DRAFT - PAM | CyberArk | Azure Shell Security

Introduction

This document has been created to detail options for securing access to and usage of an Azure Cloud Shell (or equivalent) environment within M&G via the CyberArk Privileged Account Security solution as outlined in CyberArk - Azure Console Use Cases

Powershell Approach

Use Case

Accounts in the Global Administrators role are often used to run powershell scripts for Azure administrative purposes therefore PowerShell access needs to be facilitated

Requirements

1. Allow powershell scripts to be run against Azure under a privileged role (e.g. Global Admin)
2. Ensure that sandbox/session isolation in PSM and integrity of CyberArk environment is maintained
3. Ensure Video Recording Capability is retained
4. Ensure there are adequate controls to prevent data leakage outside of the organisation
5. Ensure that there are adequate controls to prevent the ingress of Malware via any solution delivered
6. Ensure that there are adequate controls to prevent credential replay/privilege escalation to the in-scope roles outside of the PAM process (e.g. by accessing and using cached credentials which are created when logging into Azure via Powershell)
7. Avoid maintenance overhead in ensuring the tooling is always current (updating of relevant modules, which do get updated reasonably frequently)
8. Allow **ONLY** a trusted repository to be accessed from within solution to access and run trusted code, and to upload script output as required.

Solutions Discussed/Proposed

Solution ID	Description	Requirements Met /Can Be Met	Requirements NOT Met	Risks/Issues Exposed
1	Install the relevant Azure Modules into PowerShell on the CyberArk PSM servers	1,3,8	2,4,5,6,7	• RISK: Data Ex-filtration • RISK: Malware Ingress • ISSUE: Maintenance Overhead
2	Build a separate tooling server in Azure to host the AZ modules - accessed via PSM	1,2,3,8	4,5,6,7	• RISK: Cached Credential Compromise /Replay - Privilege Escalation • RISK: Data Ex-filtration • RISK: Malware Ingress • ISSUE: Maintenance Overhead
3	Utilize the Azure Cloud Shell feature, providing users of the vaulted AAD accounts with an Azure hosted PowerShell or Bash session after connection via the PSM web based connection component	1,2,3,7	4,5,6,8	• RISK: Cached Credential Compromise /Replay - Privilege Escalation • RISK: Data Ex filtration • RISK: Malware Ingress • RISK: No Tenant/Subscription Isolation

Azure Cloud Shell Risk Assessment

While the Cloud Shell solution meets the most requirements, it is far from risk free.

Malware Ingress and Data Ex-filtration

The cloud shell allows upload into the storage account used to host the cloud shell container, and also download from there to the machine the browser is running on.

As the browser would be running in a PSM session with no access to the users local desktop, this does offer some protection in that the only ingress/egress routes for data to/from the cloud shell within the organisation are those that the PSM itself has access to. The PSMs have only limited access to external URLs, such as SaaS services it is managing Privileged Access to (OracleCloud.com for example)

However, from within the cloud shell itself, it can access the internet directly, and thus bypass our Cloud Hosted Palo Alto.

Therefore there is a significant risk of both malware ingress, and data ex-filtration.

https://securitysandman.com/2021/04/28/public-cloud-web-shells-a-serious-malware-and-dlp-issue/

> - *Most of these cloud shells have direct internet access which bypasses your IDS/IPS/DLP and corporate proxies network routes*
> - *Malware can either be side-loaded through the internet or via the external persisted storage with the user's home directory path*
> - *There's no logs of the local actions, no logs of the internet egress and no antivirus or endpoint HIDS/HIPS*

Harvesting Cached Credentials for Credential Replay/Privilege Escalation outside of CyberArk

https://www.netspi.com/blog/technical/cloud-penetration-testing/attacking-azure-cloud-shell/?print=print

> *By default, Azure Subscription Contributors have access to all storage accounts in a subscription. These storage accounts can contain Azure Cloud Shell storage files (Linux home directories) that can contain sensitive information. By modifying these Cloud Shell files, an attacker can execute commands in the Cloud Shell sessions of other users. This can lead to cross-account command execution and privilege escalation.*

The Cloud Shell is a Linux container, whose home drive is stored in a storage account assigned to the Cloud Shell instance.

This instance can be be assigned to a specific, pre-existing storage account, or if not, a storage account can be created at time of initiating a Cloud Shell instance (subject to RBAC role and Azure Policy restrictions).

This has the benefit of ensuring that the tools are always kept up to date by Microsoft, but has the disadvantage that the container image and home drives may not be adequately secured, such that a malicious actor could either amend the home drive/environment variables, or exfiltrate cached credentials left behind by the `***az login***` command in the AZ CLI and the `***az-connect***` command in the Az Powershell modules (these are stored in the user profile on the machine where they are run and not cleared automatically).

So how do we mitigate these risks?

Do not allow shell access via PSM until secure solution is designed/tested/signed off

As Azure Policy does not provide the ability to disable Cloud Shell - it may be possible to block access by restricting access to the below URL from the PSM until a secure shell solution is available:

https://ux.console.azure.com

This would require these URLs to also be blocked outside of the PSM and a review of RBAC permissions and Azure Policy to prevent creation of a storage account for Cloud Shell usage (either from a corporate device, or outside the corporate network). Alternatively prevent login to the portal from device that isn't AD Hybrid Joined - though consideration must be given to breakglass scenarios.

Allow cloud shell with enforcement of Azure Security Benchmark Recommendations for Azure Cloud Shell

The Cloud Shell Security Baseline contains recommendations which would reduce the cached credential risk somewhat by:

- Requiring use of a dedicated storage account for the cloud shell instance - secured according to both Cloud Shell and Azure Storage benchmarks (such as ensuring all traffic is HTTPS).
- Requiring the storage account be tied to a specific dedicated VNET - although Cloud Shell doesn't support NSGs, this does offer the opportunity to secure via either our Azure Hosted Palo Alto Firewall infrastructure or an Azure Firewall instance as required

This does not, however, resolve all malware ingress or data-exfiltration risks - it only controls access to the storage account, not the running cloud shell container.

This would be the ***minimum*** control required:

Minimum ASB Requirements

Network Security

NS-1: Implement security for internal traffic
Guidance: When you deploy Cloud Shell resources, create or use an existing virtual network. Ensure that all Azure virtual networks follow an enterprise segmentation principle that aligns with the business risks. Any system that might incur higher risk for the organization should be isolated within its own virtual network and sufficiently secured with a network security group (NSG) and/or Azure Firewall.

NS-4: Protect applications and services from external network attacks
Guidance: Protect your Cloud Shell resources against attacks from external networks, including distributed denial of service (DDoS) attacks, application-specific attacks, and unsolicited and potentially malicious internet traffic. Use Azure Firewall to protect applications and services against potentially malicious traffic from the internet and other external locations. Protect your assets against DDoS attacks by enabling DDoS standard protection on your Azure virtual networks. Use Azure Security Center to detect misconfiguration risks to your network related resources.

Identity Management

IM-1: Standardize Azure Active Directory as the central identity and authentication system
Guidance: Cloud Shell uses Azure Active Directory (Azure AD) as the default identity and access management service.

IM-7: Eliminate unintended credential exposure
Guidance: Cloud Shell allows customers to deploy/run {code or configurations or persisted data} potentially with identities/secrets. It's recommended to implement Credential Scanner to identify credentials within {code or configurations or persisted data}. Credential Scanner will also encourage moving discovered credentials to more secure locations like Azure Key Vault.

Logging and Threat Detection

LT-3: Enable logging for Azure network activities
Guidance: By design, even though Cloud Shell resources can be deployed into a virtual network, the traffic to and from the Cloud Shell resources cannot be enforced by or passed through a network security group. Network policies need to be disabled on the subnet for the offering to function correctly. For this reason, you are unable to configure network security group flow logging for Cloud Shell.

This enforces the requirement to use a dedicated VNET to allow appropriate monitoring and network access control.

Refer to the following ASB Documents:

Azure Security Architecture Recommendation - Storage, data, and encryption

Azure Security Fundamentals - Azure Data security, encryption, and storage

Cloud Adoption Framework - Azure data security and encryption best practices

Azure Security Benchmark-Asset management

Azure Security Benchmark-Data protection

Make use of Azure Relay to ensure the cloud shell container runs within a network that M&G control

It is possible to make use of a service called **Azure Relay** to control the access to the cloud shell - such that it can only be access from internal, specified networks.

- Azure Relay is not a free service, please view their pricing. In the Cloud Shell scenario, one hybrid connection is used for each administrator while they are using Cloud Shell. The connection will automatically be shut down after the Cloud Shell session is complete.

This may or may not assist with the malware ingress/data ex-filtration risks but would need to be further investigated/tested to confirm.

Run our own cloud shell container in M&G controlled Azure infrastructure

It is possible to run your own cloud shell container instance in your own Azure infrastructure and not Microsofts.

The container image for Cloud Shell is publicly availably in the Azure Container Registry (ACR), and so it may be possible to instantiate that container from within either Azure Container Images or Azure Kubernetes Service - this would allow us to present a Cloud Shell instance internally that is entirely within our control; be able to be secured by our standard Azure security controls, and control ingress/egress.

- It would not present all of the functionality present in the native Cloud Shell such as the `***Azure:***` drive for querying azure resources.
- It may help with tenant/subscription isolation
- It would allow us to better manage malware and data ex-filtration risks as the running container would only be able to access resources that we control - both internal to and external to the organisations network (deemed to be a combination of both on-premises networks, and our azure resources/networks)
- It would avoid having to open up additional URLs in the firewall for PSM to access, as any additional access required (e.g. to access internal/external repositories) would only be possible from the cloud shell container running within the PSM, but not from the PSM itself
- The container would be continually updated by Microsoft with all the latest tools
- The container would contain no cached credentials - each time a session ends the container would stop - every new session would be spun up from the image with no cached credentials to steal.
- *There is one downside - when accessing cloud shell natively, you are running as a non-root user on the container. When you spin up the container yourself, you are running as root and so have the potential to be able to amend the configuration - this may be ably mitigated by being able to ensure that changes do not persist and by being able to control what the container can access (so you can't reach external repos). Alternatively use the Microsoft container image as base image for our own image with our own customisations and use an ACR task to automatically update our image when the Microsoft image is updated* - https://docs.microsoft.com/en-us/azure/container-registry/container-registry-tutorial-base-image-update

Use MCAS Session Policies to control upload/download within the Google Chrome browser session on the PSM

Microsoft Cloud App Security (MCAS) is a Cloud Application Security Broker (CASB) solution from microsoft that proxies access to web applications via a microsoft proxy specific to your tenant, which can enforce security policies and posture in conjunction with Azure Conditional Access.

It integrates with a number of modern browsers including Edge, and Google Chrome (which is currently the preferred browser for web connection components in our CyberArk environment).

MCAS has a number of Session Policy controls available including the ability to restrict file upload/download in the browser.

It may be possible to use MCAS to address the malware ingress and data ex-filtration risks listed in the use of native Cloud Shell (e.g. where PSM is just launching a browser to a microsoft operated standard cloud shell instance).

<Compare Options>

<Make Recommendation>