

Implementing and Managing Microsoft Defender for Cloud Apps

Microsoft Defender for Cloud Apps (MDA) is the **cloud access security broker (CASB)** in Microsoft 365 Defender. Simply put, it sits between your users and their access to websites, cloud apps, **software as a service (SaaS)**, and **infrastructure as a service (IaaS)**. As apps, services, and infrastructure have moved to the cloud, the level of visibility and control we have over them has, in some cases, been reduced. This is where MDA steps in, enhancing that visibility and giving us the means to also control what previously couldn't be in our SaaS and IaaS.

In [Chapter 2](#), you learned about the many capabilities MDA has at a high level and, in this chapter, we'll explore things in more detail. You'll leave this chapter knowing how to do the following:

- Customize MDA settings for your environment
- Manage shadow IT discovery
- Connect apps for investigation and governance, then implement session policies for reverse proxy controls
- Use the app governance add-on for OAuth app defense

While most of the operations we'll discuss can be performed in the Microsoft 365 Defender portal at security.microsoft.com, some are found in a separate MDA portal at portal.cloudappsecurity.com. This owes to the fact that MDA was previously entirely managed in its own portal, but Microsoft has started to consolidate things in the Microsoft 365 Defender portal. For any tutorials in this chapter, we will make it clear which portal you should be in, but keep in mind things may change over

time because of the centralization efforts. The main interface for MDA in the Microsoft 365 Defender portal is the **Cloud apps** section of the left navigation pane, as depicted in *Figure 16.1*:

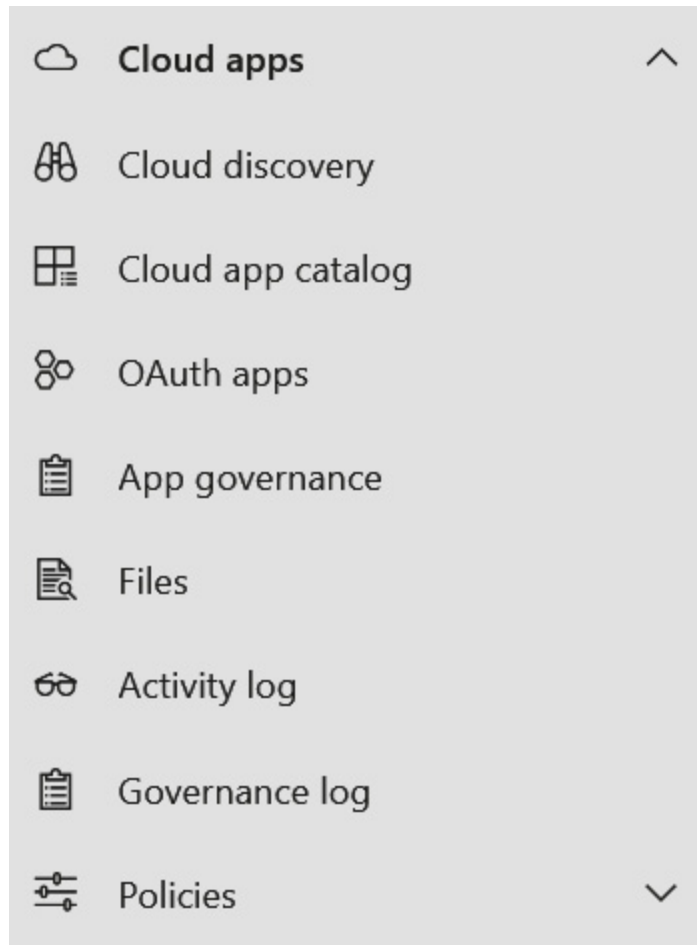


Figure 16.1 – MDA options in Microsoft 365 Defender

We'll start off our dive into MDA by looking at some fundamental settings you should be customizing.

Exploring MDA settings

As with most security services, in MDA, we must tweak several settings to get things tuned to our specific environment. General settings for MDA can be found by going to **Microsoft 365 Defender portal | Settings | Cloud apps**:

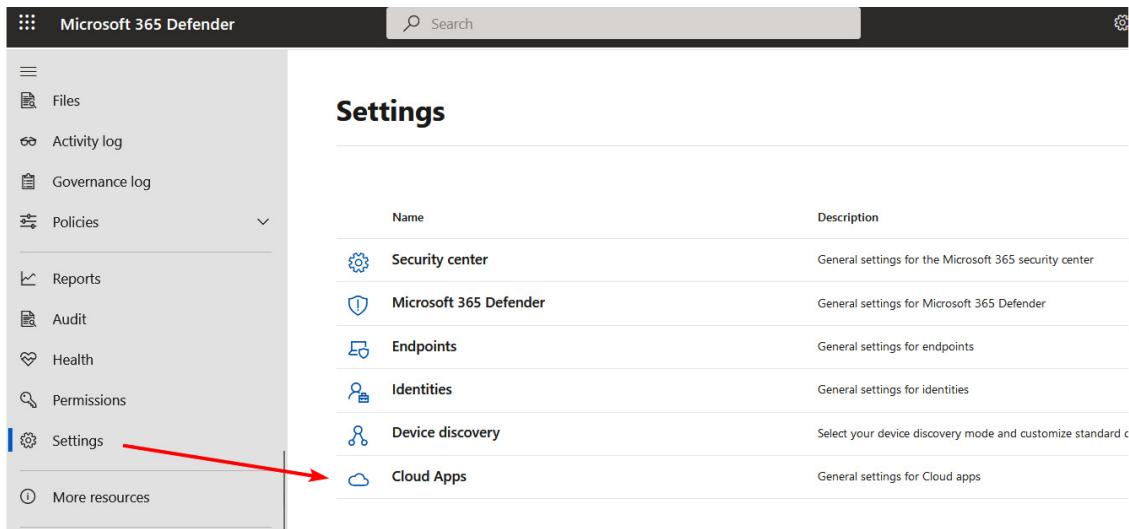


Figure 16.2 – MDA settings in the Microsoft 365 Defender portal

For now, we're going to look at the settings in the **System** and **Information Protection** categories. We'll look at the other categories throughout this chapter.

Customizing system settings

There are several changes you should make to improve defaults in the **Cloud Apps** settings page you just navigated to. Let's check them out!

Starting in **Organization details**, you can customize the environment to your organization by changing the organization's display name and environment name, as well as uploading a logo (limited to a rather small 150 x 50 pixels). You'll also see a list of **Managed domains**, which MDA uses to classify someone as internal or external in, for example, file-sharing policies. By default, it pulls this information from your Microsoft 365 domains, which is generally sufficient, but may require amending for your environment.

The customization continues on the **Mail settings** page, in which you can change the email address, display name, and reply-to address of user (not admin) email alerts. By default, these are **no-reply@cloudappsecurity.com**, **Microsoft Defender for Cloud Apps**, and **no-reply@cloudappsecurity.com**, respectively. The interesting thing

with customizing this is that you will end up using MailChimp, rather than Microsoft, to distribute the email, so you must consider the email authentication implications of this, such as DMARC and SPF. The final customization you can make is changing the email design by uploading an HTML template:

Organization details

Mail settings

Scoped deployment and privacy

IP address ranges

User groups

API tokens

SIEM agents

Playbooks

Cloud Discovery

Score metrics

Snapshot reports

Continuous reports

Email sender identity
For email notifications of alerts that are sent by Microsoft Defender for Cloud Apps to users and admins

☐ Default settings
Display name: Microsoft Defender for Cloud Apps | Email address: no-reply@cloudappsecurity.com | Reply-to address: no-reply@cloudappsecurity.com

☒ Custom settings

"From" display name

"From" email address

Reply-to email address

☒ By selecting custom settings, I acknowledge that mail notifications will be serviced by MailChimp and not by Microsoft, and that I have reviewed and agreed to MailChimp's [Terms of Service](#) and have reviewed their [Privacy Statements](#).

Figure 16.3 – Using MailChimp to send MDA alerts from a custom address

The **User groups** settings page highlights an intricacy of MDA – that, out of the box, the groups you would often use to scope policy must be actively imported from a source such as Azure AD; otherwise, they won't appear. When you connect an app to MDA (which will be discussed in the *Managing cloud apps with policies* section), it can query that app's groups for scoping. For example, connecting Office 365 lets you import Azure AD groups. It's a simple process:

1. On the **User groups** settings page, click + **Import user group**.
2. Choose the source-connected app.
3. Choose the user group from that app.
4. Optionally, choose to receive an email notification:

Import user group

The imported user group will be updated automatically by syncing with its source user group.

Select an app to import the group from:

Office 365 (Azure AD) ▼

Select user group to import:

Lab Users ▼

Note: The import process may take a while to complete based on the group size.

☒ Notify me by email when the import is complete

Import

Figure 16.4 – Importing an MDA user group

The import is not immediate, and you can be notified by email when it's complete. There's a 500 imported group limit and after the initial import, membership continues to update automatically but is always limited to active users only.

Some groups are available by default, based on dynamic rules (that you can't control). For example, **External** is based on a domain name, and there are administrator groups for Dropbox, Office 365, Google Workspace, and Box.

Now that you have imported some user groups, we can look at the **Scoped deployment and privacy** settings. Scoped deployment options are useful when you want to limit MDA's monitoring capabilities – for example, do not monitor users based on their geography (legal reasons) or only monitor users in a specific group (during initial deployments).

You'll notice tabs for **Include**, **Exclude**, and **Activity privacy**. By default, all users are included in the scope. When you use included groups, it automatically excludes all other groups. If you then choose to exclude groups, these take precedence over included groups. For example, if you

include a group called **IT Team** but exclude a group called **German Employees**, any members of both groups are excluded.

Creating inclusions and exclusions is simple:

1. Choose the **Include** or **Exclude** tab.
2. Click + **Add rule**.
3. Give the rule a name and select the user group(s) it applies to.
4. Choose to **Apply to all apps**, **Select specific apps...**, or **Select specific app instances...**:

Create new include rule

Type rule name *

External only

Select user groups ⓘ

User groups must first be imported in the [User groups page](#)

External users

- ☒ Apply to all apps
- ☐ Select specific apps...
- ☐ Select specific app instances...

Create

Figure 16.5 – Creating an inclusion or exclusion rule in MDA

MANAGING EXCLUSION GROUPS

A recurring message in this book is to manage your excluded groups carefully. There will be important reasons for exclusions, such as legal compliance, but they do introduce risk. Consider ongoing attestation options such as Azure AD Identity Governance to make sure their membership is limited.

The **Activity privacy** tab is used to limit who can view user activity. The same telemetry is obtained so that you don't need to create exclusions (and therefore blind spots), but it addresses some privacy concerns. Without adequate permission, you cannot view the activity. There is a significant *gotcha* with activity privacy: when enabled, activities that fall under its scope don't make their way to advanced hunting or other data connectors such as Sentinel.

To set up activity privacy, follow these steps:

1. Click on the **Activity privacy** tab.
2. Choose + **Add group** and select your group.
3. Head to **Microsoft 365 Defender** | **Permissions** | **Activity Privacy Permissions**:

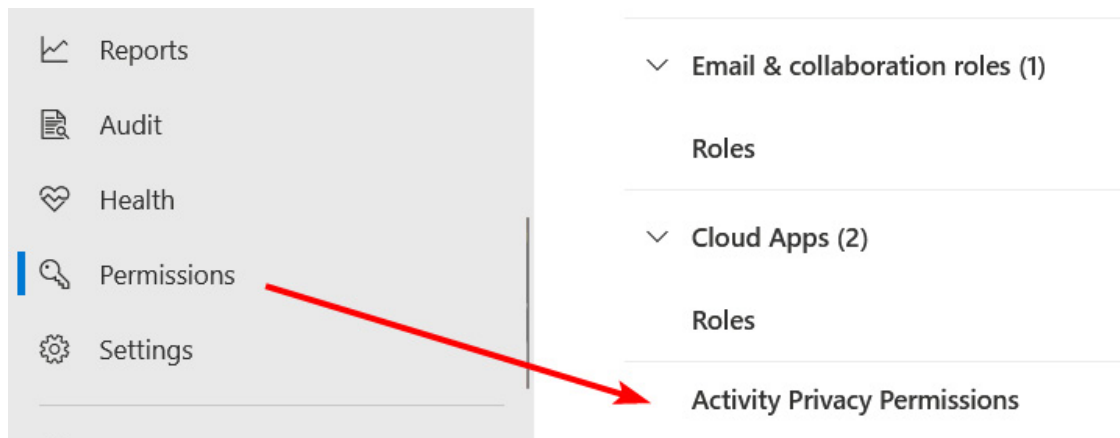


Figure 16.6 – Giving an MDA admin permission to view private activity

4. Choose + **Add user** to enter a named user with this permission.

Now, when an admin with privacy activity permissions browses the MDA activity logs, they can amend the table filter settings to **Show private activities**.

The last section of system settings we'll look at is **IP address ranges**. IP address ranges are ways of identifying the source of activity. IPv4 and IPv6 are supported for IP address ranges, and there are many built into

MDA. These may be cloud providers such as Cloudflare or **Amazon Web Services (AWS)**, or risky IPs such as the **Threat Intelligence** range.

Each range will have one category but can have several tags. Both are useful for your investigations and building alerts. For example, creating a custom IP address range, you may choose the **Corporate** category and use two tags: **Datacenter ABC** and **Public web servers**. If there's a conflict between built-in and custom tags, the custom tag will win.

Adding known and managed IP ranges is recommended so that you can improve the service's ability to understand your environment. For example, when integrated with Azure AD Identity Protection, having your corporate and VPN IP ranges reduces false positive impossible travel events. It's a straightforward process:

1. Head to **Microsoft 365 Defender | Settings | Cloud apps | IP address ranges**.
2. Click + **Add IP address range**.
3. Populate the **Name**, **IP address ranges** (in CIDR format), **Category**, and (optional) **Tags** fields.
4. Optionally, you can also choose to override the automatic geo-IP location and ISP:

New IP address range

[Learn more](#)

Name *

IP address ranges *

Category *

Corporate v

Tags


Datacentre ABC x Public web servers

Override automatic data enrichment ⓘ

☐ Override registered ISP

☐ Override location

Create a new tag ""

 Only future events will be affected by the new or modified IP address range.

Create

Cancel

Figure 16.7 – Creating a new IP address range in MDA

By working your way through all these system settings, you lay the foundations for your MDA implementation. Next up, we're going to look at **Information Protection** settings.

Customizing Information Protection settings

The group of settings under **Information Protection** in MDA leverages its integration with **Microsoft Purview Information Protection** (also known as Azure Information Protection in its previous incarnation), as well as general file and data security. The settings you customize here will be used by policies you create later.

Let's start with the **Admin quarantine** page. In the *Managing cloud apps with policies* section of this chapter, you'll learn about **file policies**, which apply actions based on criteria you specify in an *if this, then that* type of logic.

If your governance action quarantines an MDA discovered file – or an administrator manually chooses it – the location you specify on this settings page determines where that quarantine is, with the original file replaced by a **tombstone text file**. This book recommends creating a dedicated SharePoint Online site to host the admin quarantine, with SharePoint permissions limited to the appropriate administrators. Once a site becomes available, it's only a matter of using the **Select folder location** drop-down menu and, optionally, providing a **User notification** that's displayed if a file policy triggers.

The **Files** settings page simply has one tick box: **Enable file monitoring**. When ticked, this allows MDA to use your connected app's APIs to retrieve a list of files and activities surrounding them. Why do this? It allows you to investigate files (for example, during incident response) and create proactive security policies – for example, prevent files from being

shared with unauthorized parties. You'll learn more about these policies in the *Managing cloud apps with policies* section of this chapter.

Now that file monitoring is enabled, we can move on to the **Microsoft Information Protection** settings page. Here, things are simple, and there are only two options:

- **Automatically scan new files for Microsoft Information Protection sensitivity labels and content inspection warnings**

This setting, often abbreviated to *automatic scan*, enables the general integration of sensitivity labels with MDA. This is a prerequisite to any policy that applies labels, and any discovery that uses MDA for labeled files. After turning this on, MDA syncs all your tenant's sensitivity labels and continues to do so (approximately) hourly.

AUTOMATIC SCAN

Scanning for labeled files does not apply retrospectively; it only applies to files modified after this option has been enabled. To bypass this limitation, you can create a new file policy that will include existing files in scope. You'll learn how to create policies later in this chapter. Historically, creating such a policy has even fixed problems such as new labels not syncing.

- **Only scan files for Microsoft Information Protection sensitivity labels and content inspection warnings from this tenant**

When enabled, you can change MDA's default mode of scanning labels from any tenant to only labels homed in your tenant.

The last settings page we're going to cover deals with **data loss prevention (DLP)** and is called **External DLP**. On this page, you can set up connectors with third-party classification systems. This means that if you use compatible third-party DLP systems, you can keep their logic for determining if a file should be subject to DLP controls. Fully configuring this is beyond the scope of this book, as it's contextual to your third-party DLP,

but you should understand that it's a potential MDA feature for your organization.

External DLP uses the **Internet Content Adaption Protocol (ICAP)** to transmit files to the third-party service, optionally keeping those files secure with a TLS tunnel. The files that are transmitted are determined by file policies, which you'll learn how to configure in the *Managing cloud apps with policies* section later in this chapter. After the external service DLP decides, it reports back to MDA, which enforces actions that you've specified in the file policy.

We'll refer to additional controls in **Microsoft 365 Defender | Settings | Cloud apps** throughout this chapter as we explore other features, but by starting with the system and information protection settings you've learned about, you lay the foundations of MDA use. In the next section, we'll look at the next part of those foundations: **cloud discovery**.

Discovering and managing shadow IT

Reliable inventory is foundational to cybersecurity. We need to know what devices, identities, and infrastructure we have to adequately secure them and reduce the attack surface. With this in mind, ask yourself: do you know your cloud app inventory? **Cloud apps** is the umbrella term for web apps and SaaS. While you're probably aware your business uses services such as Office 365 and some others, you probably won't know it all, and this problem will grow as your organization grows.

We call this type of unknown, unmanaged IT and cloud app usage **shadow IT**. It poses a risk because the following generally haven't occurred:

- Legal assessments, such as data residency and ownership policy

- Security assessments, such as support for **single sign-on (SSO)** and admin audit trails
- Internal compliance assessments, such as disaster recovery plans

Most shadow IT usage is simply users trying to get their job done as quickly as possible, but it's important to balance this with cybersecurity requirements. This is where MDA's **Cloud Discovery** feature comes into play.

Cloud Discovery and the Cloud App Catalog

The objective of Cloud Discovery is to analyze the use of cloud apps in your environment, shining a light on shadow IT so that you can manage it.

Historically, firewalls log all inbound and outbound traffic but perhaps don't do a good job of illustrating it, and the move to hybrid work without centralized firewalls has made this more challenging. MDA's approach to Cloud Discovery can still leverage firewall appliance logs, but it also integrates with MDE to get that information straight from the client. If you don't use MDE but use a third-party **secure web gateway (SWG)**, this is also possible, with support for Corrata, iboss, Menlo, and Zscaler. Logs can be ingested ad hoc using **snapshot reports** or consistently with **continuous reports**.

To convert those logs into something useful, they can be run against the **Cloud App Catalog**. You can think of the Cloud App Catalog as the backbone of Cloud Discovery for real-world context. Without it, your data is just data. With it, your data becomes *information*.

The Cloud App Catalog is an ever-growing library of cloud apps that Microsoft manages information about. At the time of writing, it contains over 31,000 cloud apps. The type of information you'll find relates to the risks of shadow IT mentioned earlier in this section: legal, security, and compliance assessments. These are summarized in a numerical **risk**

score from 0 to 10, with 0 implying the most risk and 10 implying the least risk. You can expect your Cloud Discovery information to update against the Cloud App Catalog four times per day. An example of what to expect is demonstrated in the following figure:

App	Risk score	Tags	Traffic	Upload	Transactio...	Users	IP addresses	Devices	Last seen ...	Actions
Microsoft Outlook Webmail	10		3 MB	35 KB	22	1	15	1	21 Oct 2022	ⓘ ⚙ ⋮

Microsoft Outlook is a personal information manager from Microsoft, available as a part of the Microsoft Office suite

[Suggest an improvement](#) [Disclaimer](#) 10

GENERAL 10

Category: Webmail	Headquarters: United States	Data center: United States
Hosting company: Microsoft Corporation	Founded: 1975	Holding: Public
Domain: *.outlook.live.com, *.mail.live.com, ...	Terms of service: microsoft.com/en-us/legal/terms-o...	Domain registration: Feb 5, 1991
Consumer popularity: 10	Privacy policy: privacy.microsoft.com/en-us/privacyst...	Logon URL: login.live.com/login.srf
Vendor: Microsoft	Data types: Documents, Media files, Database file...	Disaster Recovery Plan

SECURITY 10

Latest breach: —	Data-at-rest encryption method: AES	Multi-factor authentication
IP address restriction	User audit trail	Admin audit trail
Data audit trail	User can upload data	Data classification
Remember password	User-roles support	File sharing
Valid certificate name	Trusted certificate	Encryption protocol: TLS 1.2

Figure 16.8 – Cloud Discovery information for Microsoft Outlook

Now that you understand what Cloud Discovery and the Cloud App Catalog are, let's take a look at how to set them up.

Setting up Cloud Discovery

First, we'll head to security.microsoft.com, navigate to **Settings | Cloud Apps | Cloud Discovery**, and focus on some important settings:

- The first group of settings is **Score metrics**. These settings allow you to customize how the Cloud App Catalog generates its 0-10 risk score. Attributes such as **Supports SAML**, **HIPPA**, **ISO 27001**, and **Data retention policy** can be adjusted so that you can change their importance to ignored, low, medium (default), high, or very high. You can also change an entire category's importance, such as **Security**, **Compliance**, or **Legal**. Customizing these settings can be useful if you are in highly regulated environments and want to focus on particular qualities of cloud apps.

- If you need to exclude users, groups, devices, or IP addresses from Cloud Discovery (for example, due to privacy concerns), you can do so using the **Exclude entities** options. This does not apply retrospectively.
- **User enrichment** options let you correlate traffic logs with users based on their Azure AD **user principal name (UPN)** if that data source (for example, third-party firewall logs) can supply the username and you've integrated MDA with Office 365. This will be covered in the *Connecting apps to MDA* section later in this chapter.
- Back to our privacy options, the **Anonymization** settings can be set to anonymize usernames and device names by encrypting them with AES-128 and a tenant key. This may be an important setting to prevent MDA or Microsoft 365 Defender administrators from seeing user web activity that is sensitive. After enabling anonymization, reports that use MDA will report users and devices as a random string. You can return to the **Anonymization** settings page to resolve these if they're required for an investigation or perform resolution via the **Options** button in Cloud Discovery. Both require a justification and are audited:

Anonymization



Data anonymization protects user privacy by encrypting private information.

Anonymize usernames by default

☐ Anonymize private information by default in new reports and data sources

Anonymize device names by default

☐ Anonymize device information by default in 'Win10 Endpoint Users' report



[Learn more](#)

Save...

Resolve anonymization

This action will be audited in the [Governance log](#).

1 Enter justification (required)

Enter justification...

2 Enter name to resolve

User From anonymized Type name...

Figure 16.9 – Anonymization options for Cloud Discovery

Now that we've looked at the settings for managing Cloud Discovery, it's time to get on with ingesting our logs. What better place to start than straight from the endpoint with MDE?

Microsoft Defender for Endpoint

In *Chapter 7*, you learned how to enable network protection for Windows. Network protection, real-time protection, and cloud-delivered protection are prerequisites for integration with MDA. You also configured a service-side prerequisite in *[Chapter 3](#), The Fundamentals of Microsoft Defender for Endpoint*, when you enabled MDA integration with MDE by going to [security.microsoft.com](#) and clicking on **Settings | Endpoints | Advanced features**.

So, you're almost there! There is just one more setting to enable. Head to [security.microsoft.com](#), navigate to **Settings | Cloud Apps | Cloud**

Discovery | Microsoft Defender for Endpoint, and click the checkbox next to **Enforce app access**. You can also customize the bypass duration and notification URLs when in **monitor mode**. You'll learn more about enforcing app access monitor mode in the upcoming section, *Managing cloud app access*.

Third-party automatic log uploads

MDA supports numerous firewall and proxy applications. Support includes vendors such as Check Point, Cisco, Forcepoint, Juniper, Palo Alto, SonicWall, WatchGuard, and the SWGs you learned about earlier. For a full list of Cloud Discovery-supported firewalls and proxies, you can refer to the official documentation at learn.microsoft.com/en-us/defender-cloud-apps/set-up-cloud-discovery#supported-firewalls-and-proxies.

Even if your platform isn't listed as supported, there is general support for CEF, LEEF, W3C, and CSV. With the latter, you can customize options such as headers, delimiters, timestamps, and other forms of formatting.

Automatic log uploads for any platform start with a **data source**. This is an object created in MDA that represents where logs are coming from. For example, *Win10 Endpoint Users* is the default source for MDE integration. Each data source specifies its **source appliance** (or generic format) and its **receiver type**, such as Syslog protocol, FTP, or FTPS.

A **log collector** is used to connect the data source to MDA. It's the log collector that gets the logs from the data source, then sends them to MDA as an intermediary. The log collector is containerized and runs in Docker, on infrastructure that you manage, such as Linux or Windows. Up to 10 data sources can share a log collector, with a maximum capacity of 50 GB per hour. The log collector initiates outbound traffic on port 443 to MDA and Azure blob storage, as well as requiring inbound traffic from the data source. To add Microsoft's data centers and other required addresses and ports to your log collector's allow list, refer to the official documentation:

learn.microsoft.com/en-us/defender-cloud-apps/network-requirements#log-collector.

SECURE WEB GATEWAYS – ENHANCED INTEGRATION

In the Cloud Discovery and the Cloud App Catalog section, we called out support for SWGs such as Corrata, iboss, Menlo, and Zscaler. These can send logs directly to MCA without a log collector, so refer to the vendor's documentation for configuration.

The architecture of ingesting logs into MDA with a log collector is depicted in the following figure:



Figure 16.10 – How data sources send logs to MDA

In the steps that follow, you'll learn how to connect a data source to a log collector on Linux for automatic uploads to MDA. For this guide, we've assumed Docker is already running. If not, you should refer to the official guidance for your platform at docs.docker.com/engine/install. Let's begin!

1. Head to security.microsoft.com and navigate to **Settings | Cloud Apps | Cloud Discovery | Automatic log upload**.
2. In the **Data sources** tab, click + **Add data source**. You'll repeat this and the next step for any sources you want to upload.
3. Give your data source a **Name** and choose the appliance (**Source**) and **Receiver type**. Optionally, anonymize private information. Use **View sample of expected log file** to confirm that your formats match what MDA expects:

Add data source

Name *

FW-01

Source *

PA Series Firewall

[View sample of expected log file](#), and compare it with yours

Receiver type *

Syslog - TCP

☐ Anonymize private information
Store and display only encrypted usernames.

Add Cancel

Figure 16.11 – Adding a data source

4. In the **Log collectors** tab, click + **Add log collector**.
5. Give your log collector a **Name**, the private IP or FQDN of the server running Docker, then choose your newly created **Data source(s)**.
6. When you click **Create**, the wizard will expand to give you a comment to take note of and, if needed, FTP credentials:

Create log collector

Name

LC-01

Host IP address or FQDN ⓘ

192.168.1.1

Data source(s)

FW-01 x

Create

Next steps:

1. Follow the [deployment guide](#) to install the log collector on your host
2. On the hosting machine, import the collector configuration using:

(echo [REDACTED]) | docker run --name LC-01 -p 601:601/tcp -p 21 ⓘ
3. Configure exports from data sources (in your network) to the log collector according to the following:

↓ Export ⓘ ⚙ Table settings ▾

Name	Source	Receiver type	Destination path/port
FW-01	PA Series Firewall	Syslog	TCP/601

FTP user: discovery FTP password: [REDACTED]

Close

Figure 16.12 – Creating a log collector

7. Now, you can head to your Docker server and use this information to create the log collector by pasting in the command, as shown in the preceding figure.
8. After executing the command, execute **sudo docker logs <your collector>**. You may be prompted for FTP credentials. An output of **Finished successfully!** confirms that your log collector is up and running. Alternatively, pursue any errors observed in the **Starting services** output by reviewing prerequisites such as network connectivity and Docker configuration.
9. With the log collector available, you need to set up your appliance to send logs to it. How this is conducted will vary, depending on the appliance with your logs (the data source), so review the vendor documentation, and refer to the **Destination path/port** area shown in *Figure 16.12*.

10. Heading back to security.microsoft.com, under **Settings | Cloud Apps | Cloud Discovery | Automatic log upload**, you will see your collector with a status of **Connected** after parsing has been completed:

Automatic log upload



Data sources Log collectors

Automatically sanitize, compress and transmit log data to the portal.
To use this feature a log collector machine needs to be deployed.

[Terms](#) | [Privacy statement](#)




<div><div></div><div>Table settings</div><div></div></div>					
Name	Status	Linked data sources	Last data rece...	Modified date	
	<div><div></div><div>Connected</div><div>Data is being successfully uploaded from all linke...</div></div>	1 data source	23 Oct 2022	8 Sep 2022	<div></div>

Figure 16.13 – A successfully connected log collector

With that, we've assessed two ways to ingest logs into MDA's Cloud Discovery: MDE and log collectors. There is also a third way, known as **snapshot reports**, which we'll explore next.

Snapshot reports

If you want to test MDA's Cloud Discovery capabilities before creating a log collector for automatic ingestion, you can use a snapshot report. This is a manual ad hoc approach that still lets you get some insight into cloud app usage. Let's check out how to create one:

1. Browse to security.microsoft.com and navigate to **Settings | Cloud Apps | Cloud Discovery | Snapshot reports**.
2. Click + **Create snapshot report**.
3. Click through the wizard until you get to the **Report Details** tab. Enter a **Name** and **Description**, and choose a **Source**. The source choice can be expanded with the **View log format** option to confirm your data matches MDA's expectations:

Create new Cloud Discovery snapshot report

[? Guide](#)

OVERVIEW — REPORT DETAILS — UPLOAD TRAFFIC LOGS — FINISH

Report Name *

Snapshot 1

Description

Source *

PA Series Firewall

① Verify your log format

Make sure your log files are in the expected format for your data source, otherwise they cannot be processed. To add a custom format, reconfigure your data source settings to match your format.

[View log format ^](#)

FTP (Supported in snapshots and automated upload)

Generate Time,Source address,Source User,Action,Application,URL,Destination Port,Bytes Re
2017/07/03 00:00:00,10.0.9.5,Spencer@contoso.com,allow,http,www.microsoft.net,80,121678,1!

Figure 16.14 – Creating a snapshot report

4. Proceed to the **Upload Traffic Logs** tab and upload your file.
5. Uploads are not available immediately, so you will be notified by email after processing has concluded.

You now know the three ways to ingest cloud app information into MDA. Now what? Well, it's time to start looking at that cloud app information and, better still, acting on it.

Reviewing and managing cloud app access

We can review Cloud Discovery summary information at a high level from the **Cloud Discovery dashboard** at security.microsoft.com after navigating to **Cloud Apps | Cloud Discovery | Dashboard**. This provides a visual depiction of app usage, such as traffic, categories, risk level, and users.

A more in-depth review can be conducted in the **Discovered apps** tab, which reports on the individual cloud apps, or the **IP addresses**, **Users**,

and **Devices** tabs, which do the same for their respective categorizations. For example, you could navigate to the list of users, click on a user, then see their discovered app usage.

THE WHOLE CATALOG

*Want to see all apps that are managed in the MDA catalog, regardless of traffic in your organization? Head to security.microsoft.com and go to **Cloud Apps | Cloud App Catalog**. This is useful for pre-emptive cloud app management!*

At the top right of all **Cloud Discovery** section tabs (1 in *Figure 16.15*), you'll find the option to change the timescale to up to 90 days, and a drop-down menu for choosing which report to present from your list of continuous or snapshot reports. You will also see filter options and saved queries to change the scope of which cloud apps are reported (2 in *Figure 16.15*):

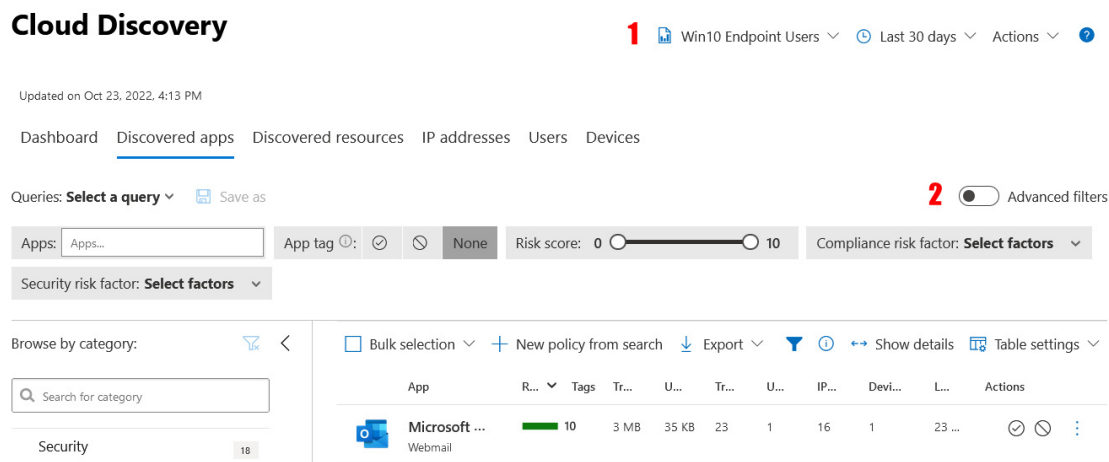


Figure 16.15 – Navigating Cloud Discovery

When we created log collectors or established MDE integration in the *Setting up Cloud Discovery* section earlier, it automatically created the continuous report options we can choose from the drop-down menu. It's possible to create custom versions of these – and therefore finesse your Cloud Discovery filters – by performing the following steps:

1. Go to security.microsoft.com and navigate to **Settings | Cloud Apps | Cloud Discovery | Continuous reports**.
2. Click the + button.
3. Enter a **Report name**, then choose if this combines **All data sources** or only **Specific data sources** (for example, your firewall or MDE devices).
4. Enter a **Filter**, which can be a **user group**, **IP address tags**, or **IP address ranges**. For example, you could create a continuous report for all marketing users in one office or all executives from any data source.

Now that we know how to navigate the catalog, what about taking action on this shadow IT we've discovered? In the subsections that follow, we will review how to tag apps and manage shadow IT policies.

Using tags to monitor, sanction, or unsanction apps

To control access to cloud apps, you can tag them as **Sanctioned** or **Unsanctioned**. You would mark a cloud app as **Sanctioned** after you had verified it was approved for use. The opposite would result in an **Unsanctioned** tag. If integrated with MDE or the third-party SWGs listed earlier, unsanctioning automatically results in a web traffic block. You can also set a tag of **Monitored**, which informs the user that access is being monitored. Alternatively, custom tags can be applied as per your reporting needs. To apply these tags from the **Discovered apps** tab, click the **Actions** button next to a cloud app. This is depicted by three dots, as shown in the following figure. You can also do this in bulk by selecting multiple apps:

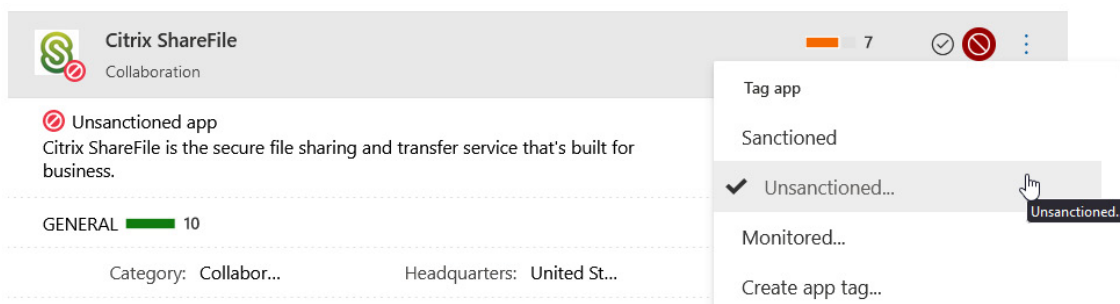


Figure 16.16 – Tagging an app as unsanctioned

When a user running MDE accesses an unsanctioned app, it is blocked in Edge and third-party browsers, though Edge will show SmartScreen-branded warning information. When that user accesses a monitor app, think of this as a *block with override* mode, where they can override the block for the time specified in the MDE integration you configured as part of the *Microsoft Defender for Endpoint and Cloud Discovery* section. This is only available in Edge, and the user experience is depicted in *Figure 16.17*:

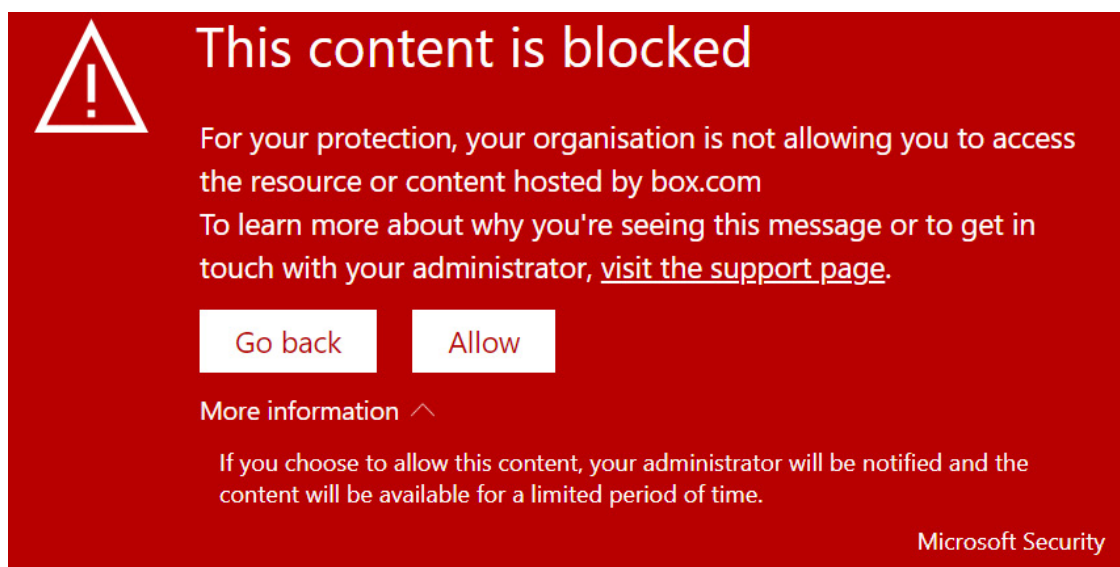


Figure 16.17 – Monitor mode on an MDE-protected device

HOW IT WORKS

*Unsanctioned cloud apps are blocked using automatically populated custom indicators in Microsoft 365 Defender. In SWGs that integrate with MDA, the block is managed by the SWG. For other solutions such as firewall appliances, using the same **Actions** button for tagging cloud apps, you can generate a block script to import manually.*

Applying these actions manually is useful, but we may need a more scalable option. For that, we can create an **app discovery policy**. We'll discuss this in the following section.

Setting up Shadow IT policies

Policies in MDA are ways of combining filter results with actions. Let's have a look at how we can get going with app discovery policies:

1. Browse to security.microsoft.com and navigate to **Cloud Apps | Policies | Policy management**.
2. Click + **Create policy** and, in the dropdown, choose **App discovery policy**.
3. You can start by using several templates or from scratch, entering usual information such as a **Name**, **Description**, and **Policy severity**.
4. In the **Apps matching all of the following** area, you build up your filter based on attributes such as cloud app category, risk factors, legal factors, risk score, and so on. This is combined with the **Apply to** option so that you can choose which continuous reports you are basing this filter on. Lastly, **Apps matching all of the following** is used to specify thresholds. For example, you can reduce noise by only looking for high-volume apps. In the example shown in *Figure 16.18*, we are looking for new risky apps not hosted in the USA that are uploading over 10 MB:

Apps matching all of the following

✕ Risk score ▾ equals 0 — 5

✕ General risk factor ▾ Data center ▾ does not equal ▾ United States ▾

+ Add a filter

Apply to:

All continuous reports ▾

☒ Trigger a policy match if all the following occur on the same day:

apps matching all of the following

Filters:

✕ Uploaded data ▾ greater than 10 MB

Figure 16.18 – Specifying policy filters in MDA

- Now that we've said what apps we want to target, we can specify what we want to do with them. Our first category of options is **Alerts**. This creates a Microsoft 365 Defender alert if the criteria are met, sends an email, or integrates with Power Automate.
- Optionally, we can apply **Governance actions**. For cloud app discovery policies, the options are **Tag app as unsanctioned**, **Tag app as sanctioned**, **Tag app as monitored**, and **Tag app with custom tag**. For example, you may want to tag apps as monitored and raise an alert until you conclude your investigations, at which point you manually sanction or unsanction them. These actions, triggered by policy or manually, typically take a couple of hours to apply on end user devices.

There is one other type of automatic policy for Cloud Discovery: **Cloud Discovery anomaly detection**. This type of policy only creates alerts and does so based on the MDA **SmartEngine**. SmartEngine establishes a normal behavior pattern baseline, then recognizes changes in uploads, downloads, users, and transaction counts. This is useful, for example, if

an app is typically used by users in low volume, then dramatically increases upload by one user: could it represent data exfiltration?

To set up this policy, follow these steps:

1. Browse to security.microsoft.com and navigate to **Cloud Apps | Policies | Policy management**.
2. Click + **Create policy** and, in the dropdown, choose **Cloud Discovery anomaly detection policy**.
3. Optionally, start from a **Policy template** or from scratch.
4. After setting your filter (or none for a universal policy), set **Apply to** for the appropriate reports, users, and/or IPs. The cloud apps are discovered, and they are linked to users and IPs; this selection chooses which to base the policy on.
5. Optionally, set a date from which this applies in the **Raise alerts only for suspicious activities occurring after** calendar pop-out.
6. Set up your alert options, such as alert-only or email. This includes specifying the sensitivity: low, medium, or high. This roughly translates to how closely atypical behavior is identified and, therefore, the number of alerts.

This concludes this section on discovering and managing shadow IT. You're now in a position to ingest network data in MDA, translate it into useful information thanks to the Cloud App Catalog, and take governance actions manually or automatically.

Hopefully, your organization is not dealing exclusively with shadow IT. You will likely be managing cloud apps and services – SaaS and IaaS – such as AWS, Office 365, Azure, Salesforce, and ServiceNow. You also might be using lesser-known or bespoke apps with Azure AD SSO integration. Next up, we'll take a look at what options MDA gives us for managing these cloud apps.

Managing cloud apps with policies

Now that you've mastered how to discover and govern shadow IT, we'll dive into managing cloud apps. MDA offers us two ways to do this, as depicted in the following figure:

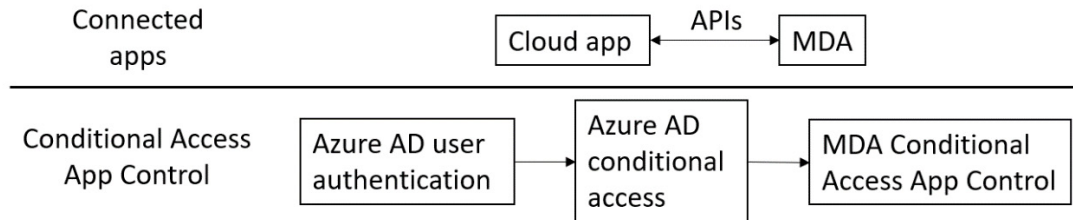


Figure 16.19 – Difference between MDA app control methods

Connected apps is the first method. This feature is for a list of supported cloud apps and integrates them with MDA directly using APIs. This is the highest level of integration with MDA as its CASB capabilities can directly communicate with and control the cloud app to the furthest extent the provider's APIs allow. This list, and the extent of your control, are managed by Microsoft. For example, you could connect Dropbox to govern files saved in it. An advantage of this method is that you achieve deep integration with the cloud app.

Conditional Access App Control is the second method. This is a reverse proxy capability that's integrated with your **identity provider (IdP)**, such as Azure AD for web apps. In Azure AD, a Conditional Access policy is configured to redirect traffic through MDA under the conditions you specify. MDA acts like a man-in-the-middle and instead of accessing the resource directly, it is routed through **mcas.ms**. For example, you could block Office 365 web app downloads on BYOD devices or block access from unauthorized computers altogether. When the user visits outlook.office.com, it is replaced with outlook.office.com.mcas.ms:

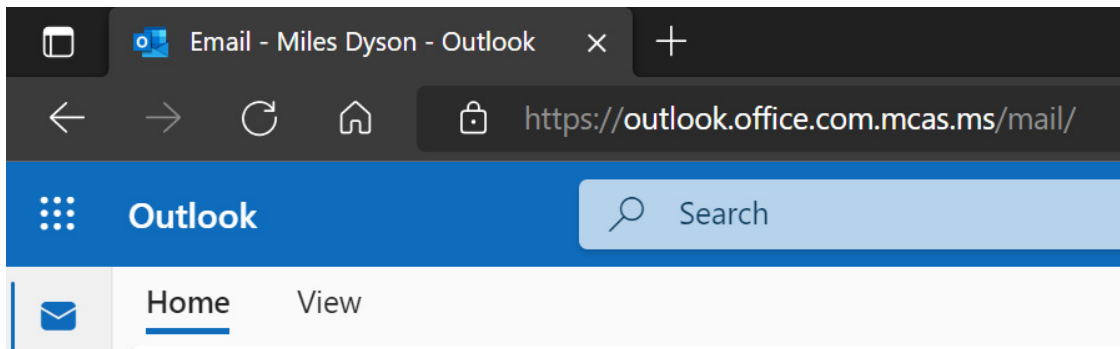


Figure 16.20 – Rerouting MDA traffic through mcas.ms

An advantage of this method is that it does not rely on a Microsoft-managed list.

Now that you’ve been introduced to our two options, let’s have a look at them in more detail, including how to set them up.

Connecting apps to MDA

The list of supported connected apps is ever-growing. At the time of writing, there are over 20. These include the following:

- SaaS solutions such as Atlassian, DocuSign, Salesforce, ServiceNow, Slack, WebEx, and Workday
- IaaS platforms such as AWS, **Google Cloud Platform (GCP)**, and Microsoft Azure
- IdPs such as Okta and OneLogin

When you connect MDA to a cloud app, your options are limited to the APIs the cloud app has available that Microsoft has introduced to MDA. This includes some limitations, such as **throttling**. Once the connection has been established, MDA indexes users, activities, and files in the cloud app. The scanning of these continues through the life of the connected app.

Except for Office 365 and Azure, you can connect a cloud app more than once. For example, large or complex organizations may have multiple instances of Google Workspace.

To connect an app, head to security.microsoft.com, navigate to **Settings** | **Cloud apps** | **App Connectors**, and choose + **Connect an app**. The specifics thereafter for each app differ due to their intricacies, with documentation (including capabilities based on API coverage) for each available at learn.microsoft.com/en-us/defender-cloud-apps/enable-instant-visibility-protection-and-governance-actions-for-your-apps. For example, the wizard for Dropbox simply requires administrator credentials to allow API access; Office 365 can be connected natively. Other apps, such as Salesforce, require more extensive configuration in the cloud app.

With your app connected, you can create policies to protect and audit them.

Activity and file policies

For supported connected apps, you can create policies such as **activity** and **file policies**. These use a filtering mechanism to choose the scope of activities and files you're interested in, then create alerts or automatic governance actions.

For example, an activity policy may look for administrative activity from browsers with strings that don't match your privileged access workstation configuration, then notify other administrators. As an example of a file policy, you could reduce file exposure by finding documents that were last modified over 3 months ago and then removing external users.

Policies like these can be created by following these steps:

1. Navigate to security.microsoft.com | **Cloud apps** | **Policies** | **Policy management**.
2. Click + **Create policy** and choose the policy type (for example, **Activity policy** or **File policy**).
3. Assign the policy a **Policy name**, **Policy severity** (used for Microsoft 365 Defender alerts), **Category**, and **Description**.
4. Build the criteria for the policy using the **Activities matching all of the following** options.

5. Specify your preferences for **Alerts**.
6. Lastly, complete the **Governance actions** section. The available actions vary by the connected app's support for them. Using the file policy example referenced earlier, actions such as **Remove external users** for OneDrive for Business and SharePoint Online could be used:

Governance actions

Dropbox

Microsoft OneDrive for Business

- ☐ Send policy-match digest to file owner ⓘ
- ☐ Notify specific users
- ☐ Make private
- ☒ Remove external users
- ☐ Inherit parent permissions
- ☐ Put in user quarantine
- ☐ Put in admin quarantine [Configure a quarantine folder](#) to enable this option
- ☐ Trash
- ☐ Remove a collaborator
- ☐ Apply sensitivity label ⓘ
- ☐ Remove sensitivity label

Microsoft SharePoint Online

Figure 16.21 – Specifying Governance actions in an MDA policy

7. Click **Create** to save the policy. Now, matches for this policy will trigger the automatic governance and alert you in Microsoft 365

Defender.

Connected apps and their policies are only part of the MDA control capabilities. We can continue to protect apps that don't have API integrations by leveraging **Conditional Access App Control**, which will be covered in the next section.

Managing access and session controls with Conditional Access App Control

Conditional Access App Control can be used with Open ID Connect or SAML 2.0 interactive sign-on apps in Azure AD, including on-premises apps made available using Azure AD Application Proxy.

AZURE AD IS THE HEART OF MICROSOFT 365

Although third-party IdPs are supported for Conditional Access App Control, you're most likely using Azure AD, and that will be the focus of this book's MDA integration guides. One point to be aware of is that third-party IdPs do not support Open ID Connect for Conditional Access App Control.

Simply put, if you have an enterprise app in Azure AD set up for SSO, you can leverage Conditional Access App Control policies. It requires Azure AD Conditional Access and therefore, at a minimum, requires users to have a license for both MDA and Azure AD Premium P1.

The process of getting Conditional Access App Control up and running looks like this:

1. Configure Conditional Access App Control's fundamental settings.
2. Add your application(s) to Conditional Access App Control.
3. Create Conditional Access App Control policies.

To start our Conditional Access App Control usage, we'll first cover *steps 1* and *2*. Then, in the *Managing access policies* and *Managing session poli-*

cies subsections, we'll cover *step 3*, by reviewing what policies there are and how to set them up.

Configuring fundamental settings

The settings we're about to review lay the foundations for Conditional Access App Control. Let's check them out:

1. Browse to security.microsoft.com and navigate to **Settings | Cloud Apps | Conditional Access App Control**.
2. In the **Default behavior** section, you have options to **Allow access** or **Block access** if the MDA reverse proxy service is down. This comes down to your risk acceptance. The cautious option is to **Block access**.
3. In the **User monitoring** section, you can choose if there should be a welcome warning screen when users sign into their reverse-proxied web cloud apps and, if so, what a custom message should be. What this looks like is depicted in the following figure:

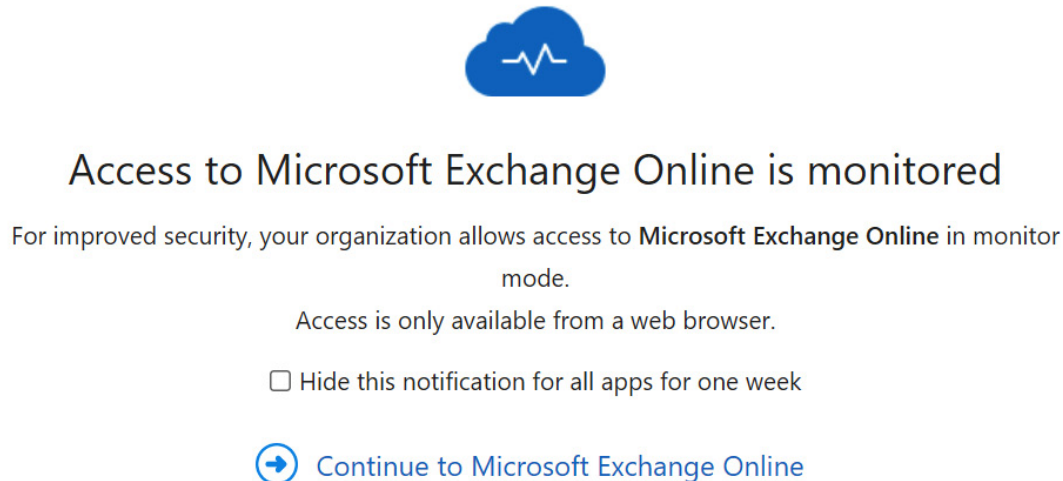


Figure 16.22 – The MDA Conditional Access App Control warning screen after signing in

4. In the **Device identification** section, you'll add trusted root or intermediate certificates (with the **.pem** extension) so that you can identify devices that have these. For example, you can apply different rules based on the presence (or lack thereof) of different certificates. On your client devices, the certificates should be in the user store.

DEVICE IDENTIFICATION DEPENDENCIES

Modern browsers support querying client certificates, but you should do testing to confirm this before production deployments. If you are checking device compliance or domain join state, this will likely require additional configuration for third-party browsers. Google Chrome, for example, needs the Windows Accounts extension.

5. In the last section, **App onboarding/maintenance**, you can list users who will onboard Conditional Access App Control apps. We recommend using a service account for this process.

With our base settings for Conditional Access App Control out the way, let's move on to onboarding apps.

Adding Conditional Access App Control apps

Before we can start protecting our SAML 2.0/Open ID Connect SSO apps with Conditional Access App Control, we must perform an onboarding operation in Azure AD. Remember, other IdPs are supported, but we'll focus on Azure AD. All apps you want to protect with Conditional Access App Control should go through the process that we'll discuss in this section.

There are some apps that Microsoft has simplified this for, known as **catalog apps** or **pre-onboarded apps**. This includes popular Microsoft apps such as Exchange Online, SharePoint Online, and Teams, as well as third-party apps such as Box, Salesforce, and Workday. For other apps, known in this context as **custom apps**, there are some additional steps to follow.

Let's get started by looking at an example of onboarding Dropbox, a catalog app:

1. Our first step is to create an Azure AD Conditional Access policy for the enterprise app. Head to entra.microsoft.com and navigate to **Protect & Secure | Conditional Access | Policies**.
2. Click + **New policy**.

3. Azure AD Conditional Access policies have five sections. Here's what you need to complete for each:
1. For **Users**, include the account you specified in the **App onboarding/maintenance** settings.
 2. For **Cloud apps or actions**, choose your enterprise. In our example, this is Dropbox.
 3. You can ignore the **Conditions** and **Grant** sections.
 4. For **Session**, tick the box next to **Use Conditional Access App Control** and, from the dropdown, choose **Monitor only**:

Session

×

☐ Use app enforced restrictions ⓘ

ⓘ This control only works with supported apps. Currently, Office 365, Exchange Online, and SharePoint Online are the only cloud apps that support app enforced restrictions. [Click here to learn more.](#)

☒ Use Conditional Access App Control ⓘ

Monitor only (Preview) ▾

Monitor only (Preview)

Block downloads (Preview)

Use custom policy...

Figure 16.23 – Enabling Conditional Access App Control in Azure AD Conditional Access

4. For **Enable policy**, choose **On** mode and then **Save** your policy.

5. Azure AD Conditional Access can take some time to become active, so allow an interval for this.
6. Using the account you specified onboarding for, log in to the enterprise app using SSO. You'll find that the URL has been rewritten so that it includes **mcas.ms**. If this doesn't happen, you may need to give the Azure AD Conditional Access policy more time to become active.
7. Head to **security.microsoft.com** and navigate to **Settings | Cloud apps | Connected apps | Conditional Access App Control apps**. You'll find your enterprise app and that the **Status** column confirms if onboarding was successful:

Conditional Access App Control apps

The Conditional Access App Control adds real-time monitoring and control capabilities for your apps. To enable Conditional Access App Control capabilities on your apps, follow the [deployment instructions](#).

 [Device identification settings](#)

Filters:

☒ Advanced filters

App: **Select apps**

App category: **Select category**

Last connected:

Select a date

+

 Add

↓

 Export

ⓘ

⌵

 Hide filters

⌄

 Table settings

⌵


App ^	Status	Available controls ⓘ	Was connect...	Last activity
<div> <div>  <div> <div>Dropbox - General</div> <div>Cloud storage</div> </div> </div> </div>	<div> <div>✔</div> <div>Connected</div> </div>	<div> <div>Azure AD conditional access</div> <div>Session control</div> </div>	<div>30 Oct 2022...</div>	<div>30 Oct 202...</div> <div>⋮</div>

Figure 16.24 – Onboarded Conditional Access App Control apps

The app is now available for the policies you'll learn about in the *Managing access policies* and *Managing session policies* subsections.

8. If you need to add a custom app, on the same page that was mentioned in *step 7*, choose + **Add** and follow the configuration wizard.

Now that apps have been onboarded to Conditional Access App Control, it's time to protect them! Two types of policy are available for this: **access** and **session**. We'll explore access policies first and learn how they supplement Azure AD Conditional Access to restrict access in authorized circumstances.

Managing access policies

Access policies are used to block access based on filter criteria. This might include checks against the following:

- Device types such as PC, mobile, tablet, and others
- Device tags such as Intune-compliant, Hybrid Azure AD joined, or with a valid certificate
- IP addresses or categories such as VPNs, cloud providers, or geo-IP location
- User agents such as specific strings or outdated browsers and OSs

While several of these use cases can be achieved with Azure AD Conditional Access and do not require MDA, MDA offers more filtering capabilities, such as outdated system blocking or checking for a specific certificate. The latter has use cases such as limiting the access that users such as contractors or independents have to only devices you have deployed a certificate to, without the need to fully manage that device (as a Conditional Access device compliance policy would require). Or in another scenario, you could restrict access to cloud app administration panes unless a specific certificate is present for tiered administration.

To create an access policy, follow these steps:

1. Head to security.microsoft.com and navigate to **Cloud Apps | Policies | Policy management**.
2. Click + **Create policy** and, in the drop-down menu, choose **Access policy**.
3. After giving your policy a **Name**, **Severity**, **Category**, and optional **Description**, build your **Activities matching all of the following** as required, based on what you want to restrict:

Policy name *

Block Devices Without Certificate

Policy severity *

Category *

Access control

Description

Blocks devices without the client certificate.

Activities matching all of the following

Device Tag does not equal Valid client certificate

Figure 16.25 – Creating an access policy

By default, only web browser sessions are in scope. **Client app equals mobile or desktop app** must be explicitly configured if that's your intended use case.

4. In **Actions**, choose to deploy the policy in **Test** or **Block** mode.
5. In **Alerts**, choose if and which types of alerts should be created, including daily limits. By default, each access policy event creates an alert in Microsoft 365 Defender.

Remember, the filters you specify are applied after the conditions of the Azure AD Conditional Access policy that directs traffic to the MDA reverse proxy have been applied. If the Azure AD Conditional Access policy and MDA access policy filters are both matched and you chose to block access, the user will authenticate successfully but will be told that their access has been blocked:



Access to Office Portal is blocked

Access to Office Portal is blocked by your organization's security policy.

Figure 16.26 – MDA access policy enforcing a block

Access policies are a *sledgehammer* approach: they block access completely. The next type of Conditional Access App Control policy allows more nuance, so let's check it out.

Managing session policies

Session policies offer a way to allow access but limit what type of behavior is permitted. For example, you may want to block data exfiltration attempts such as download, cut/copy, or print. Alternatively, you can allow the download but apply file-level encryption with a Microsoft Purview Information Protection sensitivity label. Even more refined, you could choose to do this based on content inspection, with different rules depending on data sensitivity.

The example we'll demonstrate configuring session policies for is the most common: blocking downloads on unmanaged devices, therefore allowing **bring-your-own-device (BYOD)** access but limiting the risk of data loss. We will need to perform the following steps:

1. Once again, navigate to security.microsoft.com and then to **Cloud Apps | Policies | Policy management**.
2. Click + **Create policy** and, in the drop-down menu, choose **Session policy**.
3. Several templates are available in the **Policy template** dropdown, but we'll create one from scratch so that you understand what's involved.
4. Enter a **Policy name**, **Policy severity**, **Category**, and **Description**.

5. In the **Session control type** dropdown, you can choose between **Monitor only**, **Block activities**, **Control file download (with inspection)**, or **Control file upload (with inspection)**. You can only select one per policy. Note that although the download and upload options imply inspection, this is optional, and if you select them without inspection details, you apply them to all files.

In our example of blocking downloads on BYOD devices, we'll choose **Control file download (with inspection)**:

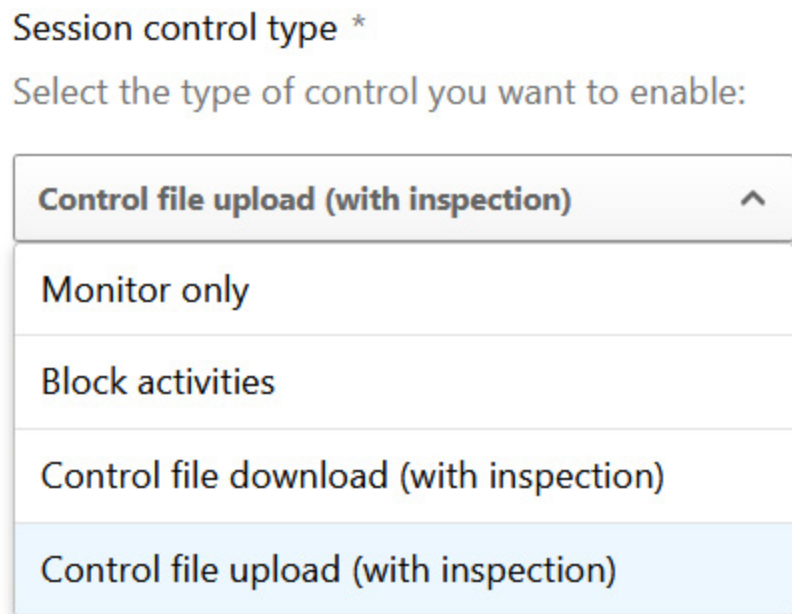


Figure 16.27 – Session control types available in MDA

6. Proceed to populate the **Activities matching all of the following** section. This is similar to the area you learned about in the *Managing access policies* section but with one addition: if you choose a **Block activities** control type, you can also filter on **Activity type**, which offers settings such as **Cut/Copy item**, **Paste item**, **Print**, and **Send Teams message**.

Our example policy applies to BYOD devices, so we'll set the filters to **Device Tag does not equal Intune compliant** and **App equals Office 365**.

7. Optionally, for file controls, you can apply **Files matching all of the following filters** to attributes such as the extension, name, or size, or if the file has a sensitivity label.
8. There are three optional **Inspection method** options: **Built-in DLP**, **Data Classification Service**, and **Malware detection**. Of the two DLP options, the **Data Classification Service** option is generally regarded as superior as it's the integrated service across Microsoft 365 and Microsoft Purview for **sensitive information types (SITs)**, **exact data matches (EDMs)**, and **trainable classifiers**.

In our example, we'll choose **None** for **Inspection method**, which means the actions we'll specify next are applied all the time.

9. In the **Actions** section, you can choose to enable **Test**, **Block**, **Protect**, or **Require step-up authentication**. In our example, we'll choose **Block**, but what do each of these mean? Let's find out:
 1. **Test** mode will create entries in the activity log, but there's no enforcement. Think of it like PowerShell's **-WhatIf** option.
 2. **Block** mode stops the activity, download, or upload. When a user downloads a file, for example, MDA intercepts the download and replaces it with a stub text file, and also informs the user with an (optional) customized message.
 3. **Protect** mode for downloads will apply an existing Microsoft Purview Information protection sensitivity label or custom protection. Only modern Word, Excel, and PowerPoint format files are supported, as well as PDFs. Optionally, you can choose **Block download of any file that is unsupported by native protection or where native protection is unsuccessful**.
 4. **Require step-up authentication** is used in conjunction with Azure AD's authentication context. If, for example, you have a SharePoint Online site with a sensitivity label that has an authentication context linked to it, and the user visits that SharePoint Online site during an MDA session, it forces the CA policy to be re-evaluated, whereas normally this only happens during sign-in.

5. For all action choices, you can choose to **Always apply the selection action even if data cannot be scanned**, which is a recommended setting.
10. Finally, before saving the policy, specify your **Alert** options. By default, Microsoft 365 Defender alerts are created for any policy matches, and you can optionally also get an email or limit the number of alerts per day.

With our policy configured to block downloads on devices that aren't Intune-compliant (often used to determine if it is a BYOD device), the user will now see the error depicted in *Figure 16.28* when they attempt to download. Remember, both the conditions in a Conditional Access policy and filters in the MDA session policy must match the user's scenario:

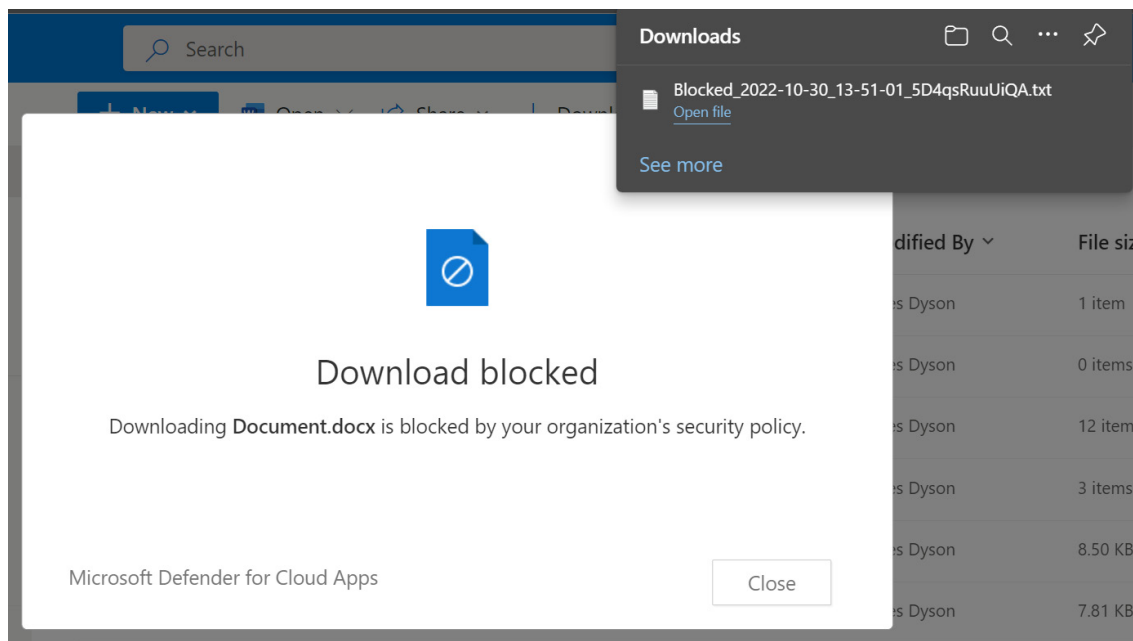


Figure 16.28 – A download blocked by an MDA session policy

As with all MDA policies, alerts are also created in Microsoft 365 Defender for your review and response, which you'll learn more about in [*Chapter 18*](#).

This concludes our configuration review of connected apps and Conditional Access App Control. In the next section, we'll cover one more important area of our cloud estate that MDA helps us protect: OAuth apps.

Governing OAuth apps

OAuth 2.0 is a standard for delegating access to app resources. Think of it as *application X* getting permission to certain parts of *application Y*. We call the app that receives this access an **OAuth app**. If you've ever given an app permission to access or manage your name details, mailbox, and calendar information, you've probably used OAuth 2.0 to do so. This is also true for any PowerShell scripts that interact with the Microsoft Graph API.

This is a great advancement from the historical way of giving access, which involved handing over your username and password, giving unrestricted abilities to the app. But it has its challenges. Attackers love persistence, as you learned about when we reviewed MITRE ATT&CK in [*Chapter 1*](#). OAuth apps do not require login credentials to get access, and unless access is revoked, they will continue to get delegated access. This means two of the main threats you'll see with OAuth apps are malicious apps, which trick users into giving them access, and supply chain compromise, where a legitimate OAuth app is hijacked by the attacker.

MDA offers us the following OAuth app capabilities:

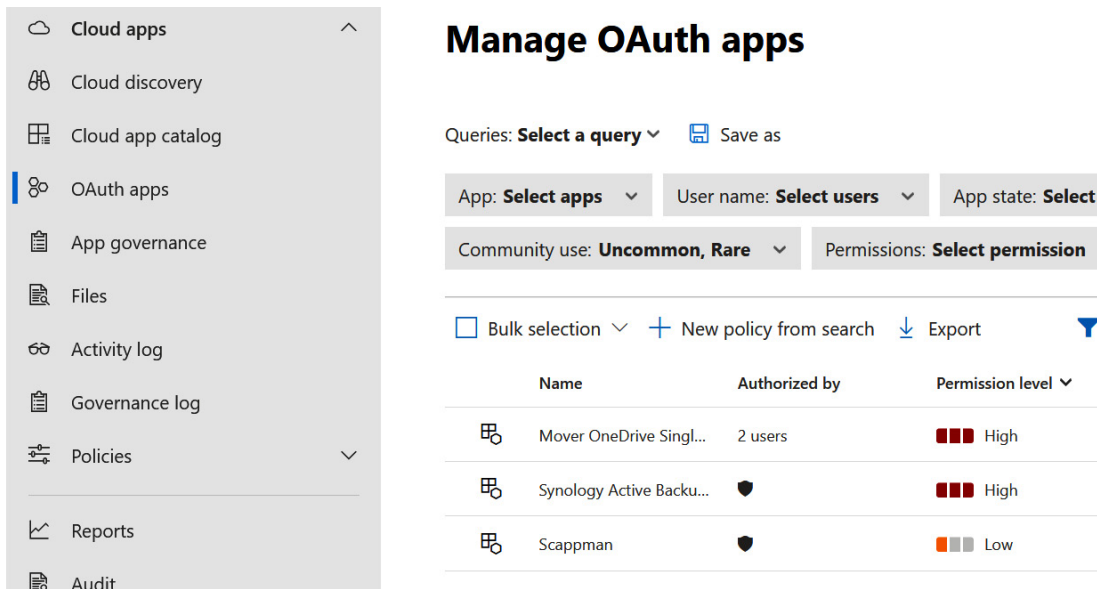
- Included with the standard MDA license, OAuth apps accessing Office 365, Google Workspace, or Salesforce will present app information and usage. You can make policies to protect your environment from potentially risky OAuth apps, as well as use the data provided to perform hunting and incident response. Consider standard capabilities scope as *app use*.
- By purchasing the **app governance** add-on license, you get additional insights into and control over OAuth apps accessing the Microsoft Graph API. This includes over-permissioning and data consumption, where changes in these may indicate supplier compromise. Consider app governance's capabilities as *app behavior*.

We'll begin by reviewing the capabilities available as standard.

Managing OAuth apps and policies

As you learned in the *Connecting apps to MDA* section, once Office 365, Google Workspace, Salesforce, and other apps are connected, MDA can start to see the OAuth apps and permissions associated with them.

To review the entire list of discovered OAuth apps, head to security.microsoft.com and navigate to **Cloud apps | OAuth apps**. You'll find a filterable list that, in the following figure, has been set to show apps with uncommon or rare community use, based on Microsoft's telemetry across all managed environments:



Name	Authorized by	Permission level
Mover OneDrive Singl...	2 users	High
Synology Active Backu...		High
Scappman		Low

Figure 16.29 – The OAuth apps page in the Microsoft 365 Defender portal

Clicking on each OAuth app provides further information, such as the permissions delegated, publisher, redirect URLs, and links to view associated entries in the activity log. The permissions the OAuth app has are consolidated into a low, medium, or high **Permission level**. This, combined with **Community use**, can be useful for prioritizing OAuth app investigation or management.

For each app, you also have the option to **Mark app as approved** or **Mark app as banned**. The latter revokes its permissions and you can, op-

tionally, notify the user who granted access.

What about automating our OAuth app management? Browse to security.microsoft.com and then to **Policies | Policy management** to see the built-in **OAuth app anomaly detection policies** from Microsoft. This includes **Misleading publisher name for an OAuth app** and **Malicious OAuth app consent**. You can't edit how these policies identify the risk, but you can set a **Governance action** of **Revoke app**. Alternatively, you can create your own policy by choosing + **Create policy** and selecting **OAuth app policy**.

These built-in policies and the extras you create help you wrestle with the risk of OAuth apps. However, many more capabilities open up with the app governance add-on, which we will discuss in the next section.

App governance

The app governance feature can be found at security.microsoft.com under **Cloud apps | App governance**. If you haven't already purchased it, you'll be given the option of a free trial.

App governance populates app information and alerts based on behavior, so give it time to populate. After some time has passed, on this same page, you'll find a dashboard of information that will include things such as data usage and incidents.

You can use the **Apps** tab to view a list of OAuth apps registered in Azure AD. Clicking on each provides information about the last 30 days of usage, including data consumption, users, permissions (both granted and unused), and if they have interfaces with sensitivity label-protected files. Of particular interest in each app is the publisher verification and certification information to establish legitimacy, and the **Disable app** button to easily revoke permission.

The **Policies** tab contains several out-of-the-box policies. By default, built-in policies only alert you in Microsoft 365 Defender; they do not perform any remediation, but you can edit them and choose the **Disable app** action. Examples of these include policies that check for spikes in Graph API calls or with high volumes of exfiltration-like behavior. Each policy can be turned off or set to a specific OAuth app scope too if, for example, you want to exclude certain ones.

The **Create new policy** button on this page launches a wizard. There are several templates you can use, or you can customize them entirely using **Conditions**. The list of conditions is vast, but significant options include filtering for specific application permissions (for example, **Directory.ReadWrite.All**), data usage in GB, or the presence of unused Graph API permissions. Optionally, you can apply the **Disable app** action for policy matches. Policies can be deployed immediately, created but not enabled, or set to audit mode to test their results.

That brings our exploration of MDA's OAuth capabilities to an end. It also concludes this chapter's review and guidance for implementing and controlling MDA. Now, let's remind ourselves of what we achieved via a summary.

Summary

Reading through this chapter, you'll have learned about a lot of security and compliance options for cloud apps. Acting as Microsoft's CASB, MDA is a service that all customers who are licensed to it are encouraged to make the most of.

To recap, we started this chapter by laying a solid foundation for MDA by covering fundamental settings. We continued by discussing its shadow IT discovery and management capabilities. By taking control of your environment's shadow IT, you reduce the risk of regulatory failures and cybersecurity threats. Then, we moved on to establishing control of con-

nected cloud apps with policies. This is important because it means you can refine users' abilities when accessing apps, such as different levels of permission when accessing apps on BYOD devices. Our review of cloud apps then sprawled over to OAuth apps, including add-on licensing options for enhanced capabilities, so that you can make sure permissions are appropriately assigned to these apps, minimize the use of risky ones, and track their behaviors over time.

In the next chapter, our focus on securing the environment will continue as we venture into Microsoft 365 Defender's threat and vulnerability management features.

Questions

If you want to test your understanding of what you learned in this chapter, have a go at the following questions about MDA:

1. If you have internal or regulatory compliance reasons to protect the activities of some members of your organization from general administrators, which of the following would be the most appropriate feature to configure?
 1. Create a file policy
 2. Connect Microsoft Priva to MDA
 3. Configure activity privacy
 4. Create an activity policy
2. Which of the following is a benefit of MDA integrating with MDE?
Choose all that apply.
 1. Prevent access to unsanctioned websites
 2. Control all network traffic with a reverse proxy
 3. Control endpoint DLP to unauthorized network shares
 4. Leverage endpoint network traffic for shadow IT discovery
3. True or false: the only ways to determine device state with a Conditional Access App Control session policy are Intune compliance and Hybrid Azure AD join.

1. True
2. False
4. You have used the activity log to create a query using the app and activity type filters. You want to be alerted about any new activities that match your results. Which of the following options in the activity log could you use?
 1. Choose a suggested query
 2. Create a new policy from search
 3. Create a continuous report
 4. Create an advanced filter
5. Which of the following are prerequisites for Conditional Access App Control? Choose all that apply.
 1. Microsoft Defender for Identity must be licensed
 2. Azure Active Directory Premium P1 must be licensed
 3. The app must use SAML 2.0
 4. The app must use Windows Integrated Authentication

Further reading

If you want to learn even more about MDA, check out the following recommended reading:

- As a general go-to resource for MDA, you should be following Sami Lamppu's blog, as he regularly shares very interesting articles and discoveries on all things MDA: samilampuu.com.
- You learned about the ability to integrate MDA with external DLP solutions. For a deep dive into configuring it, you should refer to the official documentation: learn.microsoft.com/en-us/defender-cloud-apps/icap-stunnel.
- Want to centralize the management of trusted/named locations? Thijs Lectome has you covered, with this blog on ingesting IP ranges into MDA using Azure Automation: 365bythijs.be/2020/03/31/sync-named-locations-to-mcas-ip-ranges-using-azure-automation/.

- MDA never stays still. Keep on top of the regular changes by monitoring the *What's New?* page: learn.microsoft.com/en-us/defender-cloud-apps/release-notes.