Common Cloud Controls (CCC)

Workshop - Monday April 12th

#OSinFinance | #OSFF2026

Disclaimer

“Any views or opinions expressed in this presentation are those of the presenter and not necessarily represent the view and opinions of the employer, its ownership, management or employees.”

About us

Eddie Knight
Founder
Revanite

eknight@revanite.io

Maxime Coquerel
Principal Cloud Security Architect
RBC

maxime.coquerel@rbc.com

Agenda

- 13:00 - 13:30 : Welcome + Networking
- 13:30 - 15:00 : Interactive Introduction
- 15:00 : Break (15 min)
- 15:30 - 17:00 Hands on

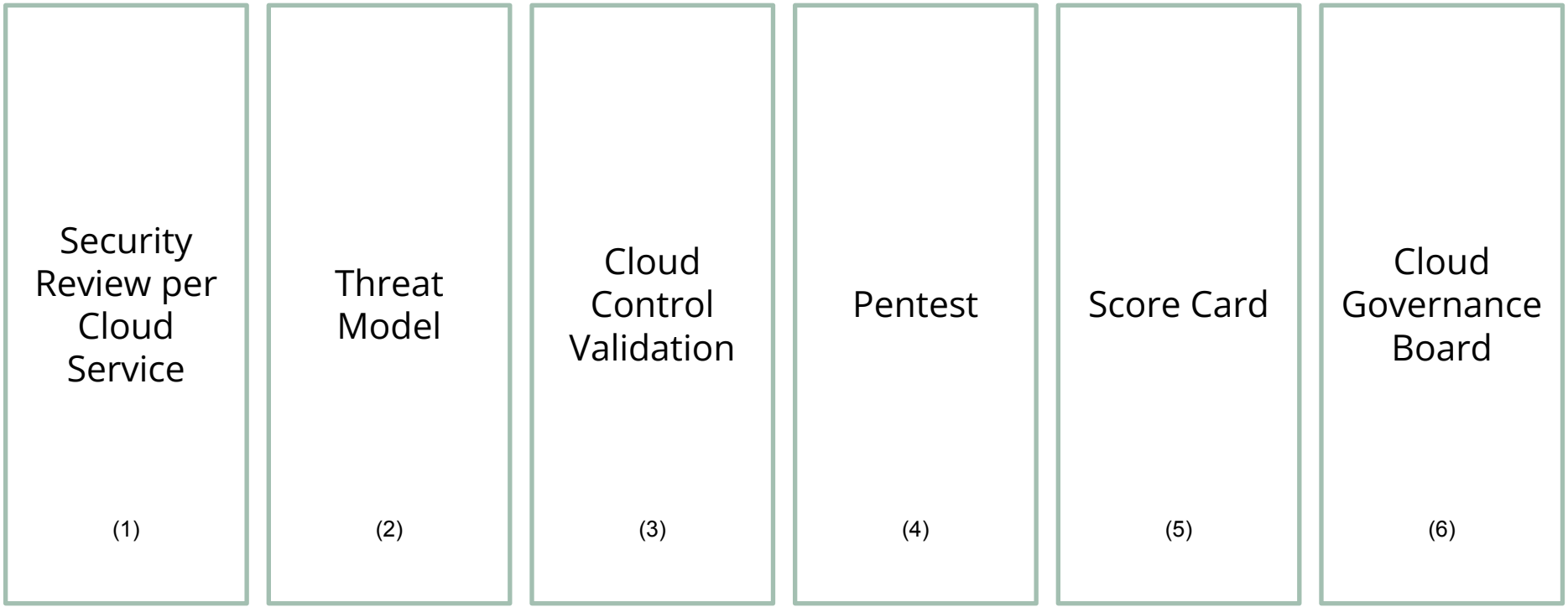
Discussion (Interactive)

- "What guidance do you provide for defining policies and controls?"
- "How do you connect requirements to evaluations?"

Discussion (Interactive)

- "How do you enforce least privilege today?"
- "Is your approach consistent across clouds?"

Cloud Security Framework

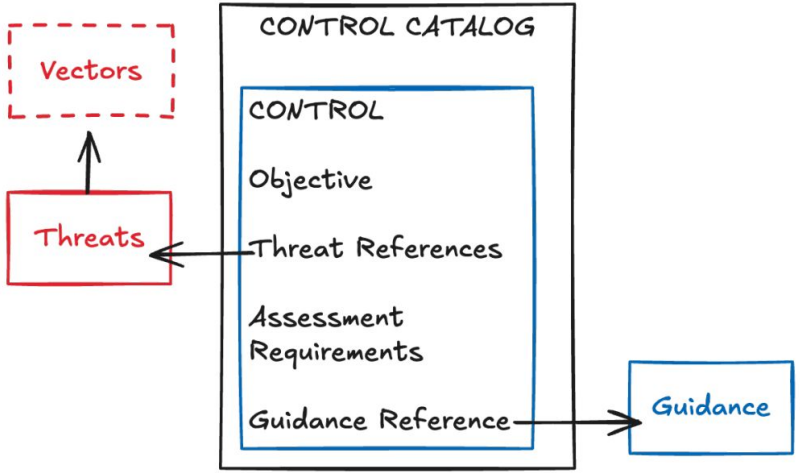


What is FINOS Common Cloud Controls (CCC)?

FINOS Common Cloud Controls (FINOS CCC) is an open standard project that describes consistent controls for compliant public cloud deployments in the financial services (FS) sector.

This standards is a collaborative project which aims to develop a unified set of cybersecurity, resiliency, and compliance controls for common services across the major cloud service providers (CSPs).

FINOS COMMON CLOUD CONTROLS



Why Common Cloud Control Exist

Problem Statement

Every organization defines its own cloud security controls. Cloud service providers (AWS, Azure, GCP) must comply with multiple, often conflicting requirements from different customers and regulators. This leads to:

- Duplication of effort across organizations solving the same problems independently
- Inconsistent security posture across providers and environments
- High compliance costs driven by redundant assessments and custom frameworks
- Vendor lock-in due to provider-specific control implementations

Vision

An open standard for cloud security controls, built collaboratively by banks, cloud service providers, and regulators. While focused on financial services, CCC is broadly applicable across industries. It covers three pillars:

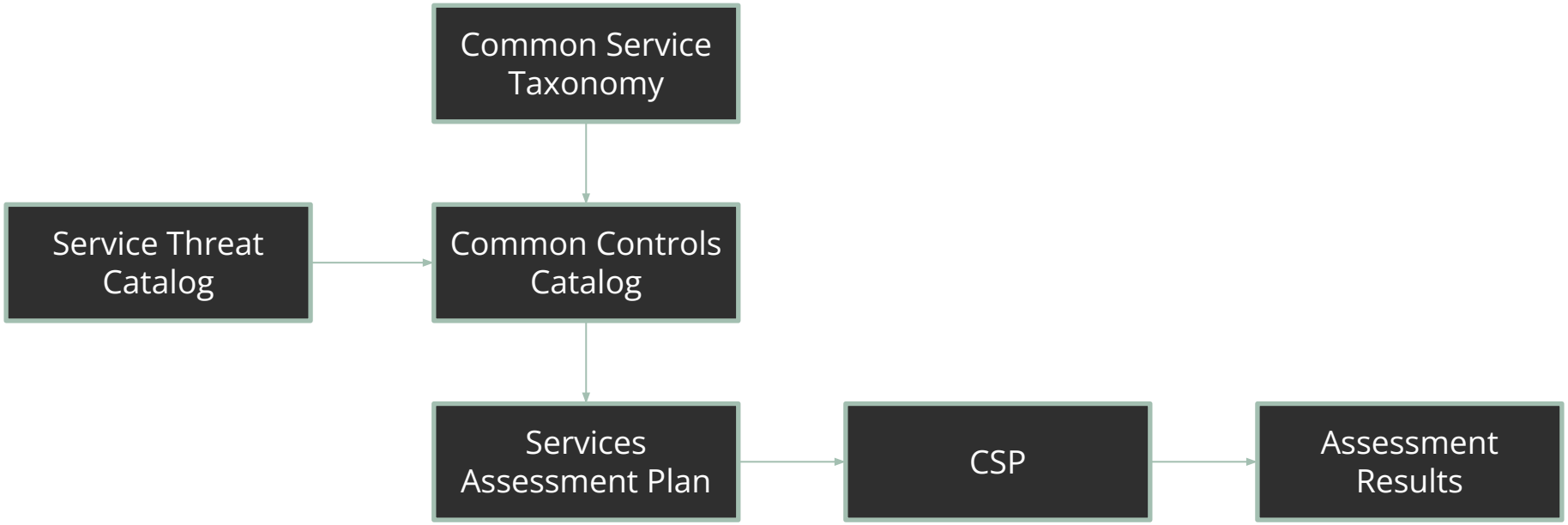
- Security - Baseline protections and threat-informed controls
- Resilience - Availability, recovery, and operational continuity
- Compliance - Regulatory alignment and auditability

Key Goals

- Define best practices for cloud security
- Provide a single target for CSPs to implement against
- Share the compliance burden across the industry
- Enable multi-cloud portability
- Enable future certification models

This reduces fragmentation and creates a common baseline across organizations and providers.

Architecture



Example

Components Overview

Component Title	ID	Number of Releases	Latest Version
CCC Audit Logging	CCC.AuditLog	1	DEV
CCC Batch Processing	CCC.BatchProc	1	DEV
CCC Build	CCC.Build	1	DEV
CCC Container Registry	CCC.CntrReg	1	DEV
Common Cloud Controls Core	CCC.Core	2	2025.10
CCC Data Warehouse	CCC.DataWar	1	DEV
CCC ETL	CCC.ETL	1	DEV
CCC Generative AI Platform	CCC.GenAI	1	DEV
CCC Identity and Access Management	CCC.IAM	1	DEV
CCC Managed Kubernetes Container Orchestration	CCC.K8S	1	DEV
Key Management	CCC.KeyMgmt	2	2025.07-MIGRATION-PREVIEW
CCC Load Balancer Capabilities	CCC.LB	2	2025.07-MIGRATION-PREVIEW
CCC Logging	CCC.Logging	1	DEV
CCC Machine Learning Development Environment	CCC.MLDE	1	DEV
CCC Messaging Services	CCC.Message	1	DEV
CCC Monitoring	CCC.Monitor	1	DEV
FINOS CCC Object Storage	CCC.ObjStor	2	2025.01-MIGRATION-PREVIEW
CCC Relational Database Management System	CCC.RDMS	2	2025.05-MIGRATION-PREVIEW
Secret Management	CCC.SecMgmt	1	DEV
CCC Serverless Computing	CCC.SvlsComp	1	DEV
Cloud Tracing & Observability	CCC.Tracing	1	DEV
CCC Virtual Machines	CCC.VM	1	DEV
CCC Virtual Private Cloud	CCC.VPC	2	2025.01-MIGRATION-PREVIEW
CCC Managed Vector Store	CCC.Vector	1	DEV

Capabilities

ID	Title	Description	Threat Mappings
CCC.LB.CP01	Static Load Balancing	Employ load balancing algorithms that follow fixed rules, independent of the current server state.	0
CCC.LB.CP02	Dynamic Load Balancing	Employ load balancing algorithms that consider the current state of servers before distributing traffic. Load balancer adjusts traffic distribution in real-time based on the current server health, resource utilization, and traffic conditions.	1
CCC.LB.CP03	Layer 7 Routing	Providing distribution of incoming traffic based on the application layer or layer 7 (on ISO model) information. Some of the supported protocols on layer 7 are HTTP, HTTPS, HTTP/2, gRPC, and WebSockets.	0
CCC.LB.CP04	Layer 4 Routing	Providing distribution of incoming traffic based on the transport layer or layer 4 (on ISO model) information. It uses the combination of IP addresses and TCP/UDP port to distribute incoming traffic rather than inspecting the actual content of the packets.	0
CCC.LB.CP05	URL-Based Routing	Direct incoming requests to different backend resources based on the content of the request URL.	0
CCC.LB.CP06	HTTP Header-Based Routing	Direct incoming requests to different backend resources based on the values of HTTP headers.	0
CCC.LB.CP07	WebSocket Support	Ability to support web socket communication.	0
CCC.LB.CP08	Dual-stack Load Balancing	Ability to support traffic originated from both IPv4 and IPv6.	0
CCC.LB.CP09	Load Balancer Autoscaling	Ability for the load balancer to dynamically adjust its capacity in response to fluctuations in incoming traffic.	0
CCC.LB.CP10	Target Autoscaling	Ability for the load balancer to trigger scaling actions of the backend instances (targets) to handle fluctuations in incoming traffic.	0
CCC.LB.CP11	SSL/TLS Termination	Process of decrypting SSL or TLS encrypted traffic at the load balancer level rather than at the backend servers. This allows the load balancer to offload the decryption task from the backend servers.	1
CCC.LB.CP12	Target Health Checks	Ability to continuously perform health checks on backend backend targets in form of checking the response to HTTP request, TCP connection or checking other application-specific parameter	1

Threats

ID	Title	Description	External Mappings	Capability Mappings	Control Mappings
CCC.Core.TH01	Access is Granted to Unauthorized Users	Logic designed to give different permissions to different entities may be misconfigured or manipulated, allowing unauthorized entities to access restricted parts of the service, its data, or its child resources. This could result in a loss of data confidentiality or tolerance of unauthorized actions which impact the integrity and availability of resources and data.	1	1	4
CCC.Core.TH02	Data is Intercepted in Transit	Data transmitted by the service is susceptible to collection by any entity with access to any part of the transmission path. Packet observations can be used to support the planning of attacks by profiling origin points, destinations, and usage patterns. The data may also be vulnerable to interception or modification in transit if not properly encrypted, impacting the confidentiality or integrity of the transmitted data.	1	1	1
CCC.Core.TH07	Logs are Tampered With or Deleted	Tampering or deletion of service logs will reduce the system's ability to maintain an accurate record of events. Any actions that compromise the integrity of logs could disrupt system availability by disrupting monitoring, hindering forensic investigations, and reducing the accuracy of audit trails.	1	1	0
CCC.Core.TH09	Runtime Logs are Read by Unauthorized Entities	Unauthorized access to logs may expose valuable information about the system's configuration, operations, and security mechanisms. This could jeopardize system availability through the exposure of vulnerabilities and support the planning of attacks on the service, system, or network. If logs are not adequately sanitized, this may also directly impact the confidentiality of sensitive data.	1	1	0
CCC.Core.TH12	Resource Constraints are Exhausted	Exceeding the resource constraints through excessive consumption, resource-intensive operations, or lowering of rate-limit thresholds can impact the availability of elements such as memory, CPU, or storage. This may disrupt availability of the service or child resources by denying the associated functionality to users. If the impacted system is not designed to expect such a failure, the effect could also cascade to other services and resources.	1	1	0
CCC.Core.TH15	Automated Enumeration and Reconnaissance by Non-human Entities	Automated processes may be used to gather details about service and child resource elements such as APIs, file systems, or directories. This information can reveal vulnerabilities, misconfigurations, and the network topology, which can be used to plan an attack against the system, the service, or its child resources.	1	1	0

Controls

ID	Title	Objective	Control Family	Threat Mappings	Guideline Mappings	Assessment Requirements
CCC.Core.CN01	Encrypt Data for Transmission	Ensure that all communications are encrypted in transit to protect data integrity and confidentiality.	Data	1	4	5
CCC.Core.CN02	Encrypt Data for Storage	Ensure that all data stored is encrypted at rest using strong encryption algorithms.	Data	1	4	1
CCC.Core.CN03	Implement Multi-factor Authentication (MFA) for Access	Ensure that all sensitive activities require two or more identity factors during authentication to prevent unauthorized access.	Identity and Access Management	1	1	4
CCC.Core.CN04	Log All Access and Changes	Ensure that all access attempts are logged to maintain a detailed audit trail for security and compliance purposes.	Logging & Monitoring	1	1	3
CCC.Core.CN05	Prevent Access from Untrusted Entities	Ensure that secure access controls enforce the principle of least privilege to restrict access to authorized entities from explicitly trusted sources only.	Identity and Access Management	1	5	6
CCC.Core.CN06	Restrict Deployments to Trust Perimeter	Ensure that the service and its child resources are only deployed on infrastructure in locations that are explicitly included within a defined trust perimeter.	Data	1	1	2
CCC.Core.CN10	Restrict Data Replication to Trust Perimeter	Ensure that data is only replicated on infrastructure in locations that are explicitly included within a defined trust perimeter.	Data	1	2	1
CCC.LB.CN01	Enforce and Detect Rate Limiting	Detect and throttle malicious or excessive requests to prevent downstream resource exhaustion and brute-force activity.	Logging & Monitoring	2	6	2
CCC.LB.CN02	Auto-Scale Load Balancer Capacity	Expand load-balancer capacity to maintain availability during traffic spikes.	Data	1	2	1

Implementation of CCC with IaC

Home > cfi

Compliant Financial Infrastructure

Implementation of Common Cloud Controls in Infrastructure as Code

CFI Configurations

Available CFI configurations across different cloud providers and services

ID	Provider	Name	Description	Repository	GitHub Link
azure-postgresql-flexibleserver	AZURE	CCC Azure PostgreSQL Flexible Server Terraform Module	This module creates a secure Azure Database for PostgreSQL ...	ccc-cfi-compliance-2	View GitHub
secure-azure-storage	AZURE	CCC Azure Storage Account Terraform Example	This example creates a secure Azure storage account with enc...	ccc-cfi-compliance-2	View GitHub

Implementation of CCC with IaC

2026 OPEN SOURCE IN FINANCE FORUM Toronto

Configuration Summary

ID	secure-azure-storage
Provider	AZURE
Name	CCC Azure Storage Account Terraform Example
Description	This example creates a secure Azure storage account with encryption, versioning, and lifecycle management enabled.
Service	Storage
Path	`examples/complete/azure`
GitHub Link	View GitHub Repository
Terraform Files	View Terraform Files

Repository Name	ccc-cfi-compliance-2
Description	CCC CFI Compliance from FINOS Labs
Repository URL	https://github.com/finos-labs/ccc-cfi-compliance
Downloaded At	October 19, 2025 at 05:38 PM
Workflow Status	failure Run #18633357206

Configuration Results

Test results partitioned by product, vendor, and version

Vendor	Product	Version	Total Tests	Passing	Failing	Actions
Prowler	Prowler (baseline)	5.13.0	85	36	49	View Details →
FINOS	CCC-Destructive	0.1	1	0	0	View Details →
Prowler	Prowler (complete)	5.13.0	102	44	58	View Details →
Prowler	Prowler (delta)	5.13.0	17	8	9	View Details →

Implementation of CCC with IaC

Control Catalog Summary

Summary of test results grouped by control catalog and resource

Control Catalog	Resources	Total Tests	Passing	Failing	Tested Requirements	Missing Requirements
CCC.AuditLog	Diagnostic Settings, Monitor	12	0	12	CCC.AuditLog.CN02.AR01, CCC.AuditLog.CN03.AR01, CCC.AuditLog.CN03.AR02, CCC.AuditLog.CN04.AR01, CCC.AuditLog.CN05.AR01, CCC.AuditLog.CN06.AR01, CCC.AuditLog.CN08.AR01, CCC.AuditLog.CN09.AR01	CCC.AuditLog.CN01.AR01, CCC.AuditLog.CN01.AR02, CCC.AuditLog.CN07.AR01, CCC.AuditLog.CN10.AR01, CCC.AuditLog.CN10.AR02
CCC.Build	nsg-lee8	4	4	0	CCC.Build.CN03.AR01	CCC.Build.CN01.AR01, CCC.Build.CN02.AR01
CCC.CntrReg	Containers	1	0	1	CCC.CntrReg.CN01.AR01	CCC.CntrReg.CN02.AR01

Lab1: Your First FINOS Contribution to CCC

1. Go to the repository: https://github.com/finos/common-cloud-controls
2. Key sections to explore:
   - **/controls/**
   - **/services/**
   - **/docs/**

Mission: Improve an Existing Control

Example tasks:

- Clarify ambiguous wording
- Add missing rationale
- Improve security precision

Lab1: Your First FINOS Contribution to CCC

Quality Checklist

Before submitting, verify your change is:

- Clear and concise
- Cloud-agnostic
- Actionable
- Security-focused
- Free of vendor bias

Lab1: Your First FINOS Contribution to CCC

2026 OPEN SOURCE IN FINANCE FORUM Toronto

Create a Pull Request

Use the following PR template:

What

Improved IAM control by adding:

- Clear definition of least privilege

- Kubernetes-specific implementation guidance

Why

To enhance clarity and improve multi-platform applicability.

Impact

- Better usability for Kubernetes users

- Stronger alignment with real-world implementations

Lab2: Add your first control

CCC is cloud-agnostic, but Kubernetes is often:

- Underrepresented in control coverage
- Only implicitly covered by broader compute controls
- Not operationally specific enough for real-world use

Mission:

Make CCC more actionable for Kubernetes environments.

Lab2: Add your first control

Suggested Control Ideas

- Prevent use of cluster-admin role binding
- Enforce namespace-level resource quotas
- Require Pod Security Standards enforcement
- Mandate network policy coverage for all namespaces

Lab3: Create your own test with Privateer

https://github.com/privateerproj

Contributing Organisations

Additional ways to contribute

- Document control catalogues
- Threat modeling
- Map the controls and threats
- Help with test requirements
- Documents capabilities
- Review open PRs
- Enhance existing capabilities, threats & Controls
- Help with completing taxonomy
- Participate in community calls to share your expertise

Resources

- FINOS CCC: https://ccc.finos.org
- FINOS CCC Catalog: https://commoncloudcontrols.com
- FINOS CCC Github Repo: https://github.com/finos/common-cloud-controls
- FINOS CCC Meetings: https://calendar.finos.org
- OpenSSF Gemera: https://openssf.org/projects/gemara

Questions?

PRESENTED BY

FINOS

THE LINUX FOUNDATION

#OSinFinance | #OSFF2026