Microsoft Azure Asset Management Hardening Guide

Microsoft Azure

Use Case: Asset Management Systems

March 2018

Revision 1

Independent Security Evaluators | www.securityevaluators.com

March 19, 2018

EXECUTIVE SUMMARY

Microsoft has engaged Independent Security Evaluators (ISE) to evaluate a Digital Asset Management (DAM) architecture using the Microsoft Azure Cloud computing infrastructure and to establish an Azure specific hardening guide for the media and entertainment (M&E) industry specific to Media Asset Management (MAM) and Digital Asset Management (DAM) system data management workflows. The M&E industry wishes to use Azure in its internal and vendors' software systems to increase the throughput, scalability, and cost efficiency of its film production activities. This document is meant for those who may be deploying MAM/DAM systems. **While Azure provides a variety of benefits, it is the responsibility of the deploying party to configure Azure properly to ensure that security goals are achieved.**

An existing production Azure based asset management system was evaluated to uncover instances of insufficient or missing security controls, security oversights, violations of best practices, or areas in which it is likely that film companies or their vendors may accidentally misconfigure the platform, thus increasing its exposure to security threats. Furthermore, the established security standard arising from this evaluation can be used to develop secure Azure-based asset management system architectures and implementations instantiated by film studios.

The overall goal of this effort is to establish a set of security controls which assist in protection of the primary asset – pre-release studio content – from unauthorized disclosure, modification, or destruction. Other assets that may be loaded into Azure Cloud include artwork, trailers, still photography, and personally identifiable information (PII).

These security controls were developed after hands-on evaluation of relevant Azure services, access to all publicly-available documentation, and assessment of existing in-production MAM/DAM setups. This effort did not include access to source code other than in publicly available clients and software development kits (SDKs), nor did it include assessment of any back-end supporting infrastructure other than that available to a cloud platform customer.

The security controls will be used by deployment administrators to harden an Azure-based MAM/DAM setup. This guide is current as of March, 2018. Changes to Azure after this date may invalidate certain recommendations or introduce new concerns. Furthermore, it is the responsibility of the users to understand their deployments and the security risks involved.

Overall, we recommend that studios consider the security controls described in this document and perform independent deployment assessments of individual asset management system instances in the future. In particular, while as in a physical data center, users can configure Azure Cloud so that the virtual machines running within it operate in a hardened environment, an assessment of the cloud service in isolation provides no assurances of the security of any software running inside those machines.

Revision	Date	Description
1.0	March 2018	Initial report.

Table 1. Revisions.

As of January 2018, we have identified 44 security controls in 10 categories that should be considered by those deploying a MAM/DAM system with Azure.

EXECUTIVE SUMMARY

Security Controls Summary

The following tables summarize security controls when deploying an asset management system in Azure. They are arranged into thirteen categories and each item is detailed in the sections on security controls below.

Portal	• Prohibit use of personal Microsoft accounts for administration • Use separate Azure subscriptions to segregate work • Logically relate and manage deployments with resource groups • Avoid concurrent access from multiple locations • Enforce custom session inactivity termination • Enforce two-factor authentication for portal access

Table 2 Azure Portal security controls

Networking	• Use network security groups • Isolate virtual appliances in their own subnet • Apply a multi-tiered architecture using VNets • Create separate VNets for production • Limit default Network Security Group VNet communications • Tightly configure endpoints • Do not use deprecated cryptography when configuring IPSec VPN

Table 3 Azure network security controls

Access Control (RBAC)	• Employ custom Role-Based Access roles to manage user access • Extend on-premises identity management for access control • Do not use the deprecated Azure Access Control service (ACS)

Table 4 Azure Role-Based Access Control security

Key Vault	• Use a separate Azure Key Vault for each production • Use Key Vault permissions to manage access • Segregate data and key/secret owners • Manage access to keys and secrets on a per key/secret case • Audit all key management activity • Periodically Rotate Keys

Table 5 Azure Key Vault security controls

Command Line	• Avoid caching session information

Table 6 Azure command line security controls

Batch	• Use obfuscated URLs for Batch accounts • Periodically update access keys • Sanitize and destroy Batch accounts when no longer needed • Create storage exclusively for specific Batch workflows • Use security-validated application packages for batch workflow • Use integrity checks on application packages for Batch workflow • Hold workflow if Batch applications fail • Log Batch events for monitoring and diagnostics

Table 7 Azure Batch security controls

Compute	• Use public key authentication for virtual machines • Use only hardened OS images for VM instantiation

Table 8 Azure Compute security controls

Storage	• Use Shared Access Signatures to access storage account resources • Periodically update access keys

Table 9 Azure Storage security controls

Redis Cache	• Disable non-SSL connections • Monitor cache performance

Table 10 Azure Redis Cache security controls

Event Hub	• Prohibit use of Event Hub tokens devices • Use separate keys for access Event Hub access • Create partitions for consumer groups

Table 11 Azure Event Hub security controls

Scheduler	• Validate scheduled actions

Table 12 Azure scheduler security controls

SQL Database	• Use separate Azure database instances for production and clients

Table 13 Azure SQL database security controls

Content Delivery Network	• Use HTTPS protocol and port settings for CDN • Use token authentication to protect CDN content • Define custom HTTP behavior for CDN

Table 14 Azure Content Delivery Network (CDN) security controls

Media	• Use separate Azure storage accounts for media service accounts • Encrypt assets • Configure Live Media archiving policy

Table 15 Azure media storage security controls

Expiration

This report expires *September 30, 2018*.

TABLE OF CONTENTS

TABLE OF CONTENTS

INTRODUCTION

OUR MISSION

ISE is an independent security firm headquartered in Baltimore, Maryland. We are dedicated to providing clients proven scientific strategies for defense. On every engagement our team of analysts and developers use adversary-centric approaches to protect digital assets, harden existing technologies, secure infrastructures, and work with development teams to improve our clients' overall security.

Assessing the client through the eyes of potential attackers allows us to understand possible threats and how to protect against those attacks. Our security analysts are experienced in information technology and product development which allows them to understand the security problems the client faces at every level. In addition, we conduct independent security research which allows us to stay at the forefront of the ever changing world of information security.

Attacks on information systems cannot be stopped, however with robust security services provided by an experienced team, the effects of these attacks can often be mitigated or even prevented.

RELATIONSHIP WITH MICROSOFT

Microsoft has engaged ISE to complete white-box, design and implementation level assessments of Azure Cloud based MAMS/DAMS implementations. In these assessments, we aimed to discover any instances of missing or broken security controls, security oversights, violations of best practices, or areas in which it is likely that Azure's customers may accidentally misconfigure the Azure services, increasing its exposure to security threats.

Based on these assessments, this guide has been created in the hopes that following these suggestions, a MAM/DAM system deployed within the Azure cloud will meet industry best practices.

ISE is an independent party to Microsoft, and does not have a stake in the outcome of our assessments. ISE does not endorse Azure products, or any other product, and is motivated only to improve the security of all products we are engaged to assess.

INTRODUCTION

Microsoft Azure Cloud provides its customers with an enterprise-grade, scalable, high-reliability computing environment that operates more cost-effectively than hardware deployed in a traditional data center. As a result of past assessments, ISE has prepared this guide for best practices when hosting MAM/DAM systems in Azure. Based on the controls specified in this hardening guide media and entertainment companies can make an informed decision whether and under what circumstances its internal teams and vendors should be permitted to use Azure Cloud for management of pre-release theatrical content and other high-value assets.

Microsoft Azure Overview

Azure Cloud includes a variety of components. Some of these components provide similar functionality but use different designs as appropriate for different types of projects. Azure Cloud is designed to be highly flexible and appeal to multiple development and IT management methodologies; it can be managed using a point-and-click Web console, a RESTful API, or client libraries that allow a user to programmatically manage Azure Cloud services from a language of choice.

Given the large number of available services and use-cases in which to interact with these services, we have focused on the following core Azure Cloud products and services. See Figure 1 for a summary of the products and services addressed in this document.

INTRODUCTION

AZURE PORTAL

The Microsoft Azure portal enables an administrator to manage the cloud platform and other Microsoft services from a point-and-click web interface, as an alternative to the command line utility (CLI), PowerShell extension, RESTful APIs, or client libraries. As part of this effort, we evaluated version 2 of Azure Portal.

AZURE NETWORKING

Networking is an integral feature of the Azure Cloud. Most cloud architectures and use-cases use some sort of virtual network component. In addition to traditional network services and security features, such as routing, firewalling, and DNS, the Virtual Private Cloud (VPC) networking capabilities of Azure include various multimodal VPN gateways and security provisions based on geographic deployment and segregation.

AZURE ROLE-BASED ACCESS CONTROL (RBAC)

Role-based access control (RBAC) is a way to manage authorized users access to the system. Under RBAC, role hierarchies are established to provide for the natural structure of an enterprise or workflow. Azure subscriptions are associated with one Azure Active Directory (AD) directory; that is, users, groups, and applications from that directory can manage resources in the Azure subscription. Administrators can grant access by assigning the appropriate RBAC role to users, groups, and applications within a certain scope.

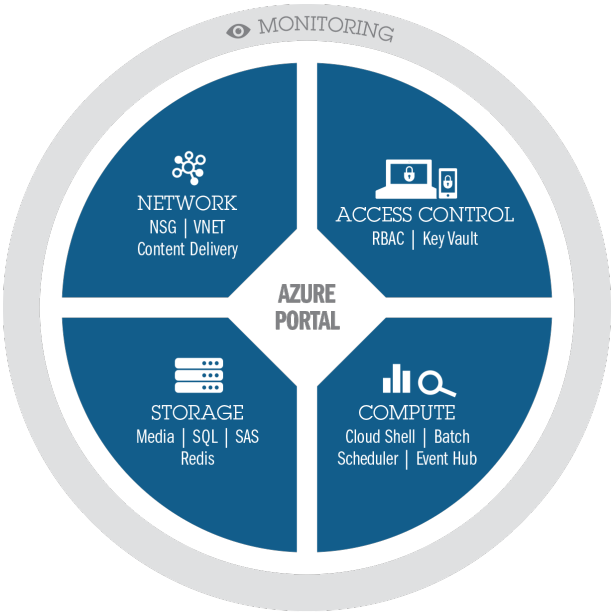

Figure 1 Microsoft Azure components

The old Azure Access Control Service (ACS) has been deprecated as of June 30, 2017 and its functionality has been merged into the Azure AD.

INTRODUCTION

AZURE KEY VAULT^

Cryptography is fundamental in protecting high value digital assets. A robust and secure cryptographic key management system is required to appropriately apply cryptographic techniques for securing digital assets in media and entertainment workflows. Azure provides the Key Vault service for controlling cryptographic keys, digital certificates, and passwords and the access to these items. The use of Key Vault is integrated into Azure's other services. Key Vault is built around a hardware security module (HSM) and has earned a FIPS 140-2 level 2 validation. Microsoft cannot see the keys and secrets stored in the HSM. Monitoring and logging key use is provided with Azure logging.

AZURE BATCH

Microsoft provides APIs to run large-scale parallel and high-performance computing (HPC) workloads in Azure. The core component of an Azure based rendering platform is Azure Batch. Batch is a managed Azure service that is used for batch processing or batch computing–running a large volume of similar tasks for a desired result. Batch computing is most commonly used by organizations that regularly process, transform, and analyze large volumes of data.

AZURE SCHEDULER

Microsoft Azure Scheduler allows developer to declaratively describe actions to run in the cloud. The scheduler services execute scheduled actions automatically. Scheduler schedules jobs, keeps a history of job execution results that one can review, and deterministically and reliably schedules workloads to be run. Scheduler allows developers to create, update, delete, view, and manage jobs and job collections programmatically, by using scripts, and in the portal. The scheduler can invoke code hosted in Azure, on-premises, or with another provider. The scheduler service itself does not host any workloads or run any code.

There are several scenarios that lend themselves to the usage of Scheduler in Media production workflows. For example:

- *Recurring application actions:* Periodically gathering running action on media assets such as recycling of computing or storage assets.
- *Daily maintenance:* Daily pruning of logs, performing backups, and other maintenance tasks.

AZURE EVENT HUBS

Microsoft Azure Event Hubs is a near real-time event ingestion service capable of receiving and processing millions of events per second. Event Hubs can process and store events, data, or telemetry produced by distributed software and devices. Data sent to an event hub can be transformed and stored using any real-time analytics provider or batching/storage adapters. Event Hubs event and telemetry handling capabilities make it especially useful for application instrumentation, user experience or workflow processing or Internet of Things (IoT) scenarios.

In media and entertainment content production environment, Event Hubs can be used to enable near real-time asset management tracking in mobile and desktop collaboration software platform, network traffic information from web farms, event capture in remote media rendering, or telemetry collected from connected devices.

AZURE STORAGE

INTRODUCTION

The ability to provide high-capacity, high-throughput, high-availability, and low-cost storage is a main feature of any cloud platform, and one of the most compelling reasons driving traditional software developers to shift to a cloud-based infrastructure. Azure storage provides the following storage services: Blob, Table, Queue, and File.

- *Blob Storage* stores unstructured object data. A blob can be any type of text or binary data, such as a document, media file, or application installer. Blob storage is also referred to as Object storage.
- *Table Storage* stores structured datasets. Table storage is a NoSQL key-attribute data store, which allows for rapid development and fast access to large quantities of data.
- *Queue Storage* provides reliable messaging for workflow processing and for communication between components of cloud services.
- *File Storage* offers shared storage for legacy applications using the standard SMB protocol. Azure virtual machines and cloud services can share file data across application components via mounted shares, and on-premises applications can access file data in a share via the File service RESTful API.

In additional to basic storage services, Azure offers unmanaged and managed disk storage capabilities as a place to store an operating system, applications, and data for virtual machines in Azure. All data disk types in Azure are stored as a VHD file. Every virtual machine has one attached operating system disk. A VHD disk image are page blobs stored in either standard (HDD) or premium (SDD) storage accounts. Each data disk has a maximum capacity of 1023 GB.

There are two types of disks offered in Azure, managed and unmanaged. Unmanaged disks are configured by the administrator at the account level; the administrator is responsible for configuring the account for maximum performance and to avoid VM throttling. Managed disks handle storage account creation/management and manage the scalability limits of the storage account. Unmanaged storage disks are obsolete and Microsoft suggests Azure Managed Disks for new VMs, and conversion of past previous unmanaged disks to managed disks.

AZURE REDIS CACHE

The Microsoft Azure Redis Cache is a cloud service based on the open source Redis cache. Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. Redis cache supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, and geospatial indexes with radius queries. The Azure Redis service can be deployed in single node, two-node primary/secondary or in a high-availability configuration.

SQL DATABASE

The Microsoft Azure SQL Database is a relational database-as-a service using the Microsoft SQL Server Engine. It is a secure, reliable, and high-performance database which developers can use to build data-driven applications without needing to manage infrastructure. The SQL-based database is fundamental for various media content protection and production workflow, for example workflow such as asset management and web application management.

CONTENT DELIVERY NETWORK

The Microsoft Azure Content Delivery Network (CDN) is a cloud service that deliver pages and other web content to a user, based on the geographic locations of the user, the origin of the webpage and the content delivery server. Using

INTRODUCTION

Azure CDN a developer can build a solution for delivering high-bandwidth content that is hosted in Azure or any other location by caching "internet-facing" objects loaded from Azure blob storage, a web application, virtual machine, application folder, or other HTTP/HTTPS location. Azure CDN offers following products: Azure CDN Standard from Akamai, Azure CDN Standard from Verizon, and Azure CDN Premium from Verizon.

The CDN cache can be held at strategic locations to provide maximum bandwidth for delivering content to users. The major advantages of using the CDN are lower latency and faster delivery of content to users irrespective of their geographical location in relation to the datacenter where the application is hosted.

MEDIA

Microsoft Azure Media Services (AMS) is a cloud-based platform that allows organizations to build and deploy scalable media management and delivery applications. Media Services is based on RESTful APIs that enable developers to securely upload, store, encode, and package video or audio content for both on-demand and live streaming delivery to various clients.

Content can be shared, protected, packaged and delivered using Media Services. The media service can be used to build content creation and delivery end-to-end workflows or integrated with third-party components for parts of the workflow, for example, encoding using a third-party encoder.

Scope

Given the large number of cloud platform services, and the small likelihood that one specific vendor application will use more than a handful of them, our assessment of Microsoft Azure cloud platform was limited in scope to those components most frequently used for visual effects rendering. We focused on the following areas.

- Portal Console
- Azure Batch APIs – limited to containers, job, pool, authentication, and task monitoring APIs
- Azure Network – limited to Network Security Groups (NSG) and Virtual Private Networks (VPN)
- Storage – Blob and Disk storage only
- Key Management – Key Vault
- Scheduler
- Redis Cache
- Azure Access control
- Event Hubs
- Scheduler
- SQL Database
- Content Delivery Network (CDN)
- Media Services (AMS)

Notably, the following areas were excluded from the assessment:

- All non-cloud platform APIs
- Storage – Table, Queue, File
- Non-publicly accessible backend infrastructure
- Non-publicly available source code
- Physical security or other internal security controls or processes

Methodology

The evaluation of the Microsoft Azure cloud platform sought to establish security controls which should be applied in setting up a cloud asset management system using Azure services and products. The first step to any security evaluation is to develop an accurate threat model that identifies the assets to be protected and the threat actors that would seek to do them harm. We propose a threat model given in the next section.

This effort consisted of a manually set up reference implementation of a cloud MAM/DAM system using internet-facing Microsoft Azure cloud platform services, plus documentation reviews of relevant, core components. In the Azure Reference Architecture section below, we outline implementation recommendations for setting up various types of asset management systems using Azure.

Timeline

This section includes a summary of the history of this report.

1: 03/2018

ISE produced this hardening guide, including 48 security controls grouped into 14 functional categories, after examining an existing asset management implementation in Azure.

Expiration

This report expires *September 30, 2018*. Report expiration ensures that ongoing security guidelines for MAM/DAM systems is timely. Given future changes in products and in the Azure environment in response to customer and economic needs, increasing insight into actual customers' usage of the product with changes in best practices for cryptography and other security-critical primitives, and the evolving nature and skill level of threat actors, security is a constant and ever-evolving process. After following the guidelines presented here, ongoing assessment facilitates constant communication between ISE and clients, providing clients with regular feedback in progress addressing past findings and an efficient means to discover new ones.

THREAT MODEL

An effective means of approaching information security is to see it as risk management and as an ongoing process of monitoring and improvement as threats evolve. This requires an understanding of both the system to be defended and the adversaries that threaten it. Assets that require protection need to be identified along with the impact a successful attack would have on those assets. Threat actors' intentions and capabilities need to be modeled and applied against vulnerabilities and their likelihood of impact to the identified assets.

The following subsections expound upon assets, which is relative to a particular application, and threats that are universal.

Assets

Before accurately assessing a system's security posture, it is necessary to identify assets and their value. Assets include tangible elements such as information or equipment and extend to more abstract elements such as reputation. The impact of the loss of these assets should be quantified to the degree possible; however, this can be a difficult and subjective process and is outside the scope of ISE's insight into a client's operations. Every organization should go through the exercise of identifying digital assets and what level of protection is warranted for each.

CREATIVE CONTENT

The most important asset for access management systems is its media content. This can include any form of data that is stored and used in the system – video, audio, still photography, scripts, schedules, and etc. This could also include meta-data such as schedules, itineraries, call sheets, etc. Loss of these assets could result in loss of competitor advantage, so the most important aspect of these assets is their confidentiality and security from release before publication.

Protection from modification and deletion is also as important, if not more so, than the confidentiality of media. Defacement can result in embarrassment for the creators. Encryption of files to extract a ransom from the owners is a tactic being increasingly used by cyber criminals and can result in loss of money or loss of content in many cases, even though the ransom has been payed.

Barriers to prevent outsiders from entering the network should be in place. Protections from erasure and modification as well as disclosure should be taken into consideration in case of intrusion into the system. Backups are always a significant concern, and for media assets, digital watermarking and logging of events performed within the asset management system are important to be able to identify insiders who might make off with confidential information.

ACCESS AND AVAILABILITY

Loss of access, from network glitches, denial of service attacks, or misconfiguration can deny users the ability to perform all the actions that an asset management system is supposed to provide. This downtime can lead to loss of productivity and ultimately money.

PERSONALLY IDENTIFIABLE INFORMATION (PII)

While an asset management system may not contain what is considered typical PII— personal addresses, phone numbers, credit card numbers, and such—it will likely contain information such as email addresses. This can lead to attacks against users of the system and might lead to damages from such activities.

REPUTATION

Downtime and lost productivity, disclosure and tampering of assets that are supposed to be protected, and harboring of attacks, i.e. using the system to attack its users, can figure into the loss of reputation of a company, whether it be the client using the asset management application or the application provider. This asset is abstract, yet it should implicitly be assumed that exploit of any vulnerability could lead to damage of this valuable asset.

Threat Actors

Any robust defensive model requires a thorough understanding not only of the system to be protected, but also of the adversary. Too often security practitioners design and review systems as if in a frictionless vacuum; this simplification can lead to solutions that have little relevance to real-world operations. An adversary-focused threat model allows companies to manage risk by directing resources toward threats and vulnerabilities that pose the greatest risk.

NATION STATE INTELLIGENCE (ADVANCED PERSISTENT THREAT)

Nation states represent the most capable threat actors in the realm of computer network exploitation. Beyond having the largest budgets and payrolls, nation states have unique capabilities in terms of access to supply chains and human agents. They can rely upon all of a nation's resources, such as intelligence infrastructures, to gather as much data as possible against an enemy, engage in active information system damage, and to set specific objectives for their hacking program.

Motivations that drive such an adversary typically revolve around national interests: espionage, i.e. extracting secret and sensitive information from another nation's government entities, subversion of information systems, and support of military operations. However, some nations are involved in extraction of proprietary information from commercial interests to boost their own economy by giving businesses in their own country an advantage. Furthermore, influencing social attitudes has been seen as a way to gain advantage.

If targeted by such an adversary, attacks will be clever, subtle, i.e. uses steps to avoid detection, yet at the same time aggressive. Being the most capable adversary, zero-day exploits, i.e. exploiting a previously unknown vulnerability, are not uncommon. Such a persistent threat underscores the adage of security as a process – one must constantly keep security in mind, watching audit trails, acting on security events, managing software updates, reviewing configurations, and enforcing policies.

While most in the private sector do not expect to be targeted by such an adversary, we have seen that these expectations are not always accurate; for example, the cyber-attack on the Sony Pictures film studio in 2014 is believed to have been sponsored by a nation state. Furthermore, many experts agree that the problem of state sponsored hacking is on the rise.

On the other hand, preparing for such sophisticated attacks will also furnish the security posture necessary to counter most other threats. For the most robust security, focus technical measures to defend against this type of adversary. A risk analysis should consider, but might decide to ignore this type of threat, or handle it in other non-information security ways such as acquiring insurance.

CORPORATE SPONSORED ESPIONAGE

Corporations, particularly certain overseas corporate environments, will at times utilize espionage techniques to gain advantage over competitors or save on research and development. Corporations have significant budgets and are able to hire professional teams to conduct computer network exploitation of their competitors' network. Targets usually comprise proprietary information such as business plans.

Network security can add a layer of defense to help mitigate such espionage. Firewalls should always be turned on and set to allow traffic only from ports that are actively being used and only for sources and destinations that are actually needed; all other access should be denied. This is an example of defense in depth – layers of different mitigations such that one layer will pick up something that has gotten past some other component (other layer) for any reason, whether it be a bug, malicious action, misconfiguration or a setting that was forgotten to be turned on or off.

HACKER GROUP (E.G., ANONYMOUS)

This adversary is a hacker organization analogous to the group "Anonymous." They are motivated by socio-political activities, pride, fun, and notoriety. These groups can have extensive membership, but it is likely that only dozens make up the core of this organization that pose a threat. They choose high profile, opportunistic targets over high value targets, and typically disclose stolen information rather than use it for identity theft or profit. The goal is notoriety and amusement, generally at the expense of, or to target the victim.

It is likely that most groups fall under the radar for this type of threat actor, however, any entity that is well-known could draw the ire of such malcontents, especially if the entity is on one side of the spectrum politically, socially or, in fact, taken a side of any popular issue. Even when staying in the center of issues, it may not be enough to one side to satisfy such actors. Attacks will generally include web-site defacement, "doxing", releasing embarrassing internal communications, denial of service, holding data ransom by encryption, or other acts of cyber-vandalism.

As with all threats, a well-established security program with ongoing review and adjustment is necessary to avoid being the target of attacks from such groups. Giving in to such a threat is not advised to appease this type of hacker group if that is the only reason for such an action. Decisions should be made after a full consideration of all the factors. It is better to be prepared so that attacks from this type of group are not successful to begin with.

CRIMINAL GROUP (E.G., RBN)

This adversary is a hacker organization analogous to the group the "Russian Business Network" (RBN). They are a for-profit organized crime syndicate focusing primarily on cyber-crime activities, such as identity theft, for hire targeted denial-of-service attacks, black market exchange of bot-nets, exploits and other information, illegal hosting of copyrighted materials and illegal pornography. They are simply motivated by money. These groups can have extensive membership, and due to available monetary resources can afford to hire skilled counterparts, fund exploit research or purchase exploits, and fund attacks in general. As cyber-crime is a business, targets of high-value will be chosen, costs to compromise assets calculated, and a bottom line decision made as to the worthiness of attempting an attack.

Organizations as well as individuals can be the target of such a group. "Spearphishing" – a "phishing" attack targeted at a specific individual or group where the attacker already knows something about the target and uses this in the phishing attack to make it more compelling for the target to fall into their trap – is an attack vector that such a group will often use. Education programs identifying suspicious links and files should be a part of every security program.

INDIVIDUAL HACKER

THREAT MODEL

The classic individual hacker is an explorer. They are generally motivated by the challenge of gaining access to restricted systems but may also work for financial gain. If she is financially motivated she will generally target banking and personal information. Individual Hacker capabilities vary widely from a so-called script kiddy that is only able to apply existing tools, all the way to a highly professional individual who is capable of custom exploit development. Generally, these threat actors have limited budgets and time; however, if properly motivated by a perceived slight or challenge, they may be persistent.

"Script kiddies" generally rely on exploiting known vulnerabilities that have long since been secured, but have not had these fixes applied to the system under attack. It is also a fact that more sophisticated threat actors also try exploiting known vulnerabilities first – known as low hanging fruit. This is why it is imperative to have an update and patch program: this should do away with low hanging fruit.

INSIDER

The insider threat encompasses any employee, contractor or other individual who has some level of authorized access to the system being assessed. Often, these are the most pernicious threats, as they have already bypassed the outer layers of defense and have a foothold on the system. In the worst case an insider threat may have administrator, root, or other elevated access. They can be disgruntled employees, disgruntled ex-employees who still have system access, or an employee who is working for a competitor.

Another class of the insider threat, not normally thought of as such, includes mistakes made by normal employees through lack of education, carelessness and inadequate policy controls. Keeping this threat in mind, it should go without saying that education, quality control, policy development, including authorization and authentication controls, and enforcement, along with contingency plans are part of a rational information security program that can help counter such threats.

SECURITY GUIDELINES

Below is a list of principles that underlie the precepts given later as security controls. These principles should be kept in mind throughout the lifetime of a MAM/DAM system – from initial conception, through design, deployment and operation.

In the guidance section below this one, recommendations will show how these principles are applied in configuring an asset management system based in Azure for production. In many cases, Azure offers options that leverage one or more of these principles of secure computing. This will be pointed out below in each security control. It remains, however, for the administrator configuring the system to use these options and to use them properly. This guide will prepare administrators for this endeavor.

Security Design Principles

Design analysis considers the high-level architecture of the system within its operational context and offers recommendations for the hardening of that system. Security of information systems should be considered a lifetime activity, and especially during design phases; it is much less costly to make changes before a system is built, than later.

Keep in mind that information security is a multi-faceted endeavor. It is instructive to think of a three-dimensional object with many sides. An adversary can choose any side to attack and using automation will attack all sides at once and, for some adversaries, constantly. If even just one side is left open, even for an instant, an attack will succeed. Identifying attack surfaces and finding vulnerabilities to exploit is the aim of a security assessment, which should be completed before a system is deployed. Core tenets of secure design that should be considered during development and assessment are presented below.

LEAST PRIVILEGE

The principle of least privilege refers to the principle that a task should be accomplished with the absolute lowest level of privilege required. Do not give a user more privilege than is needed to do their job. If the worst should happen and the user account is compromised, then damage done will be minimized if this principle is followed. There are many instances below where this principle is used to harden systems employed in the Azure Cloud.

WELL-DEFINED TRUST MODEL

A trust model clearly defines the trust assumptions made by the system and may highlight improper assumptions on the part of the designers. Using unauthenticated API calls, failure to verify digital signatures and certificate chains, and non-validation and sanitization of user input leading to injection vulnerabilities are all examples of misplaced trust. These apply to application development but there are also areas detailed below that deal with Azure configuration that can allow trust violations such as caching login credentials locally.

SECURE BY DEFAULT

A system is secure by default if the out-of-the-box settings put the system in a secure state. A corollary is that the secure state must be the easiest state to obtain and maintain—as users will typically choose convenience over security.

An example of this principle at work would be hardening virtual machine images before use and subsequently using only these hardened VM images.

SECURITY GUIDELINES

AUTOMATION

Automating activities versus letting users carry out security related actions is a corollary to the secure by default principle. User session log off after a configured period of time is an example of automation of security events.

AUDIT

Audit is a critical part of the system that assists in recovery, investigation and deterrence. Trust trusted users, but verify their actions. Azure offers auditing capabilities in association with most, if not all, of its services.

DEFENSE IN DEPTH

Defense in depth seeks to array layers of defensive measures such that the failure of any one component will not lead to total system compromise. The classic physical world example is the concentric walls of a fortification. The use of firewalls is a good example of this in the information network world. Use of firewalls and other mechanisms to close off network ports is an example of such a measure.

SEPARATION

Separation refers to segregating data storage, communication and processing from other unrelated data. This minimizes the likelihood of a compromise in one system from affecting another and helps prevent unauthorized access. Network segmentation is an example of this type of separation.

FAIL-SECURE AND FAILSAFE

Fail-secure refers to the tendency for a failure of the system to leave it in a *secure* state as opposed to an insecure state. This should not be confused with failsafe, which refers to systems that offer *safety* in the event of a failure or disaster. We are generally more concerned about fail-secure in the information security world, except where lives are endangered or injury is possible. An example of this from below is the failure of an Azure Batch job. Should such a job fail, we recommend that source code, configuration and logs be inspected to find out why before restarting the job.

BACKUP

Backup of data is a topic that should go without saying. The idea and techniques have been around since data automation first became available, and sensible practices for backup are well known. Backups, besides helping to recover from disaster or mistakes, when done according to best practices, can also save money and embarrassment after a cyber-attack. It therefore is a security matter as well as a best information technology (IT) practice.

Azure offers different types of storage and an archive services. Specific suggestions appear below.

USE OF CRYPTOGRAPHY

It can be said that cryptography is an arcane mathematical exercise, but unfortunately, it underpins many security constructs. As Bruce Schneier has proclaimed, “Cryptography Is Harder Than It Looks[1]”. Because of this, it is considered folly to create your own cryptographic algorithm or protocol unless you are an expert and have submitted your creation for scrutiny by academic and/or government authorities.

[1] https://www.schneier.com/blog/archives/2016/03/cryptography_is.html.

SECURITY GUIDELINES

It is imperative that well-known, accredited or certified cryptographic solutions be used for cryptography to remain secure. Fortunately, with Azure storage, processing and key management services, cryptography is an integral part waiting to be used. However, configuration is still vital here.

IDENTIFICATION, AUTHENTICATION, AND AUTHORIZATION

Identification (claiming a valid identity), authentication (verifying that identity, the two generally abbreviated as I&A), as well as authentication (granting permissions to users) is a topic that should be approached with care. Azure offers robust mechanisms for authentication services via Active Directory. It is strongly recommended that multi-factor authentication (MFA) be used versus one-factor passwords.

A closely related topic is that of session authentication and management. For example, sessions should last no longer than needed and APIs should require time-limited tokens with each function call.

SOFTWARE UPDATE PROGRAM

Many attacks, especially from opportunistic script kiddies, are done by exploiting vulnerabilities in software that have already been fixed by the developers, but have not had these fixes patched into a deployment of the software by those who are using it. It is very important to apply patches to all software that is being used, whether it's an application, operating system or library software.

PERIODIC THIRD-PARTY ASSESSMENT

It has previously been stated that security is an ongoing process, not a one-time blessing of a system that makes it secure forever. Previously unknown vulnerabilities surface, configurations change, software is upgraded and people make mistakes. In the ever escalating war between hackers and administrators, the bad guys just get worse and the administrators need to keep up. It is not easy to keep up with the bad guys when it is hard enough just managing information systems.

Based on risk analysis, re-assessment of a system's security posture should happen periodically – every year, six months, three months or whatever period is dictated by the analysis. Furthermore, a best practice is to have security re-assessments done by experts, which usually means bringing in a third-party to accomplish the assessment.

Security Controls

Security controls, as outlined in the executive summary, are presented here for asset management systems running within Azure. They are grouped by category according to specific Azure component. These Azure specific security controls consider the high-level architecture of the system within its operational context and offers recommendations for the hardening of that system. A detailed description of each security control, recommendation, and where to find further documentation for each follows in this section.

Azure Portal

The Azure management portal is a web based interface that allows control of Azure accounts and their configuration. The creation and initialization of an Azure based asset management system begin here. The portal allows administrators the ability to configure the overall architecture of the system which is critical to starting off correctly. Each control detailed below is a direct example of the most important security design considerations – least privilege, separation of data, identity and authentication.

SECURITY CONTROL: Prohibit use of personal Microsoft accounts for administration

Access to Azure is possible through two types of Microsoft account (formerly known as Microsoft Live ID) and a work or school account, which is an account stored in Azure Active Directory (AD). Any Microsoft Azure account can be used to set up and administer Azure portal. If an employee uses a personal (@hotmail.com) account to set up the Microsoft Cloud infrastructure, then the employer may have difficulty gaining control of the account if the employee later resigns or is terminated. Further, employers are unable to maintain security policies on personal accounts or otherwise audit the accounts' security.

RECOMMENDATION: PROHIBIT USE OF PERSONAL ACCOUNTS FOR ADMINISTRATION
It is mandatory for users to sign up for Azure as an organization and use an enterprise account to manage resources in Azure. Enterprise accounts are preferred because they can be centrally managed by the organization that issued them, they have more features than Microsoft accounts, and are directly authenticated by the Azure AD service.

This measure supports the least privilege and auditing principles of security.

Documentation: Azure accounts are managed via two separate web interfaces, 1) Azure portal ("Developer Portal") and 2) Usage and Billing. The Azure portal allows a user to configure and use Azure services and are described in detail here: https://azure.microsoft.com/en-us/features/azure-portal. The usage and billing interface is designed for tracking usage and Azure management actions, including Azure subscriptions.

SECURITY CONTROL: Use separate Azure subscriptions to segregate work

Though Azure subscriptions are a container for billing, they can also be used as a security boundary. Subscriptions can be used to limit who can access Microsoft Azure services associated with that subscription. Each Azure subscription is

associated to a single Azure Active Directory. The subscription itself, governs access to and use of the Windows Azure services that are subscribed to. The subscription administrator manages the subscription Azure service through the Azure Portal.

RECOMMENDATION: USE SEPARATE AZURE SUBSCRIPTIONS TO SEGREGATE WORK

It is mandatory that a separate Microsoft Azure subscription be used for each different production, client, and asset management workflow to limit access. Segregating workflows in this way will provide administrative anatomy and individual resource monitoring.

This security control directly supports the separation security principle.

Documentation: An understanding of the Azure subscription model is imperative for securing Azure instances. The fundamental unit of work for a developer is a subscription. An Azure subscription grants a user access to the Azure services and management portal. Azure subscriptions are described in detail here: https://account.windowsazure.com/Subscriptions.

SECURITY CONTROL: Logically relate and manage deployments with resource groups

In Azure, most things can be considered a resource. A virtual machine (VM) is a resource, the network adapter interface (NIC) used by a VM is a resource, the public IP address used by a NIC is a resource, the VNet that the NIC is connected to is a resource. Logically group related resources such as storage accounts, virtual networks, and virtual machines (VMs) to deploy, manage, and maintain them as a single entity.

Azure provides a logical resource group as a container that can hold related resources for an Azure solution. A resource group is a container for resources that share a common lifecycle. It is beneficial to logically group resources that are inter-related or share rendering workflow. For example, a virtual machine that depends on a specific storage account should be in the same Batch resource group. The resource group can include all the resources for the render workflow, or only those resources that you want to manage as a group.

RECOMMENDATION: USE AZURE RESOURCE GROUPS

Use Azure resource groups as logical containers for resources associated with production. Various asset management system resources grouped together in resource groups make it easier to manage the them as a unit. A resource group can contain resources from multiple regions, if the resources belong to the same subscription. When deploying resources assign them to a logical resource group. It is suggested that resource groups be created for each asset management workflow deployment which will combine storage accounts, batch services, virtual networks and their subnets, VMs, load balancers, etc.

During the lifecycle of the resource, manage the resources as a function of the related resource group, specifically the Azure Resource Manager can be used to manage resource groups.

Documentation: Resource group implementation is described here: https://docs.microsoft.com/en-us/azure/virtualmachines/windows/infrastructure-resource-groups-guidelines.

SECURITY CONTROL: Avoid concurrent access from multiple locations

A user can be logged into the platform concurrently from multiple computer systems, and there does not appear to be a feature allowing users or administrators to automatically log out all other sessions upon successful authentication. Highly security-conscious enterprises often desire such a feature to reduce the possibility that a user may inadvertently leave an active session unattended in a non-secure location (e.g., conference room computer) and login again elsewhere. The cloud platform does not offer such an automatic feature.

AZURE CAVEAT: NO ABILITY TO PREVENT ACCESS FROM MULTIPLE LOCATIONS
Azure does not offer the native ability to prevent access to an account from multiple locations.

RECOMMENDATION: MANDATE MANUAL USER SESSION TERMINATION WITH POLICY
Provide user policy that mandates users immediately terminate their session when access to Azure Cloud is no longer needed.

This security control supports session management and robust authentication mechanisms.

SECURITY CONTROL: Enforce custom session inactivity termination

User sessions are not terminated after a certain time of inactivity. Failing to provide an inactivity timeout of sufficiently short duration leaves Microsoft user sessions susceptible to unauthorized access in conjunction with unrelated client-side attacks.

AZURE CAVEAT: NO ABILITY TO FORCE SESSION TERMINATION AFTER A PERIOD OF TIME
Azure does not offer the native ability to force a user's session to terminate in a given period of time. If a user does not manually terminate their session, then they will remain logged in indefinitely.

RECOMMENDATION: MANDATE MANUAL USER SESSION TERMINATION WITH POLICY
Provide user policy that mandates users immediately terminate their session when access to Azure Cloud is no longer needed.

This security control supports robust session management and authorization. Note that this is the same recommendation as the previous one.

SECURITY CONTROL: Enforce two-factor authentication for portal access

Multi-factor authentication (MFA) is a method of computer access control in which a user is granted access only after successfully presenting several separate pieces of evidence to an authentication mechanism.

RECOMMENDATION: ENABLE TWO-FACTOR AUTHENTICATION
Administrators must enforce two-factor (2FA) authentication for management of the Azure environment.

This recommendation supports robust identification and authentication.

Documentation: Azure does not support MFA for specific Azure developer interfaces, though multi-factor authentication is available on general Microsoft accounts. The Azure administrator should enable 2FA on all accounts which access the enterprise Azure account. Individual users can enable 2FA by accessing account information here: https://account.microsoft.com.

Azure Networking

Azure provides a number of networking products and capabilities allowing users to tailor their connection to the services they would like to provide. Much of the configuration of securing Azure networking involves separation of data. This is largely accomplished by using the Azure Network Security Group (NSG) feature.

SECURITY CONTROL: Use Network Security Groups (NSG)

While Azure heavily restricts incoming traffic from the Internet, it is more permissive about internal traffic—essentially allowing open communication between all VM instances within the same zone. While the default endpoint security features are a useful mechanism for securing Azure VMs they have limited functionality. Network Security Groups (NSG) secure both inbound and outbound access to both Azure VMs and Azure VNets, similar to a traditional firewall. NSG rules are defined with a standard five-tuple definition (source network, source port, destination network, destination port, protocol) as well as name, type, priority, protocol and access (allow or deny).

RECOMMENDATION: SECURE TRAFFIC FLOW WITH AZURE NETWORK SECURITY GROUPS
Network Security Groups must be employed.

This recommendation supports defense-in-depth.

Documentation: Filtering of network traffic is a security and workflow function. Traffic filters can be deployed to direct or restrict access to rendering farm subnets based on a production or a source. A network security group is a list of security rules that allow or deny network traffic to resources connected to Azure Virtual Networks (VNet). AzureNetwork Security Groups are described here: https://docs.microsoft.coms/en-us/azure/virtual-network/virtualnetworks-nsg.

SECURITY CONTROL: Isolate virtual appliances in their own subnet

When using virtual appliances, such as a firewall, WAN accelerator, Active directory server, or VPN gateway in an Azure, isolate them in their own gateway subnet. Virtual appliances are useful to create routes between Azure cloud resources and on-premises data centers.

RECOMMENDATION: USE A GATEWAY SUBNET WITH USER-DEFINED ROUTING
Use a gateway subnet with a user defined routing mechanism to isolate networking appliances in their own dedicated private network subnets. Specifically, to secure these services and appliances, prevent direct internet connectivity by placing them in a separate subnet with an NSG acting as a firewall. Additionally, close all ports on the appliance or service servers except those necessary for authentication, authorization, and server synchronization.

This recommendation supports defense-in-depth.

Documentation: Asset management requires implementing a secure hybrid network that extends the on-premises network and datacenter to Azure. The user defined routing mechanism in the gateway subnet filters or blocks all user requests other than those received from the on-premises network.

The Azure network DMZ architecture is described here:
https://docs.microsoft.com/enus/azure/architecture/reference-architectures/dmz/secure-vnet-hybrid and
https://docs.microsoft.com/enus/azure/architecture/reference-architectures/dmz/secure-vnet-dmz.

SECURITY CONTROL: Apply a multi-tiered architecture for VNets

A three-tiered virtual network has front, mid and back-end network segments to create isolation between various types of assets. In rendering workflow, place asset management work machines in the backend, while placing authentication and traffic shaping software servers in the front-end.

The front end, which contains web servers in its own subnet, directly faces the internet. The mid-tier, which contains business logic, does not have direct internet access, either inbound or outbound, but can be reached from the front-end subnet. The back-end tier, again in its own isolated subnet, contains persistent data such as a database system or storage, and can communicate only with the middle-tier. In addition to these subnets, a Bastion server is set up for management of these subnets by administrators.

RECOMMENDATION: APPLY A THREE-TIERED NETWORK ARCHITECTURE WITH VNETs
Place asset management work machines in the backend, while placing authentication and scheduler software servers in the front-end. Apply different Network Security Groups (NSGs) for each subnet.

This recommendation supports separation as well as defense-in-depth.

Documentation: An Azure VNet is a logical isolation of the Azure cloud dedicated to your subscription. As mentioned before, asset management workflow should be deployed on separate subscriptions to segregate productions, vendors, or artists.

Azure virtual networks are described here: https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networksoverview. Furthermore, a reference architecture for deployment of N-tier applications is described here: https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/virtual-machines-windows/n-tier.

SECURITY CONTROL: Create separate VNets for production

Create separate VNets for each production (even if they use the same IP address space), to isolate different workloads from one another. If possible, create VNets in regions where the majority of artist or resources are located.

RECOMMENDATION: USE AZURE NETWORK SECURITY GROUPS
Create network segmentation for various asset management workflows with NSGs and network appliances.

This recommendation supports separation.

Documentation: As mentioned before, asset management workflow should be deployed on separate subscriptions to segregate productions, vendors, or artists. Azure virtual networks are described here: https://docs.microsoft.com/enus/ azure/virtual-network/virtual-networks-overview.

SECURITY CONTROL: Limit default Network Security Group VNet communications

While having the default network security group and firewall in place when setting up Azure heavily restricts incoming traffic from the Internet, it is more permissive with regard to internal traffic—essentially allowing open communication between all VM instances within the same zone. It is important that customers realize the need to harden the default firewall before allowing an environment to shift into production.

AZURE CAVEAT: DEFAULT VM FIREWALL TOO PERMISSIVE WITH INTERNAL TRAFFIC
The firewall settings for the default VMs that Azure offers set to allow all traffic within the internal network.

RECOMMENDATION: CONFIGURE THE FIREWALL TO RESTRICT INTER-VM COMMUNICATION
Administrators should remove the default firewall rule, allowing inter-VM communication, and replace it with a deny-all policy with specific IP/protocol/port-base exceptions.

This recommendation supports defense-in-depth.

Documentation: Azure Network Security groups are the primary method of filtering and restricting within the Azure platform. Azure Network Security Groups are described here: https://docs.microsoft.coms/en-us/azure/virtualnetwork/virtual-networks-nsg.

SECURITY CONTROL: Tightly configure endpoints

Endpoints are a key feature of Azure VMs deployed using the Azure Portal, similar in functionality to Network Address Translation (NAT). An endpoint is configured with a public port (TCP or UDP) and a private port (TCP or UDP); the public

port is the port open to the internet, while the private port is the port open on the Azure VM for a configured application or service.

RECOMMENDATION: CONFIGURE ENDPOINTS WITH ACCESS CONTROL LISTS FOR AZURE VMs
Configure the endpoints securely by the use of an Access Control Lists (ACL), which restrict access to an endpoint based on a series of permit/deny rules.

This recommendation supports the defense-in-depth and least privilege principles.

Documentation: Though Azure Network Security Groups (NSG) provide traffic filtering at the Azure subnet layer, it is important to apply defense in depth principles, and apply access control at VM endpoints. Endpoint security and port configuration is described here: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/endpoints-in-resource-manager.

SECURITY CONTROL: Do not use deprecated cryptography for IPSec VPNs

In contrast to SSL/TLS, configuring the set of permitted cipher suites at each end of an IPsec connection can be a manual and time-consuming process. The following concerns affect the configuration of the Cloud IPsec VPN:

IPSEC CONFIGURATION: IPSEC VPNs HAVE THE FOLLOWING CONCERNS

- *IKEv1 protocol supported*. The Cloud IPsec VPN, for compatibility, supports both IKE version 1 (introduced in 1998) and IKE version 2 (introduced in 2005). One of the goals of IKEv2 was to improve security over IKEv1, including cryptographic weaknesses[2]. Specifically, the IKEv1 supports 3DES and SHA1 (SHA128) as the encryption and hashing algorithms respectively.
- *HMAC-MD5, supported (IKEv2).* The Cloud IPsec VPN allows *HMAC-MD5* to be used for integrity checking. HMACMD5 is deprecated due to weaknesses in the underlying MD5 algorithm[3].
- SHA1 *supported (IKEv2).* The Cloud IPsec VPN allows SHA1 to be used for integrity checking. SHA-1 has been practically broken and is considered insecure and ineffective[4].
- *DES, 3_DES, supported (IKEv2)*. The Cloud IPsec VPN allows DES, 3_DES to be used for data encryption. DES is inherently insecure, while Triple-DES has much better security characteristics but is still considered problematic.

RECOMMENDATION: CONFIGURE IPSEC TO AVOID DEPRECATED CRYPTOGRAPHY
Configure the IPsec VPN to avoid use of deprecated or inherently insecure protocols and modes of operations. Administrators should configure their IPsec clients in accordance with their security policies.

[2] RFC 4306 Appendix A, https://tools.ietf.org/html/rfc4306#page-96
[3] http://tools.ietf.org/html/rfc6151
[4] https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

This recommendation supports cryptographic security principles.

Documentation: Azure VPNs support three cross-premises and VNet-to-VNet configurations:

1. Site-to-site
2. Point-to-site
3. ExpressRoute

Asset management workflows will require site-to-site VPN gateways or an ExpressRoute connection for data interface between the cloud and on-premises. Planning and design for Azure VPN gateway devices are described here: https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-plan-design. Specifically, Azure VPN IPsec and IKE parameters are described here: https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-devices.

Azure Access Control

Azure subscriptions are associated with one Azure Active Directory (AD) directory; that is, users, groups, and applications from that directory can manage resources in the Azure subscription. Administrators can grant access by assigning the appropriate RBAC role to users, groups, and applications within a certain scope.

SECURITY CONTROL: Employ custom Role-Based Access roles to manage user access

In many cases studios have multiple parallel workflows for various productions and vendors involving a myriad of artists and resources. To manage user access, Azure offers customizable Role-Base Access control (RBAC).

RBAC functions enable fine-grained access management for Azure resources. Using RBAC, an administrator can grant a minimal level of controlled level of access to user accounts to perform its jobs. RBAC makes it feasible to set up minimal privileges for users to be able to perform their functions.

RECOMMENDATION: USE AZURE RBAC WITH CUSTOM ROLES TO MANAGE USER ACCESS
Users must be granted the lowest level of required access with minimal operational privileges. Additionally, the Azure built-in role definitions are constantly evolving, and the administrator must define and use custom roles rather than using the built-in roles.

Another possible solution is to use an Azure service principal as an identity for user-created apps, services, command-line access, and automation tools to access specific Azure resources. Service principal is a security identity similar to a user with a specific role, who has tightly controlled permissions to access only designated resources. This recommendation directly supports the least privilege principle.

This recommendation supports the access control principles.

Documentation: Azure RBAC built-in roles only support the management operations of the Azure resources in the Azure portal and Azure Resource Manager APIs, thus, it is imperative that custom roles are defined for asset management operations, using attributes of a custom role.

Azure role based access control functionality is described here: https://docs.microsoft.com/en-us/azure/activedirectory/role-based-access-control-configure.

Service principal identity is described here: https://docs.microsoft.com/en-us/powershell/azure/create-azure-serviceprincipal-azureps?view=azurermps-3.8.0.

SECURITY CONTROL: Extend on-premises identity management for access control

Managing identity is as important in the Azure cloud as it is on-premises. Studios use on-premises Active Directory (AD) systems to store directory data and manage communication between users and resources, including user logon processes, authentication, and directory searches. When scaling out media asset management to Azure cloud, the cloud resources are being used as an extension of the on-premises datacenter—in this scenario there are applications which require a domain controller to handle authentication and authorization.

RECOMMENDATION: USE THE AZURE ACTIVE DIRECTORY SERVICE
Use an Active Directory service in the cloud. The Windows Server AD is running in VMs created using Azure Virtual Machines and the AD VMs should be grouped into a virtual network connected to an on-premises datacenter using the Azure Virtual Network.

The virtual network carves out a group of cloud virtual machines that interact with the on-premises network via a virtual private network (VPN) connection which allows the AD Azure virtual machines to look like just another subnet to the on-premises datacenter.

This recommendation supports the access control principles.

Documentation: Azure Active Directory services can be implemented in multiple architectural patterns, including Software as a Service (SaaS), platform's IaaS technologies and global enterprise. Information about Azure active directory is described here: https://docs.microsoft.com/en-us/azure/active-directory/active-directory-whatis.

SECURITY CONTROL: Do not use the deprecated Azure Access Control service

Though the Azure Access Control service (ACS) can provide a straightforward way of authenticating and authorizing users to gain access to web applications and services, its functionality has been merged with more advanced and centralized Azure Active Directory services. Starting June 30th, 2017, Microsoft has restricted creation of new ACS namespace creation.

RECOMMENDATION: DO NOT USE THE AZURE ACCESS CONTROL SERVICE
Use the Azure Active Directory instead of ACS. Azure Active Directory natively supports many of the scenarios enabled by ACS.

This recommendation supports the access control principles.

Documentation: Depreciation of Microsoft Azure Access Control Service (ACS) is discussed here: https://azure.microsoft.com/en-us/blog/acs-access-control-service-namespace-creation-restriction.

Azure Key Vault

Azure provides the Key Vault service for controlling cryptographic keys, digital certificates, and passwords and the access to these items. The use of Key Vault is integrated into all of Azure's other services.

SECURITY CONTROL: Use a separate Azure Key Vault for each production

Though Azure Key Vault is a generic container for keys and secrets, it should not span across multiple disjoint productions. Each production has a defined lifecycle with varying priority, protection and collaboration requirements. Production specific key vaults should be used to limit access to protected assets associated with the production lifecycle.

RECOMMENDATION: USE SEPARATE KEY VAULTS FOR PRODUCTIONS
It should me mandatory to use separate Azure Key Vaults for each production to limit access. Segregating workflows through Key Vault will provide administrative anatomy and individualized resource monitoring.

Each key vault will consist of a collection of cryptographic keys and cryptographically protected secrets logically bundled together relevant to a specific production or workflow.

This recommendation supports the cryptographic principles.

Documentation: Understanding of the Azure Key Vault model is imperative to securing Azure instances. The fundamental unit of work for a developer is a subscription. An Azure subscription grants a user access to the Azure services and management portal. A subscription is allowed to have multiple vaults in multiple geographic areas. Azure subscriptions are described here: https://account.windowsazure.com/Subscriptions.

SECURITY CONTROL: Use Azure Key Vault permissions to manage access

During a production, various parties will require access to the production key vault. Azure has built-in permissions for access to this. In conjunction with Active Directory, these permissions enable fine-grained access control to a specific Azure Key Vault. It is feasible that in workflows the applications or users will only need limited read-only, encrypt or decrypt access to a specific key vault in order to be able to perform their functions.

RECOMMENDATION: USE LEAST PRIVILEGE IN ASSIGNING ACCESS TO THE KEY VAULT
Services and users accessing the Azure Key Vault must be granted the lowest level of required access with minimal operational privileges.

As mentioned in the RBAC security control previously, a possible solution is to use service principal as an identity for user-created apps, services, command-line access, and automation tools to access specific Azure resources.

This recommendation supports the least privilege security principle.

Documentation: Azure RBAC built-in vault permission to limit access to management operations of an Azure Key Vault resources and permisssions are described here: https://docs.microsoft.com/enus/powershell/module/azurerm.keyvault/set-azurermkeyvaultaccesspolicy?view=azurermps-4.0.0.

SECURITY CONTROL: Segregate data and key/secret owners

In many cases studios will employee a security administrator or team to manage Key Vaults. The security administer or team will be responsible for initializing the vault, furnishing and maintaining keys, and controlling access to the vault, keys and secrets. The data owners will use an assigned Key Vault to encrypt, decrypt, or wrap keys and manipulate digital assets. It is imperative that key and data owners have mutually exclusive permission on a key pair.

RECOMMENDATION: ASSIGN USERS, SERVICE PRINCIPLE, KEY OWNER OR MANAGER RIGHTS

Assign these key level functions.

- Create
- Delete
- Update
- Import
- List versions
- Get
- Backup
- Restore
- Key Wrap and Unwrap

RECOMMENDATION: ASSIGN DATA OWNERS AND SERVICES RIGHTS

Assign these key level functions.

- Encrypt and Decrypt
- Sign and Verify

These recommendations support the use of cryptography, least privilege and access control principles.

Documentation: Key Vault allows an organization to securely manage and protect cryptographic keys and secrets which can be used by cloud-enabled applications and services. Key access policies are different to secret access policies.

Azure Key operations are described here: https://docs.microsoft.com/en-us/rest/api/keyvault/vaults.

Azure Secret operations are described here: https://docs.microsoft.com/en-us/rest/api/keyvault/setsecret.

SECURITY CONTROL: Manage access to keys and secrets on a per key/secret case

A production key vault will contain keys to support various independent workflows. These workflows usually are executed in parallel in a compartmentalized manner. Azure offers key and secret operations to support this. Access to keys and secrets can be managed either on a per key/secret case or on the whole of Key Vault.

RECOMMENDATION: CUSTOMIZE ACCESS CONTROL LISTS FOR KEYS AND SECRETS
Provision keys and secrets to have a defined customized access control list. Keys and secrets should only be accessible to users or applications with a functional need.

This recommendation supports the least privilege principle.

Documentation: Key access policies are different than secret access policies.

Azure Key operations are described here: https://docs.microsoft.com/en-us/rest/api/keyvault/vaults.

Azure Secret operations are described here: https://docs.microsoft.com/en-us/rest/api/keyvault/setsecret.

SECURITY CONTROL: Audit all key management activity

In media and entertainment workflows, the Azure Key Vault will be used to control access to digital assets. The content owners want to track how, when and by whom the key vault is accessed to manipulate a protected asset. Key vault logs will indicate who accessed specific assets, and what operations were performed in near real-time. These logs can be piped into a higher level event management system to build a higher level data use model.

RECOMMENDATION: LOG ALL KEY VAULT ACTIVITY
Enable logging for all Key Vaults for the subscription in “Log specific Azure storage account”. Access to the log storage account should be restricted as per storage security controls to secure the logs by restricting access.

This recommendation supports the audit principle.

Documentation: Azure Key Vault logging can be configured in a deeply customizable manner. The Azure Key Vault logging is described here: https://docs.microsoft.com/en-us/azure/key-vault/key-vault-logging.

Furthermore, Azure Key Vault event log data can be used with the Log Analytics application. Logging data can then be used for data analytics. Key Vault logging data is described here: https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-azure-keyvault.

SECURITY CONTROL: Periodically rotate keys

Rotating keys is as important in the Azure cloud as it is for non-cloud applications. Applications can offload the storage of keys and secrets to Key Vault, applications request keys from the key vault as needed. This centralized approach allows the administer to update keys and secrets without affecting the behavior or structure of applications. Though

rotation of keys doesn't decrease the risk of keys being breached, it does limit the amount of data encrypted under a certain key, so for example, if a future key gets breached past encrypted data is safe. Also, rotation of keys limits the time an adversary has to break the current key.

RECOMMENDATION: ROTATE KEY VAULT KEYS EVERY 60 DAYS

There are various options for implementing a rotation strategy for values you store as Azure Key Vault secrets. Though secrets can be rotated as part of a manual process, they should be rotated programmatically by using API calls or automation scripts.

This recommendation supports cryptographic principles.

Documentation: Azure Key Vault key rotation using automation is described here: https://docs.microsoft.com/en-us/azure/key-vault/key-vault-key-rotation-log-monitoring.

Azure Command Line Interface

The Azure Command Line Interface (CLI) is a tool that has scripting, data query and other features, that can be used locally to handle Azure services and applications.

SECURITY CONTROL: Avoid caching session information

Users authenticate to Azure using a Microsoft account email address and password. To prevent a user from needing to retype credentials upon every invocation of the command line utility, the utility persistently caches the user's OAuth session information (excluding passwords) on the local system.

AZURE CAVEAT: CLI CACHES OAUTH SESSION DATA

Caching session information presents a security threat should the local system used to log in become compromised, even at a later date. Storing the OAuth access token, client secret, and other session information on the file system means that if an adversary gains access to the credentials file, or a copy of it, the attacker may then use the session information to gain unauthorized access to the user's account.

RECOMMENDATION: MANDATE DELETION OF CREDENTIALS AT SESSION TERMINATION

Create policy that mandates deleting of credentials when logging off.

RECOMMENDATION: USE A RAMDISK FOR CREDENTIAL STORAGE

Using a ram disk ensures that credentials are never written to disk. To do so, create a RAM disk and then create a symbolic link from the credentials file to a file located on the RAM disk. This can be done using the Linux tmpfs file system, or using analogous techniques on Windows or Mac OS X. This ensures that the

> credentials are stored only in volatile memory and lost upon unmounting the RAM disk or rebooting the machine.

This recommendation supports secure session management.

Documentation: Azure CLI 2.0 and Azure PowerShell can be used to manage and administer Azure resources from the command line, and for the creation of automation scripts that work with the Azure Resource Manager. Both interfaces have built-in methods for developer authentication. Azure CLI and PowerShell are described here respectively: https://docs.microsoft.com/en-us/cli/azure/overview and https://docs.microsoft.com/en-us/powershell/azure/getstarted-azureps?view=azurermps-3.8.0. Specifically, logging into CLI is described here: https://docs.microsoft.com/enus/azure/xplat-cli-connect.

Azure Batch

Azure Batch is used for repetitive tasks, for example, where artists continuously update rendering scenes until achieving desired results and quality. Storage of results and logging is important for workflow continuity and consistency.

SECURITY CONTROL: Use obfuscated URLs for batch accounts

To develop an application with Azure Batch APIs, an account URL is needed to access the Batch resources. A Batch account URL has the following format: `https://{account-name}.{region-id}.batch.azure.com`. A Batch account URL is specified at Batch service initialization.

RECOMMENDATION: OBFUSCATE BATCH URLs
Setup a URL that does not provide any underlying production name and purpose.

This recommendation supports the defense-in-depth principle.

Documentation: Azure batch can be used to efficiently run large-scale parallel and high-performance computing applications such graphic rendering. Azure batch is described here: https://docs.microsoft.com/enus/azure/batch/batch-technical-overview.

SECURITY CONTROL: Periodically update access keys

To authenticate access to an Azure Batch account from the client application, an account access key is required.

RECOMMENDATION: REGENERATE BATCH ACCOUNT ACCESS KEYS EVERY 60 DAYS
User must regenerate a Batch account's access keys periodically to account for workflows, and personal, and production environment changes.

This recommendation supports cryptographic principles.

Documentation: Azure batch keys are required to setup and execute jobs. Azure batch configurations are described here https://docs.microsoft.com/en-us/azure/batch/batch-api-basics#pool. Specifically, Azure batch key retrieve and regenerate are described here: https://docs.microsoft.com/en-us/azure/batch/batch-management-dotnet.

SECURITY CONTROL: Sanitize and destroy Batch accounts when no longer needed

Azure Batch service is not a billable item; the underlying resource pools are billable as they are used. Batch resource pools should be destroyed before reuse across vendors or productions.

RECOMMENDATION: SANITIZE AN DESTROY THE BATCH ACCOUNT WHEN PRODUCTION ENDS
The user must sanitize (securely delete all data) and destroy the batch account and underlying job resources once production has wrapped up.

This recommendation supports the defense-in-depth principle.

Documentation: Azure batch initialization, operation and management should be automated using scripts. The process overhead of sanitizing batch resources by re-instantiating resources pool can be limited using automation. Azure batch scripting using CLI 2.0 is described here: https://docs.microsoft.com/en-us/azure/batch/batch-cli-get-started.

SECURITY CONTROL: Create storage exclusively for specific Batch workflows

The Azure Batch service uses the associated storage account for storage and retrieval of application packages. Batch can automatically deploy the packages stored in the linked storage account to your compute nodes.

RECOMMENDATION: USE A SEPARATE STORAGE ACCOUNT FOR EACH BATCH WORKFLOW
Use separate storage accounts for each Azure Batch workflow to limit data comingling between productions and rendering setups.

This recommendation supports the separation principle.

Documentation: Azure batch initialization, operation and management should be automated using scripts. The process overhead of sanitizing batch resources by re-instantiating resources pool can be limited using automation. Azure batch scripting using CLI 2.0 is described here: https://docs.microsoft.com/en-us/azure/batch/batch-cli-get-started.

SECURITY CONTROL: Use security-validated application packages for batch workflow

Every batch workflow requires application binaries and supporting files that are required to execute an application. As part of the file upload operation, a user defines collections of application and input file paths as they exist on the local machine. Then the files are uploaded to the associated storage containers.

RECOMMENDATION: UPLOAD ONLY VALIDATED AND TRUSTED APPLICATION FILES
The application packages that are uploaded should be assessed prior to use, preferably through a defined security evaluation process.

This recommendation is an example of the secure by default principle.

Documentation: Azure batch compute pool executes user defined applications to perform batch jobs. User defined applications are stored in the Azure storage. Azure based solutions for application vulnerability scanning can be deployed. Azure application store and add-on are described here: https://docs.microsoft.com/enus/azure/storage/storage-create-storage-account.

SECURITY CONTROL: Use integrity checks on application packages for Batch workflows

Each Batch workflow requires application binaries and supporting files that are required to execute applications. Users upload application and input files to associated storage containers. The pool StartTask downloads these files to nodes as they join the pool.

RECOMMENDATION: PERFORM AN INTEGRITY CHECK ON DOWNLOADED APPLICATION FILES
Users must perform a predefined integrity check on the downloaded application files as part of the pool StartTask execution.

Integrity checks may be (in order of strength) as simple as a checksum, a cryptographic hash or a digital signature. It is important to verify the integrity of the application files with such checks.

This recommendation supports the cryptographic and defense-in-depth principles.

Documentation: Azure batch compute pool executes user defined applications to perform batch jobs. User defined applications are stored in Azure storage. Some level of static and run-time integrity validation of these applications is imperative. Run-time validation of these applications can be performed as part of the initial batch start task function. The optional start task executes on each node as that node joins the pool, and each time a node is restarted or reimaged. The initial task is described here: https://docs.microsoft.com/en-us/azure/batch/batch-apibasics.

SECURITY CONTROL: Hold workflow if Batch applications fail

If an application package deployment fails for any reason, the Batch service marks the node unusable, and no tasks will be scheduled for execution on that node.

RECOMMENDATION: INVESTIGATE BATCH FAILURES BEFORE RESTARTING
In the case that the Batch service marks a node unusable, a user must investigate the cause prior to restarting the package deployment.

This recommendation is an application of the fail-secure principle.

Documentation: Azure sample workflow processes are described here: https://docs.microsoft.com/enus/azure/batch/batch-technical-overview.

SECURITY CONTROL: Log Batch events for monitoring and diagnostics

The Azure Batch service emits log events for certain resources during the lifetime of the resource. Events like pool create, pool delete, task start, task complete, and others are included in the Batch diagnostic logs. Diagnostic logging is not enabled by default. It must be explicitly enabled for diagnostic logging for each Batch account.

RECOMMENDATION: ENABLE BATCH DIAGNOSTIC LOGS
Enable Azure Batch diagnostic logs to record events for resources like pools and tasks, and then use the logs for diagnostic evaluation and monitoring. This information will allow administrators to monitor health status and anomalies.

This recommendation supports the audit principle.

Documentation: Azure batch output storage and job logging is described here: https://docs.microsoft.com/enus/azure/batch/batch-task-output.

Azure Compute

This section deals with controls designed to secure virtual machines used within the Azure cloud environment.

SECURITY CONTROL: Use public key authentication for virtual machines

Azure Virtual machines allow authentication using a username and password or using public keys. Using public key based authentication is a good practice since it eliminates weak passwords on the system or hardcoded default passwords. Brute-force attacks are infeasible for those wanting to break into the system. Disabling login via passwords and requiring login keys is a configuration option when instantiating a VM or VM scale group in Microsoft Azure.

RECOMMENDATION: USE PUBLIC KEY BASED AUTHENTICATION FOR VIRTUAL MACHINES
Configure the SSH processes on Azure VMs to use public key authentication to protect against weak and hardcoded default passwords.

In addition to replacing passwords and their associated problems, with SSH the login key option can be setup to provide the following functions:

1. Only allow access from a specific IP; and

2. Only allow the accessing user/script to execute a certain command.

This recommendation supports robust authentication.

Documentation: Public-key cryptography provides a more secure way to log in to virtual machine. Use of SSH keys with windows for Linux VMs is described here: https://docs.microsoft.com/en-us/azure/virtualmachines/linux/ssh-from-windows.

SECURITY CONTROL: Use only hardened OS images for VM instantiation

Within the Microsoft Azure cloud platform there are a number of pre-built operating system images available to users for rapid deployment. The images provided include a range of Linux distributions (including CentOS and Ubuntu) as well as Windows 2008 R2 and Windows 2012 R2. In the current state, the OS images are supplied with default configurations. Some default configurations of OS images leave VM instances open to vulnerabilities, which could be exploited by publicly known, readily available proof of concepts and exploits. For example, the images may have their SSL/TLS configurations set up for maximum compatibility, rather than security.

RECOMMENDATION: HARDEN ALL USED OPERATING SYSTEM IMAGES
Harden all supplied operating system images using current best practices.

Hardening steps should include: closing unused ports, uninstalling unnecessary applications, updating software, configuring web servers using current best practices, and setting firewall rules.

When completed, use of these hardened images will support the secure by default principle.

Documentation: Editing workloads require a specific set of resources to optimally function. These resources include a base operating system, asset management software, and other supporting software applications. Azure provides a set of virtual machine images which can be deployed using the Azure Resource Manager to realize a consistent operating environment.

Descriptions of how to create Windows virtual machines using an Azure Resource Manager template are described here: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/app-frameworks.

Descriptions of how to create Linux virtual machines using template images is defined here: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/tutorial-manage-vm.

It should be noted that these images should not be considered secure or hardened for rendering workflows. It is suggested that the administrator creates a secure golden image and imports the VM image into Azure using the process described here: https://docs.microsoft.com/en-us/azure/virtual-machines/linux/tutorial-custom-images.

Azure Storage

The section pertains to all of the storage services that Azure offers: blobs, tables, queues, and files. Storage Service encryption (SSE) is turned on by default for all storage services. Data is encrypted before being written to storage and decrypted after the data is read. Microsoft manages the keys by default but a user can provide their own keys as an option. The Azure Key Vault can be used to manage the keys or the Azure Key Vault APIs can be used by applications to support user supplied keys. This is documented at: https://docs.microsoft.com/en-us/azure/storage/common/storage-service-encryption-customer-managed-keys?toc=%2fazure%2fstorage%2fblobs%2ftoc.json.

SECURITY CONTROL: Use Shared Access Signatures to access storage account resources

Azure storage account is a logical container used to store and access Azure Storage data objects, each storage account has two (2) Azure generated 512-bit storage access keys, which are used for authentication when the storage account is accessed. Storage account keys are similar to the root password in that users with the key have unfettered access to all the storage account's services. To authenticate access to an Azure storage account from a client application, an account access key is required. But most client applications should not require access to the entire storage, which includes all storage services including Tables, Queues, Files, Blobs and Azure virtual machine disks.

RECOMMENDATION: USE SHARED ACCESS SIGNATURES (SAS)
SAS provides fine-granular access to services within a storage account. The goal is to avoid distributing the storage account key to other users or applications, hardcoding it, or saving it anywhere in plaintext that is accessible to others. Asset management applications should only require access to a subset of services within a storage account, thus access via SAS should be sufficient. Additionally, we recommend a policy be put in place to explicitly define controlled SAS expiration times, tokens and source IP address ranges.

This recommendation supports cryptographic and least privilege principles.

Documentation: Azure SAS has multiple of use-cases and deployment models. Our recommendation focuses on the use of Shared Access Signature to control access to services within a storage account. Azure SAS is defined here: https://docs.microsoft.com/en-us/azure/storage/storage-dotnet-shared-access-signature-part-1.

SECURITY CONTROL: Periodically update access keys

To authenticate to an Azure storage account from a client application, an account access key is required. Regenerating storage access keys can affect associated Azure services (e.g. Batch) that are dependent on the storage account.

RECOMMENDATION: REGENERATE STORAGE ACCESS KEY EVERY 60 DAYS
User must regenerate storage account access keys periodically (e.g. every 60 days). All storage account client services that use the access keys to access the storage account must be updated to use the regenerated key.

This recommendation supports cryptographic principles.

Documentation: Azure compute virtual machines and Batch compute processes rely on storage accounts. Each storage account has a set of access keys. Storage account keys are 512-bit strings created by Azure that, along with the storage account name, can be used to access the data objects stored in storage. In the context of graphic rendering, these storage account keys should be rotated after a pre-defined period (e.g. 30 days) or after completion of production.

Azure storage security processes are defined here: https://docs.microsoft.com/en-us/azure/storage/storage-securityguide.

Azure Redis Cache

Azure Redis Cache is based on the open source (BSD) Redis cache. It has its own secure setup items which are outlined below.

SECURITY CONTROL: Disable non-SSL connections

Redis is designed to be accessed by trusted clients inside trusted environments and natively does not support encryption for data in-transit. The Azure cloud-based Redis database cache service provides access via SSL proxy for the additional layer of protection while users can configure the service to use the optional non-SSL based connection.

RECOMMENDATION: DISABLE THE NON-SSL PORT ON THE REDIS CACHE SERVER
Disable the non-SSL port to the Redis cache server to force users to use the SSL based connection only.

This recommendation supports cryptographic principles.

Documentation: Azure Redis security guide is located here: https://docs.microsoft.com/en-us/azure/rediscache/cache-how-to-premium-vnet.

SECURITY CONTROL: Disable non-SSL connections

Redis is designed to be accessed by trusted clients inside trusted environments and natively does not support encryption for data in-transit. The Azure cloud-based Redis database cache service provides access via SSL proxy for the additional layer of protection while users can configure the service to use the optional non-SSL based connection.

RECOMMENDATION: DISABLE THE NON-SSL PORT ON THE REDIS CACHE SERVER
Disable the non-SSL port to the Redis cache server to force users to use the SSL based connection only.

This recommendation supports cryptographic principles.

Documentation: Azure Redis security guide is located here: https://docs.microsoft.com/en-us/azure/rediscache/cache-how-to-premium-vnet.

SECURITY CONTROL: Monitor Redis cache performance

Redis is susceptible to attacks triggered by carefully selected inputs from external clients.

REDIS VULNERABILITY: DOS ATTACK

An attacker could supply, via a web form, a set of strings that is known to hash to the same bucket into a hash table in order to turn the O(1) expected time (the average time) to the O(N) worst case, consuming more CPU than expected, and ultimately causing a Denial of Service. Both use-cases are strange; an advanced attack or a poorly developed application can trigger these attacks.

RECOMMENDATION: MONITOR REDIS CACHE INSTANCES

Monitor the Azure Redis Cache instances using the Azure Monitor service. Azure Monitor can provide cache metrics, e.g., cache hits, misses, number of connected clients, used memory, CPU, and cache reads/writes in megabytes per second (MB/s). In addition, it can set alerts when certain conditions are met.

This recommendation supports the audit principle.

Documentation: Azure Redis monitoring and exception rules are discussed here respectively: Monitor (https://docs.microsoft.com/en-us/azure/redis-cache/cache-how-to-monitor) and Exception Rules (https://docs.microsoft.com/en-us/azure/redis-cache/cache-how-to-monitor#operations-and-alerts).

Azure Event Hub

The Event Hub is designed to the internet facing "front door" for an event pipeline. The Event Hub ingest is composed of an event “ingestor” service that exists between event publishers and event consumers to decouple the production of an event stream from the consumption of those events. Event Hubs provides message stream handling capability but has characteristics that are different from traditional enterprise messaging. Figure 2 depicts this architecture.

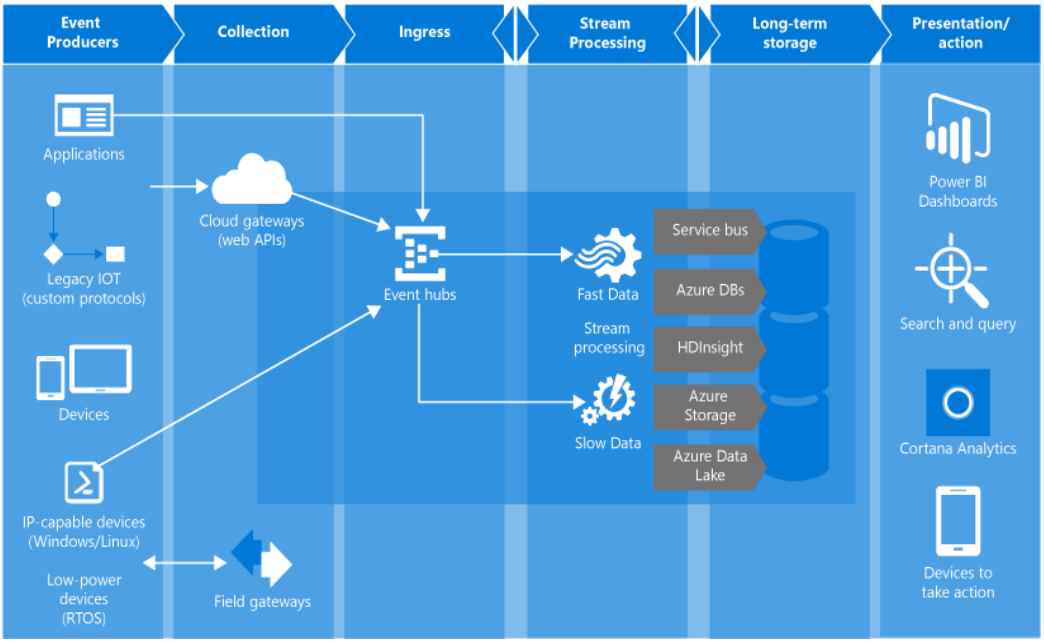

Figure 2 Event Hub implementation

SECURITY CONTROL: Prohibit use of Event Hub tokens on devices

The Event Hubs security model is based on a combination of Shared Access Signature (SAS) tokens and event publishers. Each Event Hub client is assigned a unique token, which is uploaded to the client. Clients claim access to Event Hub resources by presenting a SAS token. The tokens are produced such that each unique token grants access to a different unique publisher.

EVENT HUB CAVEAT: TOKENS ON DEVICES

A client that possesses a token can only send to one publisher. If multiple clients share the same token, then each of them shares a publisher. It is possible to equip devices with tokens that grant direct access to an event hub. Any device that holds this token can send messages directly into that event hub. Such a device will not be subject to throttling. Furthermore, the device cannot be blacklisted from sending to that event hub. All tokens are signed with a SAS key.

RECOMMENDATION: DO NOT FURNISH TOKENS DIRECTLY INTO CLIENT EVENT HUBS

Use the SAS method for token management instead of furnishing tokens directly to client even hubs.

This recommendation supports the separation principle and provides for the removal of an attack surface.

Documentation: Azure Event Hub security mode is described here: https://docs.microsoft.com/en-us/azure/eventhubs/event-hubs-authentication-and-security-model-overview.

SECURITY CONTROL: Use separate keys for Event Hub access

Each Event Hub namespace has a specific 256-bit SAS key called "RootManageSharedAccessKey". This key grants send, listen, and manage rights to the namespace. Single or separate keys can be used to control varying levels of access.

RECOMMENDATION: USE UNIQUE KEYS FOR EACH SPECIFIC EVENT HUB
Create a separate key that grants send permissions to a specific event hub.

For example, generate a key that grants only the "Send" permission and is different from a key that has listen and manage rights.

This recommendation supports separation and cryptographic principles.

Documentation: Azure Event Hub security is described here: https://docs.microsoft.com/en-us/azure/eventhubs/event-hubs-authentication-and-security-model-overview.

SECURITY CONTROL: Create partitions for consumer groups

In a multi-tenant application where Azure Event Hub is shared among different tenants, tenants can possibly send messages on different partitions. In such scenarios, a user might want to authenticate a tenant on a partition basis. Unfortunately, more than one key can send messages on a partition. The partition key gets hashed and then the hash space is divided amongst the space of partitions.

RECOMMENDATION: USE MULTIPLE EVENT HUBS
Use multiple Event Hubs to provide separation for each application or customer's data in different partitions.

This recommendation supports separation and cryptographic principles.

Documentation: Azure Event Hub event publishers and partitions are discussed here: https://docs.microsoft.com/enus/azure/event-hubs/event-hubs-features.

Azure Scheduler

Azure Scheduler allows developers to specify different one-time and recurring schedules for a job. Scheduler creates, maintains, and invokes scheduled work by calling job action services. Some of these services include other Azure services, Salesforce.com, Facebook, and secure custom websites.

SECURITY CONTROL: Perform validation of scheduled actions

As part of defense-in-depth scheduled jobs should be validated.

RECOMMENDATION: VALIDATE EXTERNAL JOBS
Validate external job services prior to setting up a scheduled service and using trusted services.

This recommendation is a defense-in-depth measure.

Documentation: Azure Scheduler service is discussed here in detail: https://docs.microsoft.com/enus/azure/scheduler/scheduler-intro.

Azure SQL Database

Microsoft has an extensive set of articles which discuss various aspects of the SQL database service. These articles are located at https://docs.microsoft.com/en-us/azure/sql-database/sql-database-security-overview. These articles discuss creation of server-level and database firewall rules, use of Azure SQL Database transparent data encryption (TDE) to automatically encrypt data at rest, auditing, data masking, backups, low-level security, and SQL database threat detection.

SECURITY CONTROL: Use separate Azure SQL database instances for production and clients

An Azure subscription can be used to create an unlimited number of database instances. Each Azure database should be assigned to a high-level business unit such as production, client, or an asset management workflow project.

RECOMMENDATION: INSTANTIATE DATABASES IN SEPARATE AZURE INSTANCES
Instantiate databases in a separate Azure subscription for each rendering production or client to limit access. Logically grouping databases through subscriptions will provide administrative anatomy and individualized resource monitoring.

This recommendation supports the separation principle.

Documentation: Azure SQL database security model is described here: https://docs.microsoft.com/en-us/azure/sqldatabase/sql-database-security-overview.

Azure Content Delivery Network

The Azure Content Delivery Network (CDN) is often used to deliver web pages but can be leveraged in an asset management system to serve media content to applications on a variety of device types.

SECURITY CONTROL: Use HTTPS protocol and port settings for CDN

When instantiating a new CDN service, the user must select a protocol and origin port, and these values specify the protocols and ports used to access resources at the origin. The Azure service requires the user to select at least one protocol (HTTP or HTTPS).

RECOMMENDATION: USE THE HTTPS PROTOCOL FOR CDN
Use the HTTPS protocol whenever possible. Using HTTPS requires a user to use an SSL certificate provided by the CDN, and use a CDN-provided domain to access HTTPS content.

This recommendation supports the separation principle.

Documentation: Azure CDN creation is described here: https://docs.microsoft.com/en-us/azure/cdn/cdn-create-newendpoint.

SECURITY CONTROL: Use token authentication to protect CDN content

Content requests should be authenticated by CDN edge POPs before delivering the asset. Lack of authentication can lead to Azure CDN serving assets to unauthorized clients. For example, links to assets can be shared on different websites and with different users.

RECOMMENDATION: USE TOKEN AUTHENTICATION
Use token authentication to prevent the Azure CDN from serving assets to unauthorized clients. Token authentication verifies requests generated by a trusted site through a token value containing encoded information about the requester.

This recommendation supports authentication.

Documentation: Azure CDN token authentication is discussed here: https://docs.microsoft.com/en-us/azure/cdn/cdntoken-auth.

SECURITY CONTROL: Define custom HTTP behavior for CDN

In media production workflows, it is imperative to protect digital assets against theft, misuse, or piracy. When servicing content through CDN, it is important to setup specific policies and rules for blocking the delivery of certain types of content, defining a caching policy, and modifying HTTP headers.

RECOMMENDATION: DEFINE EXPLICIT CDN RULES
Define explicit CDN rules to limit asset delivery to a select set of users and requests. A robust caching policy must be defined that limits the content's "time to live", for example, defining an aggressive cache purge policy for sensitive or protected content.

This recommendation supports separation and authentication.

Documentation: Overriding of HTTP behavior of CDN rules engine is discussed here: https://docs.microsoft.com/enus/azure/cdn/cdn-rules-engine.

Azure Media Services

Azure Media Services was created for video workflows. Microsoft outlines this service here: https://azure.microsoft.com/en-us/services/media-services/.

SECURITY CONTROL: Use separate Azure storage accounts for media service accounts

Storage accounts must be located in the same geographic region as the Media Services account. When a user creates a Media Services account, they either use an existing storage account in the same region, or create a new storage account in the same region.

RECOMMENDATION: USE SEPARATE STORAGE ACCOUNTS FOR MEDIA ACCOUNTS
Use separate storage accounts for media accounts since logically media accounts associated with the Media service are separate entities. Assets in the storage account associated with the Media service will likely to be shared with an audience or processed for a specific purpose. The content of the storage account associated with the media should only share a limited set of data which the user has explicitly marked for processing or sharing. The storage should be protected with guidance provided earlier in this document.

This recommendation supports separation.

Documentation: Azure Media account setup is described here: https://docs.microsoft.com/en-us/azure/mediaservices/media-services-portal-create-account.

SECURITY CONTROL: Encrypt assets

Media production workflows rely heavily on ad-hoc and near real-time asset sharing and collaboration. Azure Media service video delivery can be used to deliver video assets within a content production team across the globe. The video assets should be protected when shared in this manner.

RECOMMENDATION: USE THE KEY DELIVERY SERVICE WITH AES
Use the Key Delivery service with the Advanced Encryption Standard (AES) using 128-bit encryption keys for delivery of Http-Live-Streaming (HLS) and Smooth Streams video assets.

The Azure Key Delivery service can be used to deliver encryption keys to authorized users. A user can encrypt an asset and associate an encryption key with the asset then configure authorization policies for the key. When a stream is requested by a player, Media Services uses the specified key to dynamically encrypt content using AES encryption.

To decrypt the stream, the player will request the key from the key delivery service. To decide whether or not the user is authorized to get the key, the service evaluates the authorization policies that were specified for the key. This method allows a content publisher to control the content delivery by revoking the key.

This recommendation supports cryptography.

Documentation: Azure Media encryption is defined here: https://docs.microsoft.com/en-us/azure/mediaservices/media-services-protect-with-aes128.

SECURITY CONTROL: Configure Live Media archiving policy

Media Services offers live streaming of events directly from a camera device. While streaming the event, the service can be configured to archive content in the storage account as it is encoded and streamed live.

RECOMMENDATION: DEFINE CONTENT SPECIFIC ARCHIVE POLICY
Definie a content specific archiving policy for streamed content. In media production environments, each asset has its own lifecycle which should be mirrored in its archive policy.

This recommendation supports backup and restore.

Documentation: Azure Media live streaming is defined here: https://docs.microsoft.com/en-us/azure/mediaservices/media-services-manage-channels-overview.

We developed a deployable reference architecture that incorporates controls from this guide along with Azure best practices for extending an on-premises digital data asset management setup to Azure. This architecture is an extension of the Microsoft developed generic "Implementing a secure hybrid network" reference architecture.

Azure Cloud Asset Management – Example Architecture

The reference architecture includes provisions to extend an on-premises network and Active Directory (AD) environment to the Azure cloud. Furthermore, the architecture includes a perimeter network between an on-premises network and an Azure virtual network for protection of network appliances. All outgoing traffic from the VNet is force-tunneled to the external network through the on-premises network, so that it can be audited. The architecture requires a connection to the on-premises datacenter, which can be either a VPN gateway or an ExpressRoute connection.

This architecture depicts both virtual machine and batch processing rendering farm, but in practice, either setup is sufficient. Lastly, the architecture includes a management subnet with a bastion server to monitor the setup or a jump box to traverse to other VMs. The following figure depicts the important components of an on-premises and cloud-based VFX rendering architecture:

Placeholder for reference implementation diagram.

KEY ELEMENTS OF THE ARCHITECTURE

- On-premises network: a private local-area network implemented in an organization.
- Azure virtual network (VNet): The VNet hosts the application and other resources running in Azure.
- **Gateway:** The gateway provides connectivity between the routers in the on-premises network and the VPC. User-defined routes handle routing for on-premises traffic that passes to Azure.
- **Network appliance:** The network appliances are computing devices which perform tasks such as allowing or denying access such as a firewall, optimizing wide area network (WAN) operations (including network compression), custom routing, or other types of network functionality. These appliances are only accessible from a gateway subnet. They are shown in the diagram sitting between the private DMZ in and out segments.
- **Management subnet:** This subnet contains VMs that implements management and monitoring capabilities for the components running in the VNet.
- **Management bastion host:** A VM on the network that administrators can use to connect to the other VMs. The bastion host has an NSG that allows remote SSH traffic only from public IP addresses on a white-list.
- **User defined routes:** These rules define the flow of IP traffic within VPC subnets

- **Active Directory servers:** These servers are domain controllers implementing directory services (AD DS) running as VMs in the cloud. These servers can provide authentication of components running in your Azure virtual network.
- **Active Directory subnet**: The AD DS servers are hosted in a separate subnet. Network security group (NSG) rules protect the AD DS servers and provide a firewall against traffic from unexpected sources.

Deployment Scripts

The following set of PowerShell CLI scripts can be used to deploy hybrid VFX rendering reference architecture. This architecture deployment is based on reference scripts provided by Microsoft on their Microsoft on Microsoft patterns & practices GitHub portal. Microsoft's Microsoft patterns & practices GitHub portal is located here: https://github.com/mspnp/reference-architectures.

Prerequisites. Active Azure account and subscription. The latest version of the Azure CLI is required for the script that deploys the reference architecture.

Notes: the provided scripts are for reference purposes only. The scripts setup a simulated on-premises network in Azure for testing.

To deploy the reference architecture, follow these steps:

1. Copy and extract the attached CLI script folder. Make sure the PowerShell script execution policy is set to allow execution of unsigned scripts.
2. Open the Azure PowerShell CLI and navigate to the local folder.
3. Download blender-2.78c-windows64.zip and copy it into batchApp directory.
4. Run the following command:

   .\Deploy-AzueVFXReferenceArchitecture.ps1

   Wait for the deployment to complete, it can take several hours.

azure-rendering-deployment-v2_05222019.zip

ABOUT ISE

ISE is an independent security firm headquartered in Baltimore, Maryland. We are dedicated to providing clients proven scientific strategies for defense. On every assessment our team of analysts and developers use adversary-centric approaches to protect digital assets, harden existing technologies, secure infrastructures, and work with development teams to improve our clients' overall security.

Assessing the client through the eyes of potential attackers allows us to understand possible threats and how to protect against those attacks. Our security analysts are experienced in information technology and product development which allows them to understand the security problems the client faces at every level. In addition, we conduct independent security research which allows us to stay at the forefront of the ever changing world of information security.

Attacks on information systems cannot be stopped, however with robust security services provided by an experienced team, the effects of these attacks can often be mitigated or even prevented. We appreciate the confidence placed in us as a trusted security advisor. Please don't hesitate to get in touch for additional assistance with your security needs.

Independent Security Evaluators

4901 Springarden Drive
Suite 200
Baltimore, MD 21209
(443) 270-2296

contact@securityevaluators.com