Cluster Storage - Vendor Security Assessment Checklist

- Date: 2025-0730
- Version: 1.1

Network Security

Management Backend	Service Frontend	Requirement	Level	Validate	Weight
TRUE	FALSE	The management plane **MUST** support management interfacing through a different network interface than the storage endpoints servicing customers.	MUST		
TRUE	FALSE	The management plane **MUST** support TLS 1.2, and **MUST** block unencrypted communication.	MUST		
TRUE	FALSE	The management plane **SHOULD** support TLS 1.3.	SHOULD		
TRUE	TRUE	The management interface **SHOULD** support blocking SSH access.	SHOULD		
TRUE	TRUE	The service **SHOULD** support external, valid SSL certificate.	SHOULD		
FALSE	TRUE	The service **MUST** support Kerberos Authentication for NFS (krb5p specifically) that includes authentication and encrypts all traffic between the storage system and the target server.	MUST		

Access Control

Management Backend	Service Frontend	Requirement	Level	Validate	Weight
TRUE	TRUE	Access **MUST** support an external, different identity provider for the management backend vs. the servicing endpoints.	MUST		
FALSE	TRUE	Access **SHOULD** support multiple different identity providers for the storage-servicing endpoints.	SHOULD		
TRUE	TRUE	The system **SHOULD** support MFA for local accounts (in cases where federation is not done).	SHOULD		
TRUE	TRUE	Configuration **MUST** support role-based access control on the principle of least privilege for management interface, provisioned volumes, and S3-compatible object storage (bucket/object-level actions, CRUD granularity, segmented permissions, etc.).	MUST		
FALSE	TRUE	The system **SHOULD** support any amount of clientIDs and secrets for object storage (e.g., a single bucket may have 15 different access IDs/secrets).	SHOULD		
FALSE	TRUE	The system **MUST** support least-privilege object-storage access down to the object level (e.g., granular folder/file permissions and scoped create/list/read/change rights).	MUST		
TRUE	TRUE	For local users, the system **MUST** enforce complex password policies (≥ 12 chars, upper/lower/number/special, history ≥ 20, 90-day rotation, 5-try lockout, etc.).	MUST		
TRUE	TRUE	For local users, the system **SHOULD** enforce additional password policies (vendor/common-password dictionaries, custom dictionary, etc.).	SHOULD		
TRUE	FALSE	The management plane **MUST** support configurable web-session time-outs.	MUST		
TRUE	TRUE	The service **SHOULD** expose RESTful APIs for management, monitoring, and data operations (CRUD).	MUST		

Management Backend	Service Frontend	Requirement	Level	Validate	Weight
FALSE	TRUE	While using Object Storage, the service **MUST** authenticate the user for **each** request.	MUST		
FALSE	TRUE	The service **MUST** authorize **every** request across Object Storage and NFS.	MUST		

Logging & Auditing

Management Backend	Service Frontend	Requirement	Level	Validate	Weight
TRUE	TRUE	The service **MUST** log every user interaction (timestamp, user, action, resource, outcome).	MUST		
TRUE	TRUE	The service **MUST** log both successful **and** failed attempts.	MUST		
TRUE	TRUE	The service **MUST** store logs locally for at least 7 days.	MUST		
TRUE	TRUE	The service **MUST** send logs via Syslog to a central logging system.	MUST		
FALSE	TRUE	Customer audit trail **SHOULD** support routing event logs to separate object stores per definition (e.g., volume-based bucket targets).	SHOULD		

Data Security

Management Backend	Service Frontend	Requirement	Level	Validate	Weight
TRUE	FALSE	The management plane **MUST** integrate with an external encryption-key management system.	MUST		
FALSE	TRUE	Key exchange and lifecycle management **MUST** use the KMIP protocol.	MUST		
TRUE	TRUE	The service **MUST NOT** allow export of encryption keys.	MUST		
FALSE	TRUE	The service **SHOULD** support file/versioning on both Volumes and Object Storage.	SHOULD		
FALSE	TRUE	The service **MUST** support different encryption keys per volume and/or per bucket.	MUST		
FALSE	TRUE	The service **SHOULD** support client-side encryption (similar to S3).	SHOULD		
FALSE	TRUE	The system **SHOULD** implement a hierarchical key model (Master → Customer → Tenant → Project → Data Key) and may offer BYOK; Data Keys **SHOULD** be at least per volume/bucket.	SHOULD		