SSH Key Management Using Universal SSH Key Manager (UKM) VS CyberArk

Comparison sheet

Legend

- ✓ The solution supports the mentioned feature or functionality.
- x The solution doesn't support the functionality or feature.
- \- The functionality or feature is limited.

DISCOVERY			
Feature	**UKM**	**CyberArk**	**Notes**
SSH keys			
User accounts (local, domain, LDAP)	✓	✓	
Private SSH keys	✓	✓	
Authorized keys	✓	✓	
Encrypted private keys	✓	X	UKM records metadata on passphrase-protected private keys, including access estimation based on available public key data. The data is verified once a passphrase is provided to decrypt the key.
Host keys	✓	X	
SSH keys usage inventory			
SSH keys used for access	✓	✓	
Failed/successful SSH access with other credentials (passwords, certificates)	✓	X	
SSH server inventory			
Installed SSH software	✓	X	
SSH client and server configurations	✓	X	
Alerting on			
Changes to existing authorized keys	✓	X	For example, changes to "from" option relaxing restrictions placed on source IP/FQDN from where access is allowed.
Appeared SSH keys	✓	X	A common way to achieve persistence after perimeter breach (link to MITRE).
Key access with unknown/unsanctioned keys	✓	X	
Changes to SSH client/server configurations	✓	X	
POLICIES			
Feature	**UKM**	**CyberArk**	**Notes**
SSH key policies reporting on			
Key algorithm and size	✓	-	CyberArk considers elliptic curve encryption keys (ECDSA and Ed25519) as violations of policy standards, contrary to NIST recommendations.
Key age	✓	✓	

Unused keys	✓	X	
Segregation of duties (access from DEV to PROD)	✓	X	
Orphan keys	✓	✓	
Authorized keys without proper restrictions	✓	X	An established security practice for hardening authorized keys to prevent access from anywhere in case of private key compromise.
Incorrect key permissions	✓	X	SSH keys' file permissions might block the key usage. UKM can recognize and remediate this situation when it is detected.
Authorized key lockdown	✓	X	
SSH server policies reporting on			
SSH servers allowing password authentication for admin accounts	✓	X	
SSH server options (agent-forwarding, x11-forwading, etc.)	✓	X	
Deprecated Signature, HMACs encryption algorithms	✓	X	
Adoption of quantum-safe KEX algorithms	✓	X	
PRODUCT OPERATION			
Feature	**UKM**	**CyberArk**	**Notes**
Automation and delegation			
Rule-based automation engine in GUI	✓	X	
API/CLI for all operations including bulk actions	✓	X	
Delegating key remediation process to key owners	✓	X	
Four-eyes-principle approval process for key action	✓	X	
Agentless operations for all use cases	✓	X	With CyberArk Key Management for M2M, access requires an agent.

SSH KEY MANAGEMENT			
Feature	**UKM**	**CyberArk**	**Notes**
Lifecycle management			
Support of key types (RSA, DSA, ECDSA, Ed25519)	✓	-	CyberArk does not support modern elliptic curve algorithms for access (ECDSA and Ed25519).
Provisioning interactive user keys through a jump server	✓	✓	
Provisioning M2M keys	✓	-	CyberArk requires installing an agent and modification of scripts/integrations to make a request for using the private key from the vault.
Deleting keys	✓	✓	
Restoring keys	-	-	UKM cannot restore a private key that was deleted without using UKM. However, when using UKM, it utilizes local copies to manage restoration of even private keys without vaulting them. On the other hand, CyberArk can restore only keys that are in the vault.
Rotating keys	✓	-	In CyberArk, only RSA and DSA key types are available. Also, rotation is only practical for interactive access from CyberArk to the target. M2M keys require an agent and vaulting the keys in order to provide rotation.
Applying authorized key source restrictions	✓	X	An established security practice for hardening authorized keys is to prevent access from anywhere in case of private key compromise.
Limiting validity period of authorized keys	✓	X	While SSH servers do not support limiting validity for SSH keys, UKM is capable of automatically provisioning and removing a key at specified times.
Blacklisting keys	✓	X	
Host key management	✓	X	
Zero Trust SSH access			
Provisioning SSH access using ephemeral certificates	✓	✓	
Automated migration of existing SSH key access to Just-In-Time (JIT) ephemeral certificates	✓	X	