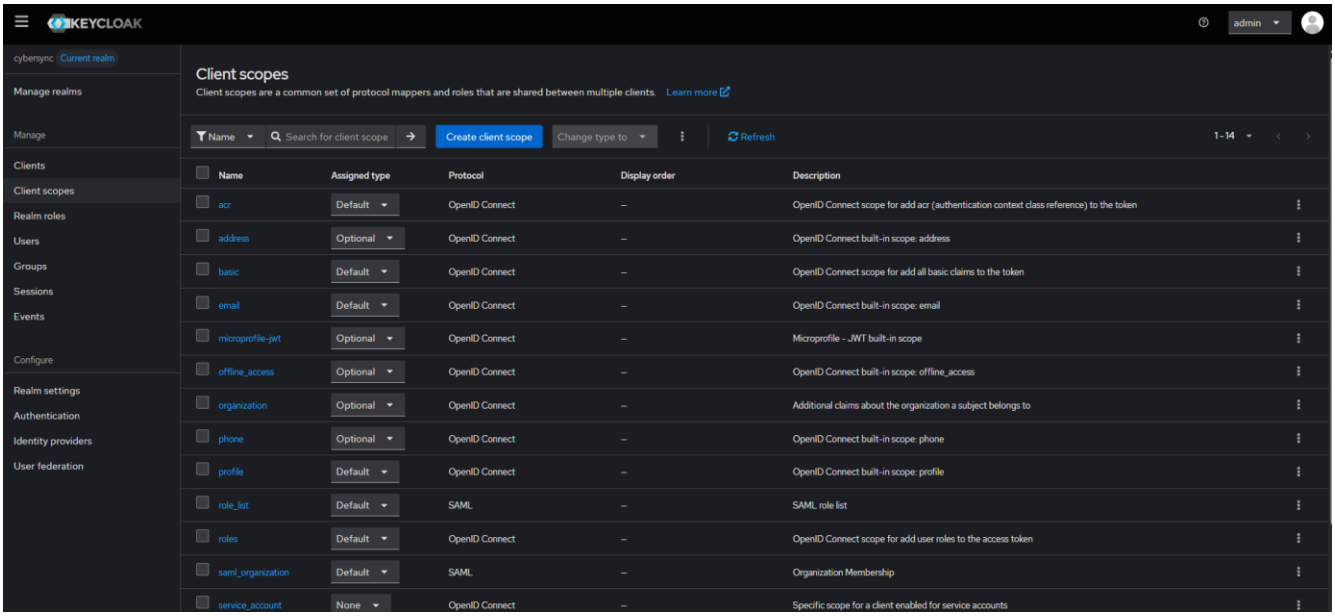


Figure 3. Platform's Client scopes

4.5.3 Integration With CYberSynchrony Modules

All platform modules rely on Keycloak-issued tokens to validate user identity and permissions. This integration ensures:

- Secure API calls between the Cyber Awareness Platform and modules such as CYBERWISE, CYBERGOPLUS, and CYBRITE.
- Consistent user identification across training records, behavioral analytics, phishing campaign participation, and policy compliance activities.
- Harmonized user management across cross-border or multi-organization deployments.

Module access control is achieved by verifying Keycloak-issued JWT tokens within each service's backend.

© CYberSynchorny, 2024

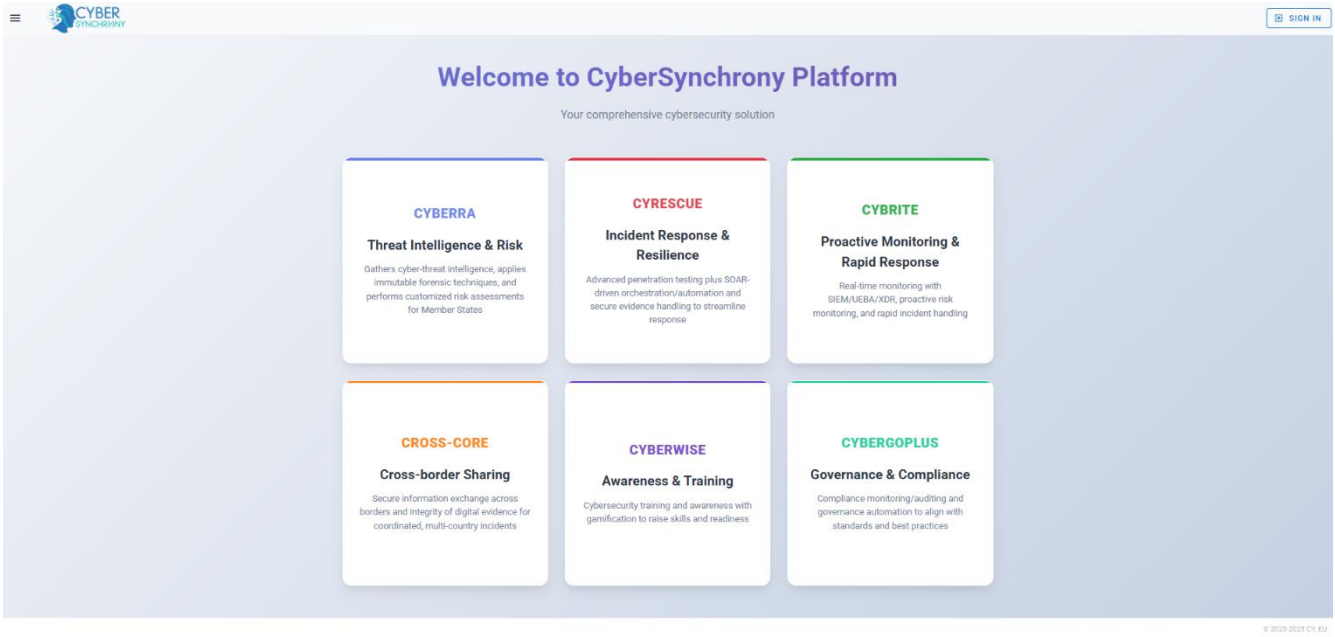


Figure 4. CYberSynchrony modules

Figure 5. Example of CYBRITE module environment

4.5.4 Attribute-Based Personalization

Keycloak also provides user attributes—such as sector affiliation, organization, or department—used by the Awareness Platform to deliver **personalized learning pathways**. These attributes enable:

- **Dynamic content recommendations** (e.g., sector-specific modules).
- **Adaptive difficulty or sequencing** based on user role or previous performance.
- **Sector-based harmonization**, leveraging metadata tags within the training engine.

© CYberSynchorny, 2024

4.5.5 Security and Privacy Considerations

Keycloak configuration follows cybersecurity and privacy best practices, ensuring:

- Enforced **multi-factor authentication (MFA)** where applicable.
- Secure handling of personal data in alignment with **GDPR** and relevant organizational policies.
- Regular token refresh and session management controls.
- Encrypted communication between Keycloak, backend services, and user clients.

4.6 Technology stack

The technology stack supporting the Cyber Awareness Platform is designed to ensure interoperability with the broader CYberSynchrony ecosystem, secure deployment, modular extensibility, and alignment with modern DevSecOps practices. The stack combines container orchestration, identity and access management, certificate automation, networking components, and CI/CD integrations, as described in the deployment and integration sections of this deliverable.

Containerisation and Orchestration

The platform is deployed on a **Kubernetes-based environment**, prepared and initialised using *kubeadm*, ensuring consistency with industry-standard cluster lifecycle management. This includes:

- **Kubernetes Control Plane (kubeadm)** for managing core cluster services.
- **Docker** as the underlying container runtime.
- **Calico (CNI)** for network overlay, routing, and network policy enforcement.
- **ingress-nginx** for managing external HTTP/S traffic and ingress routing.

Security and Certificate Management

Security mechanisms for service communication and ingress traffic rely on:

- **cert-manager**, providing automated certificate issuance and renewal.
- Support for **ACME/Let's Encrypt** or internal Certificate Authorities.
- **TLS-based service exposure**, with cluster issuers for staging and production environments
- **ImagePullSecrets** for private registry access.

Identity and Access Management

- **Keycloak** as the unified IAM provider across the CYberSynchrony ecosystem (referenced in the Identity & Access Management section of this deliverable)

DevOps and CI/CD

The platform integrates with GitLab for CI/CD, leveraging:

- **GitLab Kubernetes Agent (KAS)** for secure cluster-agent communication.

Frontend/UI

- Javascript (vue.js)

4.7 Deployment models

This chapter outlines a structured, repeatable process for deploying a production-ready Kubernetes cluster and integrating it with GitLab.com through the GitLab Kubernetes Agent (KAS). The deployment framework is designed to ensure consistency, reliability, and minimal manual intervention, using a sequence of six idempotent stages that can be executed multiple times without adverse effects.

© CYberSynchorny, 2024

The deployment begins by preparing the host node, configuring core components such as containers and the **Certificate Authority (CA) trust store)** to establish a secure and stable runtime environment. The **control plane** is then initialized using **kubeadm**, which automates the setup of essential Kubernetes control components and ensures alignment with production best practices.

Next, the network layer is configured through the installation of **Calico**, a Container Network Interface (CNI) that provides robust networking and network policy enforcement. Following this, **ingress-nginx** is deployed to manage external HTTP/S traffic, enabling controlled and scalable access to cluster-hosted applications.

Security and certificate management are addressed through the installation of **cert-manager**, which automates the issuance and renewal of TLS certificates. The deployment supports both internal certificate authorities and external ACME issuers, such as Let's Encrypt, allowing flexible integration with existing security models.

Finally, the **GitLab Kubernetes Agent** is deployed, establishing a secure communication channel between the cluster and GitLab.com. This integration enables continuous integration (CI) workflows, GitOps-style deployments, and centralized policy management directly from the GitLab interface.

Additional configuration topics include the management of **private registry access** using ImagePullSecret, **DNS and TLS validation** to ensure service reliability, and the implementation of **health checks** for continuous cluster monitoring. The document also provides focused troubleshooting guidance to streamline issue resolution during deployment or operation.

Upon completion of this process, the resulting cluster is fully observable and manageable from GitLab, offering a foundation for automated CI/CD pipelines, declarative infrastructure control, and scalable production operations.

4.7.1 Prerequisites

- OS: Ubuntu 22.04+ with sudo
- Network: Outbound internet to APT, GitHub (manifests/Helm), and any private registries; firewall must allow Node → Internet and (optionally) inbound NodePort/LB ports.
- Files (provided):
    - ebos-root-ca.crt — Organization Root CA (used by containerd, OpenSSL, and scripts).
    - registry-secret.yaml — Contains example kubectl create secret commands. Replace credentials before use.
- Hostnames/IPs used by scripts (examples): registry.ebos.local, kas.ebos.local, gitlab.ebos.local, plus a VIP/IP for services.
- Tools: curl, openssl, helm (auto-installed by scripts when missing), kubeadm/kubelet/kubectl (installed during Step 1).

Idempotency: All scripts are written to be safely re-run; they check state and skip completed actions where possible. They also support interactive prompts and environment overrides.

4.7.2 Node Preparation — 01-prepare-node.sh

The Node Preparation stage establishes the operating system and base runtime environment necessary for Kubernetes. This includes disabling swaps, applying kernel parameters, installing and configuring containerd, installing Kubernetes packages, integrating enterprise CA trust, and ensuring reliable hostname resolution. By automating these steps, the script ensures a consistent, production-ready foundation across all cluster nodes.

Script Functionality

The script "*01-prepare-node.sh*" automates node configuration to ensure standardization and minimize manual intervention. It begins by elevating privileges to root, enabling strict bash safety flags (set -euo pipefail), and initializing structured logging for readable and traceable execution output. The script is designed to prompt for key configuration parameters, such as the Kubernetes release channel, enterprise CA details, virtual IP, and

© CYberSynchorny, 2024

registry access credentials, but also allows these to be provided as environment variables for unattended runs. Typical inputs include:

- **Kubernetes version/channel:**
  - K8S_CHANNEL — Example: v1.30 (defines versions of kubeadm, kubelet, and kubectl)
- **Enterprise CA configuration:**
  - EBOS_CA_SRC — Path to Root CA PEM
  - EBOS_CA_NAME — Root CA filename
- **Network and host configuration:**
  - EBOS_IP — Virtual IP (VIP)
  - EBOS_HOSTS_INPUT — Hostnames to inject into /etc/hosts

```
systemctl is-active containerd kubelet
ctr images ls
openssl s_client -connect ${EBOS_REGISTRY_HOST}:${EBOS_REGISTRY_PORT} -
servername ${EBOS_REGISTRY_HOST} </dev/null
```

Figure 6. Sample of prompts with env overrides.

- **Private container registry:**
  - EBOS_REGISTRY_HOST — Registry hostname
  - EBOS_REGISTRY_PORT — Registry port

Once parameters are defined, the script disables swap and comments out related entries in /etc/fstab, meeting Kubernetes' scheduling requirements. It loads necessary kernel modules (overlay and br_netfilter) and applies system network parameters via /etc/sysctl.d/99-kubernetes-cri.conf, including:

- net.bridge.bridge-nf-call-iptables=1
- net.bridge.bridge-nf-call-ip6tables=1
- net.ipv4.ip_forward=1

These settings are applied with sysctl --system to ensure proper network traffic handling.

The script proceeds to install core system utilities, including apt-transport-https, ca-certificates, curl, gnupg, lsb-release, and netcat-openbsd, which provide secure communication and package management capabilities. It updates the /etc/hosts file with the defined VIP and hostnames, ensuring reliable hostname resolution across nodes. It also integrates the enterprise Root CA into both the system trust store and the containerd runtime configuration, enabling secure access to internal container registries. Next, the script installs and configures container with SystemdCgroup=true, ensuring compatibility with Kubernetes' control mechanisms, and proceeds to install the Kubernetes components kubeadm, kubelet, and kubectl. These services are enabled and verified to ensure the correct startup order and operational readiness. Then, the script performs a series of connectivity and trust checks. It validates hostname resolution, tests TCP connectivity to the specified container registry, and verifies TLS certificate chains against the enterprise CA. These checks confirm that the node can securely communicate with required internal services and registries.

Inputs and Outputs

- Inputs (interactive or environment variables):
  - K8S_CHANNEL

© CYberSynchorny, 2024

- EBOS_CA_SRC / EBOS_CA_NAME
- EBOS_IP / EBOS_HOSTS_INPUT
- EBOS_REGISTRY_HOST / EBOS_REGISTRY_PORT
- PAUSE_IMAGE

- Outputs / Resulting state:
  - Node is ready for kubeadm init or kubeadm join
  - Enterprise CA is installed and trusted system-wide and within containerd
  - Containerd can pull images from the private registry
  - Kubelet is enabled and operational

Verification

Node readiness can be quickly verified by checking:

- Swap is disabled
- kubelet and containerd services are active
- TLS-secured connectivity to the private registry succeeds

4.7.3 Control Plane Initialization — 02-init-control-plane.sh

The Control Plane Initialization stage uses kubeadm to set up the Kubernetes control plane, configure access credentials (kubeconfig), and generate a reusable command for adding worker nodes. This step establishes the core cluster services and ensures that subsequent nodes can join the cluster securely and consistently.

Script Functionality

The script *"02-init-control-plane.sh"* begins by elevating privileges to root and configuring the environment to use KUBECONFIG=/etc/kubernetes/admin.conf. Structured logging is enabled to produce clear and readable output. The script also supports interactive prompts to gather essential configuration parameters, which can alternatively be supplied as environment variables.

Key prompts include:

- CLUSTER_NAME (Name of the Kubernetes cluster)
- CONTROL_PLANE_ENDPOINT (Control plane VIP or hostname with port; e.g., <IP-or-host>:6443)
- POD_CIDR (CIDR range for pod networking)
- SERVICE_CIDR (CIDR range for services)
- API_SANS_EXTRA (*optional*) (Additional Subject Alternative Names – SANs, for the API server)

To ensure idempotency, the script checks for the presence of /etc/kubernetes/manifests/kube-apiserver.yaml. If this file exists, the control plane has already been initialized, and the script skips reinitialization, generating only a new join-workers.sh command to safely add additional nodes.

The core initialization process involves writing a kubeadm-config.yaml configuration file and running kubeadm init. After initialization, the script copies admin.conf to the invoking user's ~/.kube/config and optionally to root, providing immediate access to the cluster using standard Kubernetes CLI tools.

Finally, the script generates an executable join-workers.sh file that contains the kubeadm token create --print-join-command output. This file allows worker nodes to be added to the cluster in a repeatable, secure manner without re-running the full initialization process.

© CYberSynchorny, 2024

Inputs and Outputs

- Inputs can be provided either interactively or via environment variables:
  - CLUSTER_NAME
  - CONTROL_PLANE_ENDPOINT
  - POD_CIDR
  - SERVICE_CIDR
  - API_SANS_EXTRA (optional)
  - OUT_DIR — Directory for output files, defaulting to /root/kubeadm
- Outputs / Resulting State

Upon successful execution the control plane is operational; worker nodes will appear as **NotReady** until the CNI plugin is installed, and output files are created in /root/kubeadm/:

- kubeadm-config.yaml (The generated kubeadm configuration file)
- join-workers.sh (Executable script for adding worker nodes)

These outputs provide a reusable, documented foundation for cluster expansion and subsequent configuration.

Example Configuration

A sample kubeadm-config.yaml is generated, reflecting the cluster name, network CIDRs, API endpoint, and any additional SANs provided. This file serves both as documentation and as a reference for cluster recreation or troubleshooting.

```yaml
apiVersion: kubeadm.k8s.io/v1beta3
kind: InitConfiguration
nodeRegistration:
  criSocket: unix:///run/containerd/containerd.sock
---
apiVersion: kubeadm.k8s.io/v1beta3
kind: ClusterConfiguration
clusterName: <CLUSTER_NAME>
networking:
  podSubnet: <POD_CIDR>
  serviceSubnet: <SERVICE_CIDR>
controlPlaneEndpoint: <CONTROL_PLANE_ENDPOINT>
apiServer:
  certSANs:
    - <CONTROL_PLANE_ENDPOINT_HOST>
    # + values from API_SANS_EXTRA
```

Figure 7. Sample kubeadm-config.yaml (generated).

```
kubectl get nodes -o wide
cat /root/kubeadm/join-workers.sh
```

Figure 8. Sample of quick verify in control plane.

© CYberSynchorny, 2024

4.7.4 Calico Network Installation (CNI) — 03-calico.sh

The Calico Network Installation stage configures Kubernetes networking by deploying the Calico CNI plugin and ensures that all cluster nodes are fully operational. This step establishes pod-to-pod connectivity, enforces network policies, and transitions nodes to a "**Ready"** state so workloads can be scheduled reliably.

Script Functionality

The script "*03-install-calico.sh*" first validates that the Kubernetes API is reachable using kubectl, ensuring that the control plane is responsive before proceeding. It then prompts for essential configuration parameters, which can also be supplied as environment variables:

- CALICO_VERSION (Specifies the Calico release version to deploy)
- POD_CIDR (Defines the pod network CIDR for the cluster)
- ROLLOUT_TIMEOUT (Maximum wait time for Calico pods to become ready)
- FORCE_IPTABLES (Option to enforce iptables mode instead of eBPF)
- UNTAINT_CONTROL_PLANE (Optional flag to allow workloads on the control plane (typically used in lab or testing environments)

Once parameters are set, the script ensures that the host directories required by CNI exist. It downloads and applies the appropriate Calico manifest (saved to ${CALICO_MANIFEST}), configured according to the cluster's POD_CIDR. If enabled, the script removes the "**NoSchedule"** taint from the control plane, allowing pods to be scheduled there.

Following deployment, the script monitors the rollout of Calico components, providing progress output, and polls all nodes until they report as "**Ready"**. This ensures the cluster networking is fully operational before scheduling workloads.

Inputs and Outputs

- **Inputs**
    - CALICO_VERSION
    - POD_CIDR
    - ROLLOUT_TIMEOUT
    - FORCE_IPTABLES
    - UNTAINT_CONTROL_PLANE
- **Outputs / Resulting State**

Upon successful execution Calico DaemonSets and Deployments are running in the kube-system namespace, and kubectl get nodes reports all nodes as "**Ready"**, indicating that the network is fully operational and nodes are prepared to host workloads.

Verification

Cluster readiness can be quickly verified by checking:

- Calico pods are running and available in kube-system
- All nodes are in the **Ready** state using kubectl get nodes

© CYberSynchorny, 2024

```
kubectl get nodes -o wide
kubectl -n kube-system get pods -l k8s-app=calico-node
```

Figure 9. Sample of Quick Verify in CNI.

4.7.5 Ingress Controller — 04-ingress-nginx.sh

The Ingress-NGINX Installation stage deploys and configures the ingress controller using Helm, providing HTTP and HTTPS access to services running within the Kubernetes cluster. This ensures that workloads can be exposed securely and consistently, while allowing advanced features such as ingress class management and metrics collection.

Script Functionality

The script *"04-install-ingress.sh"* first validates that the Kubernetes API is reachable via kubectl and checks for the presence of Helm, automatically installing it if it is missing. It then prompts for key configuration parameters, which can also be provided as environment variables, including:

- **Namespace and Release Configuration:**
  - NS (Namespace for the ingress controller)
  - RELEASE (Helm release name)
- **Service Exposure Settings:**
  - SVC_TYPE (Service type NodePort or LoadBalancer)
  - HTTP_NODEPORT / HTTPS_NODEPORT (NodePort values for HTTP and HTTPS traffic)
- **Ingress Class Settings:**
  - INGRESS_CLASS (Name of the ingress class for the controller)
  - CLASS_DEFAULT (Whether this ingress class is the default)
  - WATCH_INGRESS_WITHOUT_CLASS (Enable controller to watch ingresses without a class)
- **Additional Features:**
  - ENABLE_METRICS (Enable Prometheus metrics)
  - CHART_VERSION (Specific version of the ingress-nginx Helm chart)
  - ROLLOUT_TIMEOUT (Maximum wait time for rollout completion)

The script creates the specified namespace if it does not exist, adds or updates the Helm repository for ingress-nginx, and installs or upgrades the chart using the values derived from the prompts. Once deployed, it waits for the rollout to complete, discovers the controller Service, and prints the access details, including Node IPs and NodePorts or LoadBalancer IP depending on the service type.

Inputs and Outputs

- **Inputs**
  - NS
  - RELEASE
  - SVC_TYPE
  - HTTP_NODEPORT / HTTPS_NODEPORT

© CYberSynchorny, 2024

- INGRESS_CLASS
- CLASS_DEFAULT
- WATCH_INGRESS_WITHOUT_CLASS
- ENABLE_METRICS
- CHART_VERSION
- ROLLOUT_TIMEOUT

- **Outputs / Resulting State**

Upon successful execution, the ingress-nginx namespace was created. The ingress controller Deployment and Service are running and accessible, and cluster services can now be exposed externally over HTTP and HTTPS.

Verification

Quick verification can be performed by checking that:

- The ingress-nginx namespace contains running controller pods and Services.
- Access details (Node IP + NodePorts or LoadBalancer IP) correspond to the configured parameters.

```
kubectl -n ingress-nginx get pods,svc
# If NodePort:
curl -I http://<NODE_IP>:<HTTP_NODEPORT>
```

Figure 10. Sample of Quick verify in Ingress Controller.

4.7.6 cert-manager Core — 05-cert-manager-core.sh

The cert-manager Core Installation stage deploys the cert-manager components using Helm and optionally creates "Let's Encrypt ACME ClusterIssuers" for automated TLS certificate management using the HTTP-01 challenge. This step ensures that the cluster can automatically provision and renew TLS certificates, enabling secure ingress of traffic for workloads.

Script Functionality

The "*script 05-cert-manager-core.sh*" begins by validating Kubernetes API reachability via kubectl and confirming that Helm is installed, installing it automatically if necessary. It then prompts for configuration parameters, which can also be provided through environment variables:

- **Namespace and Release Settings:**
  - NS (Namespace in which to install cert-manager)
  - RELEASE (Helm release name)
  - CHART_VERSION (Version of the cert-manager Helm chart)
- **Let's Encrypt ACME Issuer Settings – *optional*:**
  - CREATE_HTTP01_ISSUERS (Enable creation of HTTP-01 ClusterIssuers)
  - LE_INGRESS_CLASS (Ingress class to bind the HTTP-01 solver)
  - LE_EMAIL (Email address for Let's Encrypt registration)

© CYberSynchorny, 2024

Once parameters are configured, the script adds the Jetstack Helm repository and installs or updates the cert-manager chart, ensuring that all required CRDs (Custom Resource Definitions) are applied. If ACME issuers are enabled, the script creates two ClusterIssuers—letsencrypt-staging and letsencrypt-prod—configured to solve HTTP-01 challenges through the specified ingress class.

After deployment, the script prints usage guidance, including tips for annotating Ingress resources to request certificates and verifying HTTP-01 reachability. This ensures administrators can immediately leverage automated certificate management within their cluster.

Inputs and Outputs

- **Inputs**
  - NS
  - RELEASE
  - CHART_VERSION
  - CREATE_HTTP01_ISSUERS
  - LE_INGRESS_CLASS
  - LE_EMAIL (required if issuers are enabled)
- **Outputs / Resulting State**

Upon successful execution the cert-manager namespace is created with all cert-manager Deployments running. If requested, Let's Encrypt ClusterIssuers (letsencrypt-staging and letsencrypt-prod) are installed and ready for certificate issuance.

Below is an example of a letsencrypt-prod ClusterIssuer as generated by the script:

```yaml
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt-prod
spec:
  acme:
    email: <you@example.com>
    server: https://acme-v02.api.letsencrypt.org/directory
    privateKeySecretRef:
      name: le-prod-account-key
    solvers:
      - http01:
          ingress:
            class: nginx
```

Figure 11. Example of ClusterIssuer (prod).

Verification

Quick verification can be performed by:

- Confirming cert-manager pods are running in the cert-manager namespace.
- Checking that the optional ClusterIssuers exist and are in a Ready state using kubectl get clusterissuer.
- Optionally, testing certificate issuance via an annotated Ingress using the HTTP-01 challenge.

© CYberSynchorny, 2024

```
kubectl -n cert-manager get deploy,pods
```

Figure 12. Quick verify of cert-manager.

4.7.7 GitLab Agent (KAS) — 06-install-gitlab-agent.sh

The GitLab Kubernetes Agent (KAS) Installation stage installs and configures the GitLab Agent within the cluster, enabling management from GitLab.com. This integration provides secure CI/CD access, GitOps workflows, and remote debugging capabilities, allowing teams to control cluster operations directly from GitLab.

Script Functionality

The script 06-install-gitlab-agent.sh first validates connectivity to the Kubernetes API and confirms that both kubectl and Helm are installed. It then prompts for essential configuration parameters, which can also be supplied through environment variables:

- **Agent and Deployment Settings:**
  - AGENT_NAME (Name of the GitLab Agent)
  - NAMESPACE (Kubernetes namespace for the agent deployment)
  - IMAGE_TAG (Container image tag for the agent)
- **KAS Endpoint Configuration:**
  - KAS_HOST (Hostname of the GitLab Kubernetes Agent server)
  - KAS_PORT (Port for agent communication)
  - KAS_SCHEME (Communication scheme - typically wss)
  - KAS_CA_CERT (CA certificate for TLS verification)
- **Optional DNS Configuration:**
  - PATCH_COREDNS_HOSTS (Enable static host injection in CoreDNS)
  - HOSTS_IP / HOSTS_NAMES (IP and hostname entries for internal name resolution)
  - RUN_DNS_TEST (Test DNS resolution after patching)

The script optionally patches CoreDNS to add static host entries, which is useful when the KAS hostname is internal-only and external DNS is unavailable. CoreDNS is restarted to apply these changes.

TLS connectivity to the KAS endpoint is verified using the provided CA via openssl s_client with SNI to ensure secure communication. After validation, the script installs or upgrades the GitLab Agent Helm chart in the specified namespace and waits for the deployment to reach a Ready state. For immediate feedback, it then tails the last 50 log lines from the agent pod, providing quick diagnostics.

Inputs and Outputs

- **Inputs**
  - AGENT_NAME
  - NAMESPACE
  - IMAGE_TAG
  - KAS_HOST
  - KAS_PORT
  - KAS_SCHEME
  - KAS_CA_CERT
  - AGENT_TOKEN

© CYberSynchorny, 2024

- PATCH_COREDNS_HOSTS (optional)
- HOSTS_IP / HOSTS_NAMES (optional)

- **Outputs / Resulting State**

Upon successful execution the GitLab Agent is deployed and running in the chosen namespace. Also,:

- The agent is visible under **GitLab → Infrastructure → Kubernetes clusters**, confirming connectivity.
- If CoreDNS patching was enabled, internal hostnames are resolvable within the cluster without modifying external DNS.

Verification

Quick verification can be performed by:

- Confirming that the agent deployment is running and pods are Ready.
- Checking logs for connection status using kubectl logs (last 50 lines).
- Ensuring the cluster appears under GitLab's infrastructure view and is responsive to GitOps operations.
- Validating internal DNS resolution if CoreDNS patching was applied.

4.7.8 Linking the Cluster to GitLab.com

This stage connects the Kubernetes cluster to GitLab.com using the GitLab Kubernetes Agent (KAS), enabling GitLab to manage the cluster for CI/CD, GitOps, and remote debugging. Linking the cluster establishes a secure, token-based connection that allows GitLab pipelines and deployments to interact with the cluster without manual credentials management.

Procedure

To link the Kubernetes cluster to GitLab.com, the first step involved creating a GitLab Agent by navigating to the target project or group, selecting Infrastructure → Kubernetes clusters → Connect a cluster (agent), and copying the provided registration token. The agent was then installed in the cluster by executing the "*06-install-gitlab-agent.sh*" script and providing the registration token when prompted, or by exporting it as AGENT_TOKEN for non-interactive execution. Following installation, the agent configuration was committed to the repository, either as a minimal configuration to enable the connection or with additional files for GitOps-based deployments and policy management. Optionally, private registry access was configured using a registry-secret.yaml file, which contained example commands to create a Docker registry secret and patch the default ServiceAccount.

© CYberSynchorny, 2024

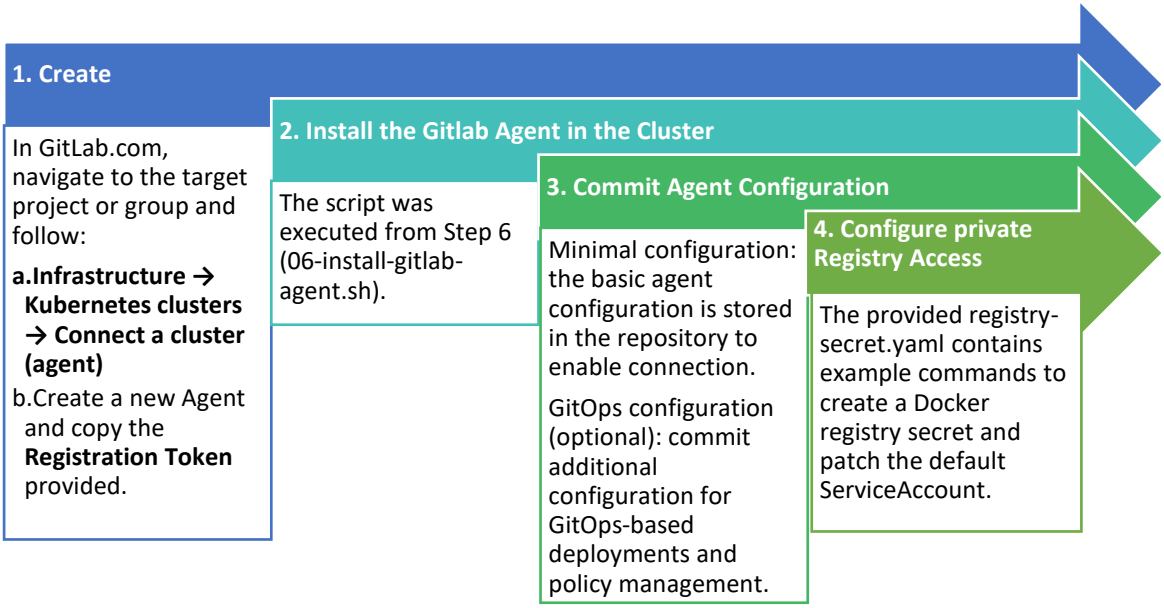


Figure 13. Scheme outlining the procedure of linking the cluster to Gitlab.com.

Security Considerations and Cluster Outputs

To maintain security when integrating a Kubernetes cluster with GitLab.com, credentials had been rotated regularly to minimize exposure, and robot or service accounts had been configured with only the minimal privileges required for cluster operations. Registry permissions had been restricted to the least necessary for workloads, reducing the potential attack surface.

After the integration, the cluster had been fully connected to GitLab.com and had appeared under Infrastructure → Kubernetes clusters. The GitLab Agent deployment had been activated and had been ready to manage workloads, enabling CI/CD pipelines, GitOps workflows, and remote cluster management. Additionally, if private registry access was configured, workloads were able to securely pull images using the provided secret, without exposing credentials in the repository. This setup had established a secure, manageable, and production-ready environment for deployment and operational tasks.

© CYberSynchorny, 2024

```
.gitlab/agents/<AGENT_NAME>/config.yaml
```

```
ci_access:
  projects:
    - id: <group>/<project>
```

```
gitops:
  manifest_projects:
    - id: <group>/<infra-repo>
      default_namespace: default
      paths:
        - glob: /clusters/<CLUSTER>/**/*.yaml
```

```
kubectl create secret docker-registry regcred \
  --docker-server=<registry-host> \
  --docker-username='<username>' \
  --docker-password='<password>' \
  --docker-email=<email>

# Optionally patch the default SA in a namespace to use regcred automatically
kubectl -n <namespace> patch serviceaccount default \
  --type='json' \
  -p='[{"op":"add","path":"/imagePullSecrets","value":[{"name":"regcred"}]}]'
```

```
spec:
  imagePullSecrets:
    - name: regcred
```

Figure 14. Samples of "commit agent configuration", "minimal configuration", "GitOps" (optional), "Private Registry & ImagePullSecret" (optional), and "reference the workloads".

4.8 Execution Order (Summary) and Cluster Validation Plan

The deployment of the Kubernetes cluster followed a structured, sequential process to ensure a fully operational environment. The scripts were executed in the following order:

1. **Node Preparation (01-prepare-node.sh)** – Configured the operating system, installed and configured containerd, established enterprise CA trust, and installed core Kubernetes packages.
2. **Control Plane Initialization (02-init-control-plane.sh)** – Initialized the control plane using kubeadm, configured kubeconfig, and generated join-workers.sh for worker nodes.
3. **Calico Installation (03-calico.sh)** – Deployed the Calico CNI plugin and ensured all nodes reached a Ready state.
4. **Ingress Controller (04-ingress-nginx.sh)** – Installed the ingress-nginx controller with NodePort or LoadBalancer service types for exposing workloads.
5. **Cert-Manager (05-cert-manager-core.sh)** – Installed cert-manager and, optionally, Let's Encrypt ACME ClusterIssuers for automated TLS certificate management.
6. **GitLab Agent (06-install-gitlab-agent.sh)** – Installed the GitLab Kubernetes Agent and connected the cluster to GitLab for CI/CD, GitOps, and remote management.

© CYberSynchorny, 2024

Quick Health Checks:

After deployment, cluster health will be validated using the following checks:

- **Nodes and System Pods:**
  - kubectl get nodes -o wide
  - kubectl -n kube-system get pods
- **Ingress and Cert-Manager:**
  - kubectl -n ingress-nginx get pods,svc
  - kubectl -n cert-manager get deploy,pods
- **GitLab Agent Connectivity:**
  - kubectl -n gitlab-agent-<AGENT_NAME> get pods
  - In GitLab UI, the cluster status should show Connected

While the scripted deployment process ensures a high degree of automation and reliability, certain post-deployment issues may arise due to environmental factors such as network restrictions, DNS configuration, or registry access. The following table summarizes the most common issues encountered during or after cluster setup, along with their likely causes and recommended corrective actions. These checks are intended to assist in diagnosing connectivity, readiness, and configuration problems efficiently, ensuring that the cluster reaches a stable and production-ready state.

Table 3. Possible issues and troubleshooting recommended actions.

Issue	Possible Cause / Check	Recommended Action
Agent shows "**Disconnected"**	• kasAddress misconfigured (scheme, host, port) • Firewall blocking TCP/443 • DNS for KAS hostname not resolving inside cluster	✓ Verify kasAddress parameters ✓ Adjust firewall rules to allow TCP/443 ✓ Check CoreDNS logs: kubectl -n kube-system logs deploy/coredns ✓ Enable CoreDNS hosts patch in Step 6 if internal-only DNS
Nodes remain "**NotReady"** after Calico deployment	• Calico pods not running • POD_CIDR mismatch • CNI errors in kubelet	✓ -Confirm Calico pods: kubectl -n kube-system get pods -l k8s-app=calico-node ✓ Ensure Calico POD_CIDR matches kubeadm init value ✓ Inspect kubelet logs: journalctl -u kubelet -e
Ingress / ACME certificate fails	• DNS A/AAAA records not pointing to cluster LB/Node IP • Ports 80/443 not publicly reachable	✓ Verify DNS records ✓ Ensure ports 80/443 are open ✓ Inspect challenge events: kubectl -n cert-manager describe challenge <name>
Registry pulls fail	• ImagePullSecret missing or not attached • TLS issues with registry	✓ Ensure ImagePullSecret exists in namespace and is attached to ServiceAccount ✓ Test TLS: openssl s_client -connect <registry>:443 and confirm CA trust

© CYberSynchorny, 2024

5 User/Cyber Awareness Content

5.1 Content pipeline overview

The Cyber Awareness Platform delivers training and awareness material through a structured content pipeline that ensures alignment with CYberSynchrony's threat landscape analysis, human-factor insights, training methodology, and sector-specific harmonization requirements. The pipeline defines how content is sourced, shaped, and delivered across modules such as CYBERWISE, CYBERGOPLUS, HRM analytics, gamified components, and social-engineering assessments, ensuring consistency and coherence across all user-facing materials.

The pipeline operates across four interlinked stages:

Input Gathering and Content Definition

Content creation begins with inputs from the project's foundational analyses, including:

- The multi-actor threat landscape and human-factor vulnerabilities identified in D2.1,
- Risk and behavioral insights emerging from proactive social-engineering activities (Section 5.3), and
- Governance and compliance requirements drawn from the Cybersecurity Policy Toolkit (Section 5.5).

These insights ensure that awareness content directly addresses real-world risks such as phishing, unsafe digital practices, sector-specific threats, and policy-relevant topics.

Content Design and Structuring

Using the CYBERWISE learning objectives (Chapter 3) and the methodology outlined in the Training Plan (Section 5.2), content is designed as modular learning objects that fit into structured campaigns or thematic training cycles.
This includes:

- core cyber hygiene content (e.g., phishing awareness, secure authentication),
- interactive and immersive components developed under Task 6.4 (Section 5.4),
- policy-aligned material based on the harmonized policy tools (Section 5.5), and
- practical technical modules such as the SNYK-based vulnerability assessment training (Section 5.8).

Content is tagged to enable sector-based customization as described in Section 5.6 (Sector-Specific Harmonization).

Content Integration into the Platform

Once designed, training components are integrated into the Cyber Awareness Platform, which serves as the unified environment for delivering:

- Phishing simulation results and follow-up remediation (Section 5.3),
- Structured training cycles (Section 5.2),
- Gamified scenarios from the Cybergate system (Section 5.4),
- Policy-driven training linked to CYBERGOPLUS governance tools (Section 5.5), and
- Sector-tailored modules identified through the harmonization model (Section 5.6).

Integration is supported by Keycloak-based user profiling and role-based access control (Section 4.5), enabling personalized content delivery based on user attributes, sector, or behavioral indicators.

Continuous Update and Improvement

The content pipeline follows an iterative improvement process grounded in:

© CYberSynchorny, 2024

- Feedback mechanisms and KPIs described in the Training Plan (Section 5.2.3),
- Behavioral indicators and HRM analytics used to identify training needs (referenced in CYBERWISE module description, Section 4.3.5),
- Insights from real-world phishing campaigns (Section 5.3), and
- Lessons learned from running experiential modules such as the SNYK training (Section 5.8).

These outputs inform future content refinement, enabling the platform to adjust difficulty, update materials, and incorporate new risks or compliance requirements as the ecosystem evolves

5.2 Training plan outline

The training plan establishes the methodological framework for executing all cybersecurity educational initiatives within the CYberSynchrony project.

Designed to directly support the objectives of the CYBERWISE module, the plan aims to implement cybersecurity awareness campaigns and educational programs targeting citizens, businesses and organizations, to foster a culture of cybersecurity awareness and best practices across all levels of society.

5.2.1 Core Methodology and Structure

The initiative mandates a meticulously strategized approach to training; this strategy involves providing clear methods to make the training easily accessible and widely publicized to diverse societal segments.

Training delivery is structured into two cycles, incorporating both online sessions and in-person training sessions to cater effectively to different learning preferences and technical expertise levels. This approach ensures that all training initiatives are thorough, scheduled and results-oriented.

5.2.2 Training Plan Content Integration and Objectives

The training content focuses on cultivating a culture that prioritizes secure online behaviors.

Key areas specified for inclusion in the training plan align to the objectives of the CYBERWISE module and learning goals:

1. **Cyber Hygiene and Behavioral Change**: content will aim to strengthen foundational cyber hygiene, enabling users to recognize social engineering attempts and avoid risky behaviors. This directly addresses common human-factor vulnerabilities identified in the project's foundational work.
2. **Immersive Learning and Gamification**: the plan incorporates methodologies that utilize gamification techniques (T6.4/D6.2 [11]) to make learning impactful and encourage sustained engagement; this includes experiential, hands-on scenarios in a safe environment.
3. **Proactive Assessments**: the plan integrates insights derived from proactive assessments, such as the planning and execution of simulated social engineering phishing campaigns (T6.2), turning these tests into learning opportunities to evaluate and improve user vulnerability.
4. **Policy and Compliance**: training reinforces the internalization of organizational policies and compliance requirements, complementing the governance features provided by the CYBERGOPLUS module.
5. **Technical Skill Development**: the plan specifies incorporating practical training modules, such as the Practical Software Vulnerability Assessment using SNYK (Section 5.8), to empower personnel involved in software development, QA, and IT operations to efficiently identify and mitigate security flaws early, adhering to modern DevSecOps standards.

The overall training program integrates practical content across the entire CYberSynchrony ecosystem, emphasizing hands-on learning and relevance to real-world threats. It emphasizes customization and high relevance by integrating project tools and addressing specific user roles and risk profiles, which is crucial for maximizing impact.

© CYberSynchorny, 2024

Content development considers human factors such as attackers' profiles, security awareness level/culture, and behavioral, psychological, societal, and economic characteristics of the enterprise.

5.2.3 Measurement and Effectiveness

A crucial aspect specified within the training planning is the establishment of feedback mechanisms, such as quizzes, to measure the training effectiveness systematically.

Feedback collected will be leveraged to continuously refine the modules, including elements such as increasing interactive engagement levels and providing detailed code examples for practical analysis; this iterative process ensures the strategies remain relevant and malleable to evolving cyber scenarios.

The efficacy of the plan will be measured against specific key performance indicators (KPIs), including aiming to increase cybersecurity training coverage to 85% (KPI 8.1) and achieving a cybersecurity awareness index of 75 (KPI 8.2). Measurement ensures that training is effective, results-oriented, and that the overall campaign effectiveness can be measured using data analytics.

5.3 Real word social engineering

5.3.1 Introduction and Strategic Objectives

This section details the planning and design of a critical training and awareness exercise focusing on social engineering and more specifically the use of phishing emails using the **Gophish** open-source platform. The human element remains the most critical, yet often least addressed, vulnerability in organizational cybersecurity. For any company, effective security awareness must evolve beyond static training to include realistic, experiential learning. This report details the planning, execution, and outcomes of a controlled, simulated social engineering phishing campaign conducted within such an organization. The primary objective was to leverage Social Engineering as a Cybersecurity Awareness Tool to transform passive policy knowledge into active, reflexive cyber-resilience among employees.

This campaign is a key component in fulfilling the broader objectives of the CYberSynchrony project:

- **Task 6.1 (CYberSynchrony Platform and Pervasive Cybersecurity Awareness Platform):** The educational parts of the campaigns will be available to the **Cybersecurity Awareness Platform** to ensure high accessibility to all interested parties.
- **Task 6.2 - Proactive Social Engineering Assessments:** Targeted campaigns will simulate real-world social engineering threats focusing on phishing. The objective is not just to test users but to evaluate their responses, gauging their vulnerability to such attacks. These assessments help identify areas where further training and awareness are essential for the targeted organizations. It's a proactive approach that both tests and educates simultaneously. These assessments are vital for understanding and addressing the human aspect of cybersecurity.
- **Task 6.4 (Immersive Gamified Cybersecurity):** The campaign's methodology incorporates experiential learning and follow up lectures, which apply the principles of gamification to practical cybersecurity challenges in a controlled and safe environment.

By providing direct, practical experience with a leading open access tool like Gophish, followed by training lectures, these campaigns aim to transform phishing exercises from a compliance checkpoint into an integrated, continuous practice.

© CYberSynchorny, 2024

5.3.2 Methodology

The project adhered to a structured methodology, commencing with a rigorous Planning and Scoping phase, essential for a high-stakes corporate environment. We first conducted a thorough market analysis of available Phishing Tools in the Market (e.g., open-source frameworks, commercial platforms) to select a solution offering high customizability, detailed logging, and secure infrastructure isolation. This informed the Tooling and Infrastructure Setup, ensuring the campaign infrastructure could mimic real-world threat actors while remaining completely isolated from production systems. Critically, extensive effort was dedicated to Legal and Ethical Considerations. This involved securing formal approval from executive leadership, establishing a clear opt-out mechanism, and ensuring all collected data was anonymized and used strictly for educational and aggregate reporting purposes.

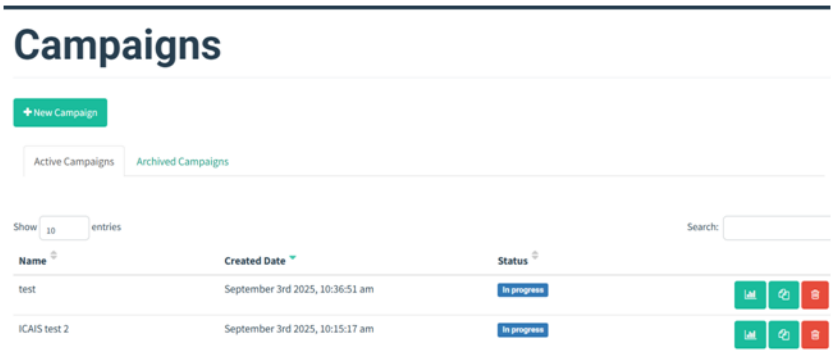


Figure 15. Gophish campaign design internal system

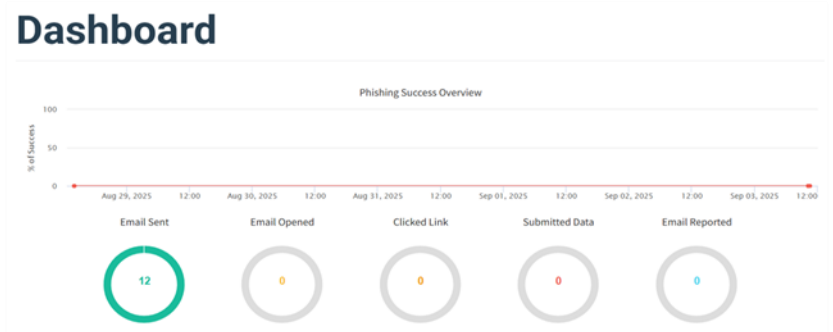


Figure 16. Gophish statistics about the emails sent during the campaign (sent, opened, clicked, submitted data, reported

The subsequent Campaign Setup involved intricate Attack Vector Development and Execution focused on a single, highly effective lure. For the first campaign, we designed a sophisticated email targeting the common institutional practice of securing external funding. The simulated email utilized specific European Commission branding, contact details, and a plausible subject line ("Tender application call number EU.6/43/2025") to create extreme urgency and legitimacy. Sent from a spoofed address the email falsely notified the recipient that their tender status was "pending," requiring immediate confirmation. The campaign was executed in phased waves, targeting a statistically significant sample group across departments involved in grant writing and external collaboration. Success metrics are focused on two key actions: the click rate (clicking the embedded link) and the credential submission rate (entering credentials on the malicious landing page).

© CYberSynchorny, 2024

ecas.ec.commission@europeaa.eu
προς εγώ

Dear tender applicant,

There has been an update regarding your call number **EU.6/43/2025**. The call status has changed to pending. To continue the process, the pending status requires your confirmation in the eProcurement Portal

Use the portal https://tenders.ec-europa.eu/apply to examine your tender application.

Kind regards,

Thomas v'd Berg
Head of Unit

European Commission
DG Grow
Unit C.2 – European Citizen Action Service
MADO 26/003
B-1049 Brussels/Belgium
+32 2 298 67 16

European Union

Figure 17. Phishing Email from the 1st campaign

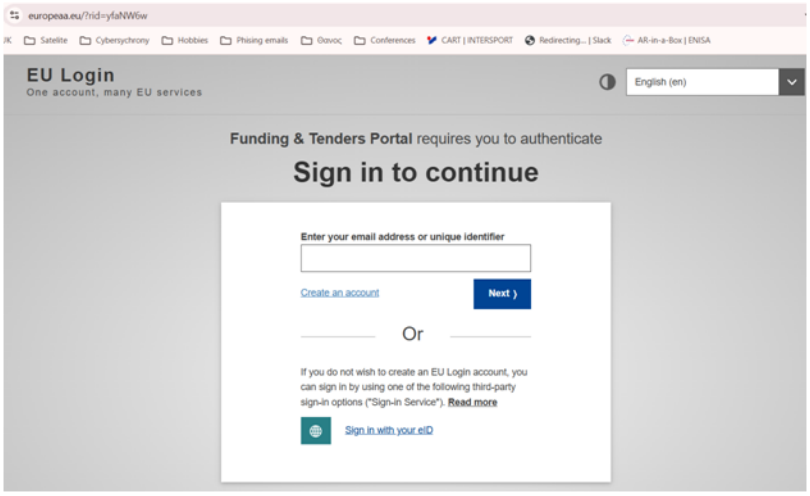


Figure 18. The loading webpage where the victim is asked to add their information

© CYberSynchorny, 2024

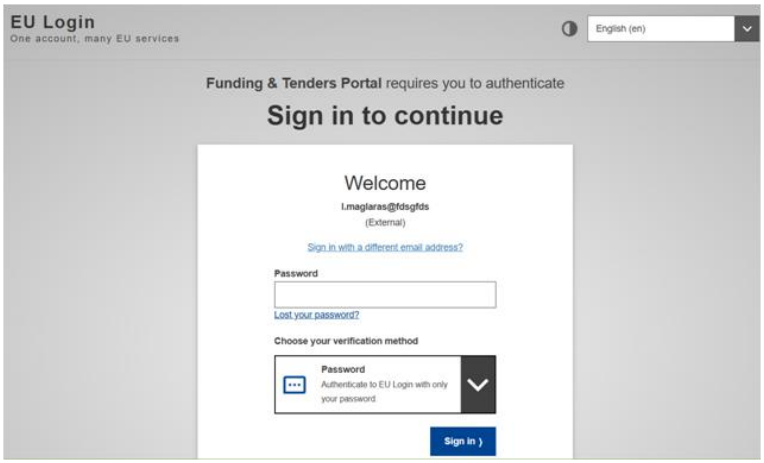


Figure 19. The victim is asked to add username and password, but this information is not captured in Gophish

5.3.3 Next steps and future work

The Analysis, Reporting, and Education phase will begin immediately post-execution. Our results provide a quantitative baseline for the organization's vulnerability, identifying the specific psychological triggers exploited by the tender lure and quantifying the overall organizational risk profile. This raw data will be translated into a comprehensive report for leadership, detailing aggregate vulnerability scores, time-to-first-click metrics, and observed behavioral patterns.

The campaign will be followed by mandatory, immediate, and targeted remediation. All employees who participate in the test will receive a non-punitive training module focusing on identifying highly-spoofed government/tender communications and will be presented with an organization-wide presentation that deconstructed the campaign and reinforced best practices for verifying external communications.

Finally, these campaigns can be used for establishing a framework for Continuous Improvement and Iteration. The baseline metrics gathered serve as a benchmark for future campaigns, adapting the complexity of the lures based on the organization's current performance. This iterative approach allows the organization to measure the sustained impact of the training, adapt to emerging threat trends like highly targeted spear-phishing, and foster a proactive, security-conscious culture.

This report demonstrates that ethical, simulated social engineering, even with a single, highly tailored vector, is an invaluable, high-fidelity instrument for hardening a research-centric organization from the inside out. For this first version, the focus is on analyzing the available phishing tools in the market, choosing the open-source tool that could effectively be used in our campaigns (Gophish), performing the planning, setting up the infrastructure, dealing with Legal and Ethical Considerations and finally running the 1st campaign (Attack Vector Development and Execution).

For the subsequent version, the project will report on the execution and performance of all the campaigns (or the iteration of the 1st campaign). This report will incorporate key metrics and lessons learned gathered from the campaigns. Crucially, feedback will be leveraged to continuously refine the attack vector development and campaign design.

© CYberSynchorny, 2024

5.4 Integration of gamified cybersecurity

The CYberSynchrony platform integrates gamified cybersecurity through the **Cybergate training environment**, a modular, simulation-based learning system that combines offensive and defensive cyber perspectives within a game-driven ecosystem. This integration within the **CYBERWISE module** operationalizes the concept of *learning-by-doing*, allowing users to experience cyber incidents from both attacker and defender viewpoints in a controlled environment. The goal is to transform traditional awareness into active competence, bridging theoretical understanding with hands-on decision-making under realistic conditions.

5.4.1 Cybergate's Learning Methodology

Cybergate employs a dual-phase methodology that mirrors the dynamics of real cyber operations. Each mission unfolds through structured gameplay cycles (A *tack*, *Defense*, *Dome*, and *Outcome). The game is* aligned with industry-standard frameworks such as **MITRE ATT&CK** for adversarial behavior and **MITRE D3FEND** for defensive actions. During the *Attack Phase*, trainees perform simulated offensive actions through interactive mini-games (e.g., phishing, credential theft, lateral movement), helping them internalize adversarial logic and tactics. In the subsequent *Defense Phase*, teams collaboratively allocate limited defensive resources, such as multi-factor authentication, segmentation, or privilege control, to mitigate the simulated attack chain. This structure reinforces strategic prioritization, teamwork, and time-sensitive response under resource constraints. The *Dome Phase* then challenges players to deploy one decisive, organization-wide measure (e.g., network isolation or cross-sector information sharing), teaching large-scale mitigation thinking. Finally, the *Outcome Phase* provides a visual replay and analytical debrief, mapping the players' defensive effectiveness against optimal real-world responses.

5.4.2 Alignment with CYBERWISE Objectives

Within CYBERWISE, gamified cybersecurity acts as the experiential pillar of awareness-building. Traditional awareness modules (such as phishing simulations and HRM-based behavioral analytics) focus on recognizing human risk; Cybergate extends this by **immersing users in the adversarial logic of cyberattacks**, thereby strengthening cognitive resilience. Its design aligns directly with WP6 objectives by fostering critical reasoning, risk anticipation, and collaboration. The integration also complements **HRM analytics**, allowing performance data from gameplay (e.g., decision quality, reaction speed, and mitigation effectiveness) to feed into behavioral risk metrics. This closes the loop between awareness assessment and skill development, making CYBERWISE both diagnostic and formative.

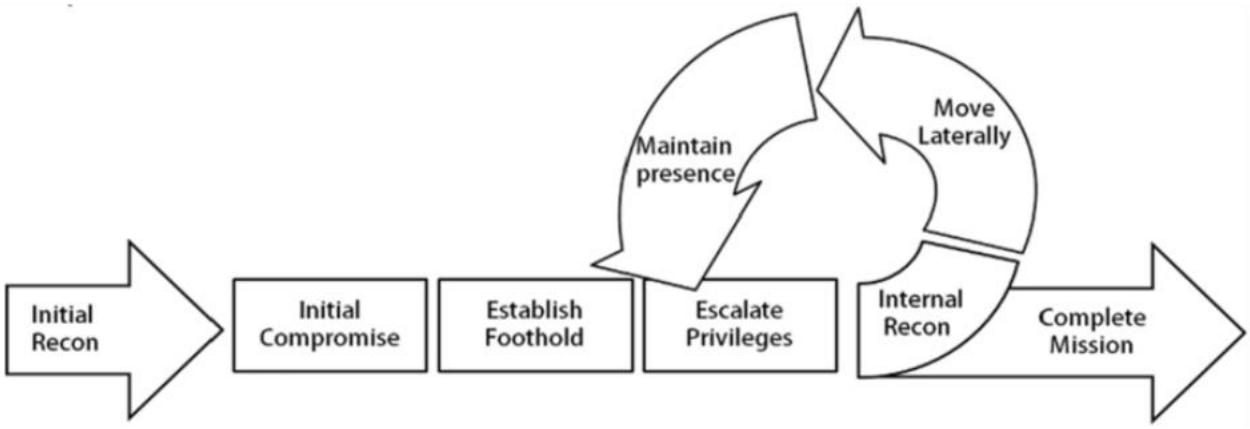


Figure 20. Typical adversarial attack lifecycle

© CYberSynchorny, 2024

5.4.3 Technical Integration

From a technical perspective, Cybergate is deployed as a **modular component within the CYBERWISE environment**, interoperable with the broader CYberSynchrony ecosystem through secure APIs. It can authenticate users via the platform's unified identity management (Keycloak) and exchange anonymized behavioral metrics with HRM and CSPA components. Missions and progression data are stored securely and optionally anchored via BYTE's Blockchain Integrity Validator to ensure data immutability and verifiable learning outcomes. The gamified engine itself operates in a sandboxed mode: all simulated attacks are purely emulated within the game layer, ensuring no exposure to actual systems while preserving the realism of operational decision-making. This guarantees compliance with CYberSynchrony's privacy, ethics, and data protection requirements.

Ultimately, the Cybergate integration transforms CYBERWISE from a passive awareness tool into an **active cyber range for societal training**, where users engage in meaningful, data-driven experiences that enhance their readiness for real-world threats. By embedding gamified simulations alongside AI-based human risk analytics, CYberSynchrony sets a new standard in awareness-driven cybersecurity capacity building, turning awareness into preparedness through immersive gamification.

5.5 Integration of cybersecurity policy toolkit

The purpose of the Cybersecurity Policy Toolkit is to provide a structured, standardized set of cybersecurity policies and supporting tools aligned with the CYberSynchrony's objectives and the needs of stakeholders involved in cybersecurity governance and operations.

The CYberSynchrony Policy Toolkit is a combination of operational cybersecurity policy tools applied by the Digital Security Authority of Cyprus (DSA) and the Hellenic National Intelligence Service for hands-on management of national cyber-security. The Tools are as follows:

Tool 1 - Information Security Strategy

Tool 2 - Information Security Governance Policy

Tool 3 - Risk Management Policy & Methodology

Tool 4 - Asset Management Policy

Tool 5 - Supplier and Third-Party Management Policy (TPS)

Tool 6 - Access Control Policy

Tool 7 - Vulnerability and Patch Management Policy

Tool 8 - Network Security Policy

Tool 9 - Systems & Applications Security Policy

Tool 10 - Data Protection Policy

Tool 11 - Acceptable Use Policy

Tool 12 - Human Resources Security, Training & Awareness Policy

Tool 13 - Physical Security Policy

Tool 14 - Incident and Event Management Policy

Tool 15 - Business Continuity and Disaster Recovery Policy

The fifteen (15) Tailored Cybersecurity Policy Tools are adaptable and ready/easy-to-use, enabling security officers to obtain, adjust and update their policies to the dynamic nature of the cyber landscape, ensuring the

© CYberSynchorny, 2024

security of their organization. All policies are in compliance with national, EU and international standards and legal frameworks.

Any Organizations can, easily and directly, adopt and implement the present CYberSynchrony Policy Tools by inserting their individual name.

5.6 Sector-Specific Cybersecurity Harmonization

This section details the implementation of **Task 6.6 "Sector-Specific Cybersecurity Harmonization"** within the Cyber Awareness Platform. The platform bridges the gap between diverse industry sectors by advocating for a consistent cybersecurity framework while acknowledging the unique operational realities of **Healthcare**, **Public Administration**, and **Academia**. To achieve this, the platform employs a "Harmonized Core, Sectoral Overlay" content strategy, grounded in the four cross-sector principles defined in the project's compliance framework (D6.4 [12]).

5.6.1 The Harmonization Model

The platform addresses the fragmentation of cybersecurity practices identified in the D2.2 [6] landscape analysis by establishing a unified baseline of competence. While the *delivery mechanism* (the platform itself) is standardized, the *learning objects* are tagged and filtered to ensure relevance.

- **Unified Baseline:** All users, regardless of sector, receive training on universal cyber hygiene topics such as Password Security, Phishing Recognition, and GDPR basics. This ensures a consistent minimum standard across the European ecosystem.
- **Sectoral Adaptation:** Advanced modules are tailored to specific domains. For instance, a "Data Privacy" module for healthcare focuses on patient records (EHR), while the same module for academia focuses on intellectual property and research data integrity.

5.6.2 Application of Harmonised Principles to Training

To ensure the educational content directly supports the organization's compliance and security posture, the training curriculum is structured around the four harmonized principles derived from the compliance oversight tasks:

Risk-Driven Governance Awareness: Training modules under this principle focus on ensuring users understand their specific role in the security hierarchy.

- **Harmonization:** All sectors are trained in the importance of policy adherence and the "Human-as-a-Firewall" concept.
- **Sector Specificity:**
    - *Healthcare:* Training emphasizes the critical risk to patient safety and the role of medical staff in protecting IoMT devices.
    - *Governance:* Training focuses on the accountability of public servants and the risks of essential public services.

Asset-Centric Protection Training: Users are trained to identify and protect the specific assets relevant to their daily work.

- **Harmonization:** Standard training on device locking, clear desk policies, and reporting lost equipment.
- **Sector Specificity:**
    - *Academia:* Modules address the security of "Open Science" environments and the protection of student PII on personal devices (BYOD).

© CYberSynchorny, 2024

- *Public Admin:* Modules focus on the integrity of public registries and the protection of citizen digital identities.

Continuous Threat Detection Skills: The platform enhances human sensor capabilities by training users to recognize sector-specific attack vectors.

- **Harmonization:** Universal recognition of social engineering cues (urgency, authority, scarcity).
- **Sector Specificity:**
  - *Healthcare:* Simulations replicate "Medical Emergency" phishing lures.
  - *Governance:* Simulations mimic "Government Grant" or "Tax Authority" scams.

Resilience and Recovery Protocols: Training ensures staff know how to act during and after an incident to support organizational recovery.

- **Harmonization:** Standard procedures for incident reporting and maintaining communication during outages.
- **Sector Specificity:**
  - *Healthcare:* Training on manual fall-back procedures for patient care during ransomware outages.
  - *Public Admin:* Protocols for maintaining essential utility services during cyberattacks.

5.6.3 Technical Implementation of Harmonization

The platform utilizes metadata tagging to enforce this harmonization dynamically.

- **Content Tagging:** Every learning object is tagged with a sector_id (e.g., HEALTH, GOV, EDU) and a competency_level (Basic, Advanced).
- **Dynamic Learning Paths:** When a user logs in, the Keycloak Identity Manager provides their organizational attributes. The platform's recommendation engine uses these attributes to automatically curate a learning path that combines mandatory harmonized modules with relevant sector-specific content, ensuring efficient and targeted skill development.

5.7 Integration of compliance monitoring

Integration of the Cyber Awareness Platform with the compliance monitoring mechanisms is defined under Task 6.7 and detailed in Deliverable 6.4. The objective of this integration is to ensure that awareness activities, training performance, and user behaviour contribute meaningful evidence to the organization's real-time compliance posture. The integration enables CYBERGOPLUS to incorporate user-centric metrics, derived from awareness content, training modules, and social-engineering exercises, into the broader auditing and oversight processes of the CYberSynchrony ecosystem.

5.7.1 Role of the Cyber Awareness Platform in Compliance Monitoring

D6.4 [12] establishes that compliance monitoring relies on multiple categories of evidence: **policy**, **implementation**, and **operational/runtime evidence.** Within this model, the Cyber Awareness Platform primarily contributes **human-centric operational evidence**. This includes:

- **Training completion and participation records** from CYBERWISE learning modules (linked to Awareness & Training controls)
- **Performance outcomes** from immersive simulations such as Cybergate missions (e.g., phishing resilience, defensive behaviours)
- **User authentication behaviour** through Keycloak logs (e.g., MFA usage, role assignments, authentication success/failure)

© CYberSynchorny, 2024

These datasets support verification of controls mapped to NIS2 and ISO 27001, such as TA2 (Information Security Awareness), IAM4 (Strong Authentication), and policy acknowledgement requirements.

5.7.2 Data Flows and Integration Interfaces

Compliance monitoring requires the Cyber Awareness Platform to expose structured information that CYBERGOPLUS can ingest as audit evidence. D6.4 defines the **integration specifications**, identifying the Cyber Awareness Platform as a southbound data source for identity-related and training-related evidence. Integration relies on:

- **Keycloak Identity Manager** for verifying MFA enforcement and RBAC policies, using OIDC and JWT tokens as evidence inputs for IAM controls
- **Training and simulation output feeds** that provide mission scores, training timestamps, and performance metrics required for TA2 compliance indicators (e.g., Training Coverage, Simulation Success Rate)
- **REST APIs and JSON-based evidence objects** aligned with the schemas in Annex I of D6.4 (e.g., identity verification schema, compliance-status JSON)

Through these interfaces, the Cyber Awareness Platform participates in the **evidence ingestion pipeline** that also includes Wazuh, Suricata, the Evidence Integrity System, and SNYK.

5.7.3 Alignment with CYBERGOPLUS Compliance Domains

D6.4 defines six assessment domains for real-time oversight (Governance, Asset Security, Identity Control, Threat Protection, Vulnerability Status, Resilience) and explicitly notes the role of training metrics in verifying Awareness & Training (TA2) compliance. The Cyber Awareness Platform contributes directly to the following:

Awareness & Training (TA2): (a) Mission completion logs and simulation scores from Cybergate (D6.2 [11]) serve as evidence of training effectiveness, (b) completion of platform modules validates mandatory awareness requirements.

Identity & Access Management (IAM4): Authentication logs and MFA fields from Keycloak confirm enforcement of strong access controls.

Governance & Policy Alignment: Policy acknowledgment records and metadata (e.g., acceptance timestamps) feed into Governance controls under D6.3 [9].

These contributions allow CYBERGOPLUS to perform quantitative monitoring—such as Training Coverage, MFA Coverage, and Policy Compliance Rate—using metrics defined in D6.4's compliance indicator map.

5.7.4 Contribution to Real-Time Oversight

Real-time compliance oversight depends on continuously updated technical evidence rather than static assessments. The Cyber Awareness Platform supports this by providing:

- **Real-time updates of user activity and training status**, improving detection of awareness gaps that may present compliance risk.
- **Immediate reporting of simulation outcomes**, allowing CYBERGOPLUS to identify human-factor vulnerabilities (e.g., susceptibility to phishing) and track improvements over time.
- **Continuous identity-related telemetry** from Keycloak, enabling instant visibility into authentication anomalies or deviations from access policies.

This ensures that human-factor performance—traditionally difficult to measure—is fully incorporated into the compliance engine.

© CYberSynchorny, 2024

5.7.5 Architectural Positioning

CYBERGOPLUS is the **central compliance engine**, while the Cyber Awareness Platform acts as an upstream contributor of behavioral and identity evidence. According to the architecture description in D6.4, CYBERGOPLUS integrates:

- data ingestion pipelines from awareness content (North–South API flow),
- Keycloak-based identity evidence,
- runtime operational logs from CYBRITE and CROSS-CORE, and
- vulnerability evidence from SNYK

This confirms the platform's positioning as a key **evidence provider**, enabling harmonized oversight across training, policy, risk, and technical controls.

5.8 Training Module: Practical Software Vulnerability Assessment using SNYK

5.8.1 Introduction and Strategic Objectives

This section details the planning and design of a critical training module focusing on Vulnerability Assessment of Software Applications through the practical integration and use of the SNYK (Snyk.io) security analysis platform. As modern cyber threats frequently target weaknesses embedded within application code and dependencies, a proactive approach to vulnerability assessment is essential. This module is designed to empower personnel involved in software development, QA, and IT operations to efficiently identify and mitigate security flaws early in the development process, thereby adhering to modern DevSecOps standards.

This training module is a key component in fulfilling the broader objectives of the CYberSynchrony project:

Task 6.1 (CYberSynchrony Platform and Pervasive Cybersecurity Awareness Platform): The module will be deployed through the user-friendly CYberSynchrony platform, utilising interactive elements and multimedia content (such as slides and quizzes) to ensure high accessibility and cater to a diverse range of technical expertise among users.

Task 6.4 (Immersive Gamified Cybersecurity): The module's methodology incorporates experiential learning and hands-on scenarios, which apply the principles of gamification to practical cybersecurity challenges in a controlled and safe environment.

By providing direct, practical experience with a leading industry tool like SNYK, this module aims to transform vulnerability assessment from a compliance bottleneck into an integrated, continuous practice.

5.8.2 Core Learning Objectives and Expected Outcomes

The module is structured to provide participants with the skills necessary to perform effective, continuous software vulnerability assessment. Upon successful completion of this training, participants will gain the necessary skills and knowledge to:

Define and explain fundamental software security principles, the role of code auditing and Quality Assurance (QA) in reducing risk, and the importance of DevSecOps security methodologies.

Identify and analyze common software vulnerabilities, with a focus on understanding and mitigating those outlined in the OWASP Top 10 risks.

Utilize the SNYK Platform to perform robust automated Static Code Analysis on both application source code and its third-party components (dependencies).

Apply Code Auditing Techniques during the software development lifecycle to systematically uncover and document security weaknesses detected during code reviews.

© CYberSynchorny, 2024

Prioritize and resolve identified vulnerabilities efficiently by interpreting SNYK's risk classification scores and implementing proven remediation and fixing practices.

5.8.3 Pedagogical Methodology and Content Delivery

This module employs a highly practical and engaging pedagogical strategy to ensure deep understanding and skill transference.

A. Instructional Delivery and Structure

The learning experience is structured around a blend of directed instruction and hands-on practice:

Expert-Led Instruction: Focused sessions convey essential theoretical concepts, such as secure coding principles and DevSecOps models.

Lab-Based Exercises (Hands-on): Participants will engage in scenario-based, practical activities in a safe environment, where they will apply their knowledge directly using the security tools.

Live Demonstrations: Real-time visual presentations will showcase the effective integration and operation of the SNYK tool, demonstrating the end-to-end vulnerability assessment workflow.

Interactive Sessions: Group discussions and collaborative exercises will be used to foster a dynamic learning environment, encouraging the sharing of practical experiences and knowledge.

B. Training Resources and Tools

The training content is designed to be comprehensive and readily accessible:

Interactive Lecture Slides: Engaging visual presentations used to clearly explain key security concepts and demonstrate practical scenarios.

Detailed Lab Manuals: Step-by-step instructions that guide participants through the technical exercises, specifically detailing the functionalities of the SNYK platform.

Case Studies: Real-world examples will be used to illustrate the consequences of software vulnerabilities and demonstrate successful assessment and mitigation strategies.

The SNYK.io tool is the core component of the module, serving as the central web-based platform for hands-on, collaborative vulnerability analysis and secure code review.

5.8.4 Assessment Strategy and Implementation Planning

A continuous and comprehensive assessment strategy is employed to evaluate skill acquisition and knowledge retention.

A. Assessment Components

The module’s evaluation process is divided into formative and summative parts:

Formative Assessments: These activities include a pre-assessment to determine baseline knowledge, followed by intermittent quizzes and interactive exercises integrated throughout the module to ensure continuous knowledge reinforcement.

Summative Assessment (Final Project): The culminating exercise is a Simulated Code Auditing Project. Participants will be required to use the SNYK platform to analyze a provided application codebase, accurately identify and prioritize security weaknesses, and submit a detailed remediation plan. This practical task serves to measure the participants' ability to perform a technical vulnerability assessment in a controlled environment.

© CYberSynchorny, 2024

B. Planning for Execution and Feedback

For this first version, the focus is on establishing strategic design. Looking ahead to execution, the delivery will prioritize creating adequate lead time between scheduling and delivery to ensure smooth logistical coordination and participant preparation.

For the subsequent version, the project will report on the execution and performance of this module. This report will incorporate key metrics and lessons learned gathered from initial training rounds. Crucially, feedback will be leveraged to continuously refine the module, focusing on aspects such as providing detailed code examples for practical SNYK analysis and increasing the interactive engagement level during sessions to further enhance practical skill development.

© CYberSynchorny, 2024

6 Conclusions

Deliverable 6.1 presents the first version of the Cyber Awareness Platform, establishing the foundations for human-centric cybersecurity within the CYberSynchrony ecosystem. This version consolidates the platform's design principles, learning objectives, content pipeline, and technical architecture, and introduces the initial set of training, simulation, and harmonisation components that will support continuous awareness development across all participating sectors.

The deliverable demonstrates significant progress toward CYberSynchrony's goal of harmonizing people, processes, and technology. The platform now includes a modular system architecture built on Kubernetes, aligned with the integration blueprint and supported by secure IAM mechanisms through Keycloak. It incorporates a complete deployment workflow, containerized microservices, automated certificate management, and CI/CD integration, enabling consistent and scalable delivery across environments. This release also formalizes the role of surrounding modules, such as CYBERWISE, CYBERGOPLUS, CYBERRA, CYBRITE, and CYRESCUE, in shaping awareness content, behavioral insights, and governance alignment.

From a content perspective, the document outlines the structured content pipeline, the initial training plan, a complete phishing assessment using Gophish, integration of gamified scenarios through the Cybergate environment, and the first hands-on technical training module using SNYK. These components ensure that awareness activities are grounded in real operational risks, threat intelligence, and the sector-specific needs identified earlier in the project. The deliverable also presents the first model for sector harmonization and establishes the link between awareness activities and compliance evidence, supporting real-time oversight through integration with the compliance monitoring framework described in D6.4.

This version of the platform adds business value by enabling organisations to build cybersecurity capabilities that reflect their operational realities, measure user performance, and provide evidence for NIS2- and ISO-aligned compliance controls. It further demonstrates how awareness, training, governance, and monitoring can be delivered coherently through a single ecosystem.

The work completed in D6.1 also highlights several best practices, such as embedding awareness within operational data flows, adopting immersive and scenario-driven training methods, and applying harmonised sector models, which will guide future enhancements. The platform will evolve further in subsequent releases, particularly D6.5, which will include expanded LMS capabilities, additional gamified missions, enriched behavioural analytics, further integration with compliance monitoring, and refined sector-based learning pathways.

In summary, D6.1 delivers the conceptual, architectural, and operational foundations for the Cyber Awareness Platform and sets a clear trajectory for its evolution. It marks a major milestone in operationalising human-centric cybersecurity within CYberSynchrony and provides the basis for continued development toward a mature, scalable, and integrated awareness solution.

© CYberSynchorny, 2024

7 References

[1] "Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union (NIS2 Directive)," [Online]. Available: https://nis2directive.eu/.

[2] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation)," [Online]. Available: https://gdpr-info.eu/.

[3] "Deliverable D8.1 – Integration Blueprint and Use Case Portfolio".

[4] "Keycloak – Open Source Identity and Access Management," [Online]. Available: https://www.keycloak.org/.

[5] "Deliverable D2.1 – Cybersecurity landscape and threat analysis, end-user requirement".

[6] "Deliverable D2.2 – Cybersecurity landscape and threat analysis, end-user requirement analysis and use case definition – Final version".

[7] "Web Content Accessibility Guidelines (WCAG) 2.1, World Wide Web Consortium (W3C)," [Online]. Available: https://www.w3.org/TR/WCAG21/.

[8] "Accessible Rich Internet Applications (WAI-ARIA) Authoring Practices, W3C," [Online]. Available: https://www.w3.org/WAI/ARIA/apg/.

[9] "Deliverable D6.3 – Cybersecurity Policy Toolkit v1".

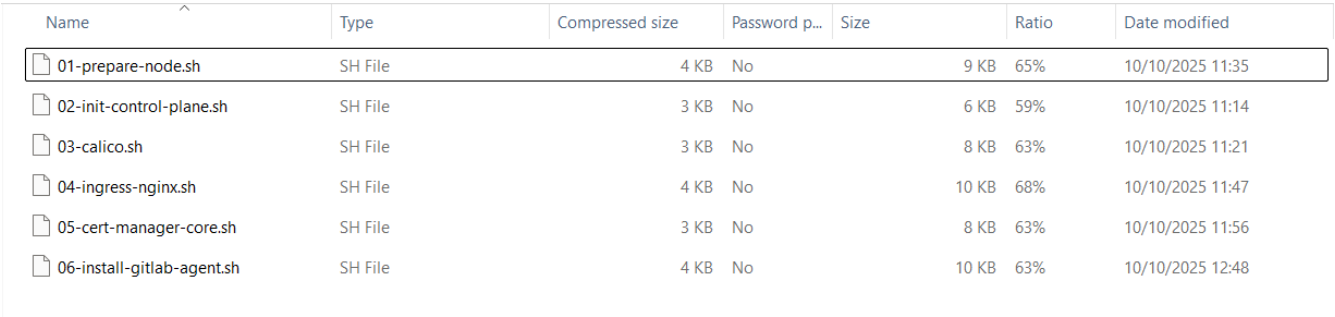
[10] "European Cybersecurity Skills Framework (ECSF), ENISA," [Online]. Available: https://www.enisa.europa.eu/topics/skills-and-competences/skills-development/european-cybersecurity-skills-framework-ecsf.

[11] "Deliverable D6.2 – Gamified Training Modules v1".

[12] "Deliverable D6.4 – Compliance Monitoring v1".

© CYberSynchorny, 2024

Annex I: Environment variable reference (per script)



Name	Type	Compressed size	Password p...	Size	Ratio	Date modified
01-prepare-node.sh	SH File	4 KB	No	9 KB	65%	10/10/2025 11:35
02-init-control-plane.sh	SH File	3 KB	No	6 KB	59%	10/10/2025 11:14
03-calico.sh	SH File	3 KB	No	8 KB	63%	10/10/2025 11:21
04-ingress-nginx.sh	SH File	4 KB	No	10 KB	68%	10/10/2025 11:47
05-cert-manager-core.sh	SH File	3 KB	No	8 KB	63%	10/10/2025 11:56
06-install-gitlab-agent.sh	SH File	4 KB	No	10 KB	63%	10/10/2025 12:48

01-prepare-node.sh
K8S_CHANNEL, EBOS_CA_SRC, EBOS_CA_NAME, EBOS_IP, EBOS_HOSTS_INPUT, EBOS_REGISTRY_HOST, EBOS_REGISTRY_PORT, PAUSE_IMAGE.

02-init-control-plane.sh
CLUSTER_NAME, CONTROL_PLANE_ENDPOINT, POD_CIDR, SERVICE_CIDR, API_SANS_EXTRA, OUT_DIR.

03-calico.sh
CALICO_VERSION, POD_CIDR, ROLLOUT_TIMEOUT, FORCE_IPTABLES, UNTAINT_CONTROL_PLANE.

04-ingress-nginx.sh
NS, RELEASE, SVC_TYPE, HTTP_NODEPORT, HTTPS_NODEPORT, INGRESS_CLASS, CLASS_DEFAULT, WATCH_INGRESS_WITHOUT_CLASS, ENABLE_METRICS, CHART_VERSION, ROLLOUT_TIMEOUT.

05-cert-manager-core.sh
NS, RELEASE, CHART_VERSION, CREATE_HTTP01_ISSUERS, LE_INGRESS_CLASS, LE_EMAIL.

06-install-gitlab-agent.sh
AGENT_NAME, NAMESPACE, IMAGE_TAG, KAS_HOST, KAS_PORT, KAS_SCHEME, KAS_CA_CERT, AGENT_TOKEN, PATCH_COREDNS_HOSTS, HOSTS_IP, HOSTS_NAMES, RUN_DNS_TEST.

© CYberSynchorny, 2024

Annex II: Sample Manifests & Snippets

Ingress with TLS (cert-manager)

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: demo
  annotations:
    kubernetes.io/ingress.class: nginx
    cert-manager.io/cluster-issuer: letsencrypt-prod
spec:
  tls:
    - hosts: [ demo.example.com ]
      secretName: demo-tls
  rules:
    - host: demo.example.com
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: demo-svc
                port:
                  number: 80
```

Agent config with GitOps:

```yaml
# .gitlab/agents/<AGENT_NAME>/config.yaml
ci_access:
  projects:
    - id: mygroup/myproject
      default_namespace: default

gitops:
  manifest_projects:
    - id: mygroup/infra
      default_namespace: default
      paths:
        - glob: /clusters/cybersynchroni/**/*.yaml
```

© CYberSynchorny, 2024