Storage System - Security Requirements

Vendor Requirements

Management Backend	**Service Frontend**	**Requirement**	**Level**	**Category**	**Notes**
✅	❌	The management plane MUST support management interfacing through a different network interface than the storage endpoints servicing customers.	MUST	Network Security	
✅	❌	The management plane MUST support TLS 1.2, and MUST block unencrypted communication.	MUST	Network Security	
✅	❌	The management plane SHOULD support TLS 1.3	SHOULD	Network Security	
✅	✅	The management interface SHOULD support blocking SSH access	SHOULD	Network Security	
✅	✅	The service SHOULD support external, valid SSL certificate	SHOULD	Network Security	
❌	✅	The service MUST support Kerberos Authentication for NFS (*krb5p specifically)* that includes authentication and encrypt all traffic between the storage system and the target server.	MUST	Network Security	
✅	✅	Access MUST support external a different identity provider for the management backend vs. the servicing endpoints	MUST	Access Control	
❌	✅	Access SHOULD support multiple different identity providers for the storage servicing endpoints	SHOULD	Access Control	
✅	✅	The system SHOULD support MFA for local accounts (in cases where federation is not done).	SHOULD	Access Control	

✅	✅	Configuration MUST support role based access control based on least privileges and support at least: • Management Interface ○ Administrative Permissions ○ Read-Only Permissions ○ Segmented permissions based on attributes in the system (e.g. user X should not see nor control volume 1, see volume 2 but not change it and manage volume 3 completely) ○ Segmented permissions based on modules / capabilities of the system (e.g., user X is a NOC operator and is only allowed to see performance metrics within the system). • Provisioned Storage (Volumes) - Full CRUD capabilities with granularity up to the file level • Provisioned Storage (Object Storage) - Full S3-compatible permissions granularity including (in example): ○ Bucket level actions (list, create, delete etc.) ○ Object level actions (get, put, delete, update)	MUST	Access Control	
❌	✅	The system SHOULD support any amount of clientIDs and Secrets for object storage (e.g. for a specific bucket have 15 different access IDs and secrets)	SHOULD	Access Control	
❌	✅	The system MUST support least privileges for object storage clients reaching to the object level itself (e.g. User X can list all folders, read 1 folder, change 3 files within that folder and only create new files within another folder)	MUST	Access Control	

✅	✅	For local users, the system MUST support complex password policies for users and IDs/Secrets including: • Password length (12 characters or more) • Password complexity (Uppercase, Lowercase, Number & Special Character) • Blocking same password reuse (keep history of last 20 passwords and prevent reuse) • Password rotation every 90 days for users and 1 year for service accounts. • Account Lockout after 5 failed authentication attempts	MUST	Access Control	
✅	✅	For local users, the system SHOULD support further password policies for users and IDs/Secrets including: • Prevent a usage of a password from a vendor predefined dictionary (e.g. common passwords) • Prevent a usage of a password from a Cerebras defined dictionary (e.g. known generic passwords)	SHOULD	Access Control	
✅	❌	The management plane MUST support time-out configuration for web sessions	MUST	Access Control	
✅	✅	The service SHOULD support RESTful APIs for management, monitoring and instrumentation of services and data management (Create, Read, Update, Delete) etc.	MUST	Access Control	
❌	✅	While using **Object Storage**, the service MUST authenticate the user for each request	MUST	Access Control	
❌	✅	The service MUST support authorizing every request across object storage and NFS	MUST	Access Control	

✓	✓	The service MUST log every user interaction with it, and incorporate at least: • Date/Time of the event • Who is the user • What was the action (Create, Read, Update, Delete) • What was the resource on which the action was taken • What was the outcome (Success, Failure)	MUST	Logging & Auditing	
✓	✓	The service MUST log both success and failed attempts.	MUST	Logging & Auditing	
✓	✓	The service MUST be configured to store logs them locally at least for 7 days.	MUST	Logging & Auditing	
✓	✓	The service MUST be configured to send logs via Syslog to a central logging system	MUST	Logging & Auditing	
✗	✓	Customer audit trail SHOULD support the ability to store event logs in a separate object store per definition (e.g. logs from all volumes including the ID 12345 would go to bucket X while logs from all volumes including the ID 7890 would go to bucket Y)	SHOULD	Logging & Auditing	
✓	✗	The management plane MUST support an external system for the management of encryption keys	MUST	Data Security	
✗	✓	The exchange of encryption keys and management of the key lifecycle MUST be done using the KMIP protocol	MUST	Data Security	
✓	✓	The service MUST NOT allow for export of encryption keys.	MUST	Data Security	
✗	✓	The service SHOULD support versioning for files on both Volumes and Object storage	SHOULD	Data Security	
✗	✓	The service MUST support utilization of different encryption keys per volume and/or per bucket	MUST	Data Security	

✖	✅	The service SHOULD support using Client-Side Encryption (similarly to S3)	SHOULD	Data Security	
✖	✅	The system SHOULD implement the following Hierarchy: • Cerebras Master Key - Stored within a secure vault (not within the storage system) ○ Customer Master Key - Created by Cerebras*, specific to each customer at the organization level ▪ Tenant Key - Specific to each tenant of the customer ▪ Project Key - Created for each project ▪ Data Key - Created for each volume / bucket** **Customer Master Key MAY be leveraging a BYOK (Bring Your Own Key) model that allows clients to share their key with Cerebras.* *** Data Key SHOULD be created at least at the volume / bucket level. it is preferable to get to the file level, however this should be evaluated against performance and other operational needs.*	SHOULD	Data Security	Elaboration and vendor proposals are welcomed.