

MVP

# Global Summit

March 29 – 31, 2022

Learn more at [summit.microsoft.com](https://summit.microsoft.com)



# Microsoft Confidential

---

Please note, all the content you hear today is considered under NDA unless otherwise stated by the presenters. Please do not blog, tweet, or share any content publicly.

---

If you have any questions as to whether content can be shared, please ask the presenters. We encourage you to use the meeting chat at any time to ask questions regarding content or to request support.

---

At the end of the session, don't forget to respond the survey. We appreciate your feedback as we use this to improve future events.



# Inclusive Session Guidelines

Please practice these tips while in Summit Sessions.

## Code of Conduct

---

Be welcoming and respectful

Be aware of others

Be open to all questions & viewpoints

Be understanding of differences

Be friendly and patient

Be kind and considerate to others

## Audience Guidance

---

Use raise hand to ask a question

Questions in chat to be prefaced with Q:

@mention to respond to an individual

Return to mute when not talking

Use alt-text for images and gifs

Immersive reader can be found by clicking on ... in the meeting chat

Visit the full Event Code of Conduct or report an issue by visiting [aka.ms/SummitCoC](https://aka.ms/SummitCoC)

## Speaker Bio



### **Corissa Koopmans**

Her/She

Senior Program Manager

**Identity Network & Access Management**

[CoKoopma@microsoft.com](mailto:CoKoopma@microsoft.com)



## Speaker Bio



**Tosin Lufadeju**

He/Him

Program Manager

**Identity Network & Access Management**

[Tolufade@microsoft.com](mailto:Tolufade@microsoft.com)



# Azure AD Workbooks

## Lessons learned and how you can contribute



## Session learning Objective

The objective of this session is to share what we have done, what's new and what we need from you, the MVP Community!

# Agenda

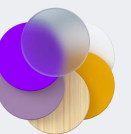
- Introduction
- Azure AD Workbooks overview
- Business use cases and new workbooks
- Lessons learned in Log Analytics and KQL
- Our ask to the MVP Community



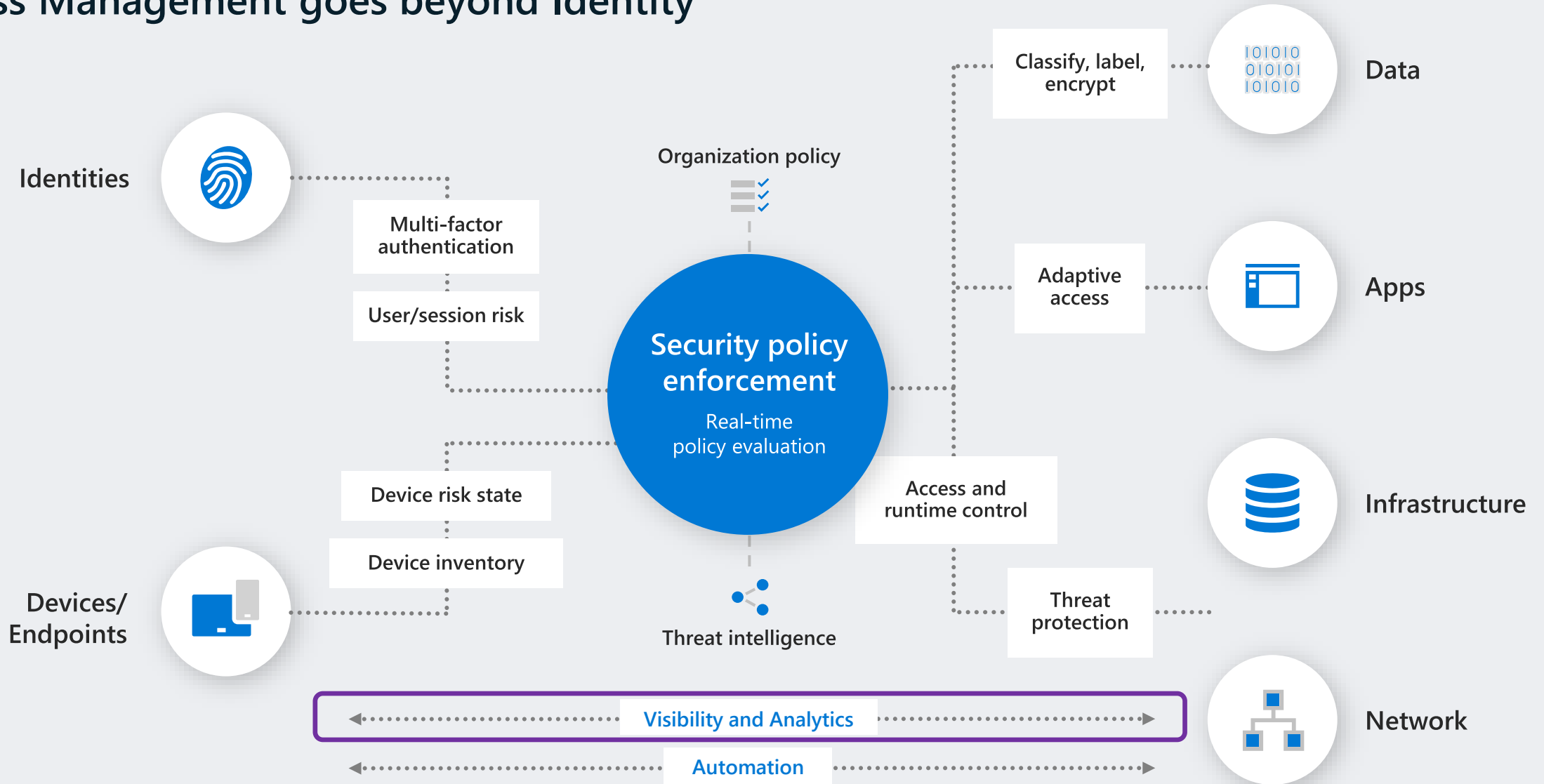
## Previous MVP Summit Presentations ([aka.ms/KQLMVP](https://aka.ms/KQLMVP))

- 2020: EM-053 Log Analytics Wizardry (Beginner)
- 2021: EM-003/004 KQL: From Hero to Superhero (Intermediate)
- 2022: EM-014 Lesson learned from new Azure AD Workbooks and how you can contribute to them (Intermediate/Advanced)

All resources shared in the presentation can be found at [aka.ms/KQLMVP](https://aka.ms/KQLMVP)



# Access Management goes beyond Identity



# Azure AD Workbooks overview

The screenshot shows the Azure AD Workbooks interface. The left sidebar contains a navigation menu with categories like Monitoring, Log Analytics, and Troubleshooting + Support. The main content area displays a grid of workbook templates. Annotations with arrows point to specific features:

- Create a new workbook.** Points to the '+ New' button in the top toolbar.
- Discover workbooks with tabs and search.** Points to the 'Filter by name or category' search bar and the 'Workbooks' tab.
- Open a workbook template; use it as-is or edit it to suit your needs.** Points to a workbook template card in the grid.
- Set up data to flow into Log Analytics.** Points to the 'Log Analytics' option in the left sidebar.
- Browse workbook templates in the Azure AD Workbooks gallery.** Points to the 'Workbooks' option in the left sidebar.

The interface includes a top navigation bar with the Microsoft Azure logo, a search bar, and various utility icons. The main content area has tabs for 'All', 'Workbooks', 'Public Templates', and 'My Templates'. A message at the top states: 'Private and favorite Workbooks are deprecated and not accessible in Workbook gallery. If you want to retrieve them, follow these instructions. →'. Below this, there are filters for 'Subscription' and 'Resource Group'. The grid of templates is organized into sections: 'Quick start' (with an 'Empty' workbook), 'Recently modified workbooks (0)', 'Usage (10)', 'Conditional access (5)', and 'Health (1)'.

[Home](#) > [Woodgrove](#) >

# D diagnostic setting



Save Discard Delete Feedback

A diagnostic setting specifies a list of categories of platform logs and/or metrics that you want to collect from a resource, and one or more destinations that you would stream them to. Normal usage charges for the destination will occur. [Learn more about the different log categories and contents of those logs](#)

Diagnostic setting name Woodgrove\_Diagnostics\_Settings

## Logs

### Categories

<input checked="" type="checkbox"/> AuditLogs	Retention (days) 180
<input checked="" type="checkbox"/> SignInLogs	Retention (days) 180
<input checked="" type="checkbox"/> NonInteractiveUserSignInLogs	Retention (days) 180
<input checked="" type="checkbox"/> ServicePrincipalSignInLogs	Retention (days) 180
<input checked="" type="checkbox"/> ManagedIdentitySignInLogs	Retention (days) 180
<input checked="" type="checkbox"/> ProvisioningLogs	Retention (days) 180
<input checked="" type="checkbox"/> ADFSSignInLogs	Retention (days) 180
<input checked="" type="checkbox"/> RiskyUsers	Retention (days) 180
<input checked="" type="checkbox"/> UserRiskEvents	Retention (days) 180
<input checked="" type="checkbox"/> NetworkAccessTrafficLogs	Retention (days) 180
<input checked="" type="checkbox"/> RiskyServicePrincipals	Retention (days) 180
<input checked="" type="checkbox"/> ServicePrincipalRiskEvents	Retention (days) 180

### Destination details

☒ Send to Log Analytics workspace

#### Subscription

Woodgrove - GTP Demos (External/Sponsored)

#### Log Analytics workspace

Woodgrove-LogAnalyticsWorkspace ( westus2 )

☒ Archive to a storage account

*i* Showing all storage accounts including classic storage accounts

#### Location

All

#### Subscription

Woodgrove - GTP Demos (External/Sponsored)

#### Storage account \*

woodgrovesigninstorage

☒ Stream to an event hub

For potential partner integrations, [click to learn more about event hub integration](#).

#### Subscription

Woodgrove - GTP Demos (External/Sponsored)



## Business Use Cases

- Executive complains about too many MFA prompts
- Microsoft announces the deprecation of Legacy TLS
- Transitioning from ADFS to Azure AD Authentication
- Customer wants to understand their cross-tenant activity



# A look at new Azure AD Workbooks



[Home](#) > [Woodgrove](#)

# Woodgrove | Workbooks | Gallery

Azure Active Directory

- Overview
- Preview features
- Diagnose and solve problems

## Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations
- Identity Governance
- Application proxy
- Custom security attributes (Preview)
- Licenses
- Azure AD Connect
- Custom domain names
- Mobility (MDM and MAM)
- Password reset
- Company branding
- User settings

[+ New](#) [Refresh](#) [Feedback](#) [Help](#) [Community Git repo](#) [Browse across galleries](#)[All](#) [Workbooks](#) [Public Templates](#) [My Templates](#)Subscription : **All**Resource Group : **All**[Reset filters](#)

### Quick start

**Empty**

A completely empty workbook.

### Recently modified workbooks (0)

No items found.

### Usage (10)

**Sign-ins using Legacy Aut...****Sign-ins****Access Package Activity****App Consent Audit****SSPR Reset Funnel****Sign-In Analysis (Preview: ...****Identity Protection Risk An...****Authentication Prompts A...**  
Monitor authentication prompts ...**Tenant restriction insights****Cross-tenant access activity**

### Conditional access (5)

**Conditional Access Insight...**  
Monitor the impact of your Cond...**Continuous access evaluat...****Sign-ins by Conditional Ac...****Sign-ins by Grant Controls...****Conditional Access Gap A...**

### Troubleshoot (4)

[Home](#) > [Woodgrove](#)

# Woodgrove | Workbooks | Gallery

Azure Active Directory

- Overview
- Preview features
- Diagnose and solve problems

## Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations
- Identity Governance
- Application proxy
- Custom security attributes (Preview)
- Licenses
- Azure AD Connect
- Custom domain names
- Mobility (MDM and MAM)
- Password reset
- Company branding
- User settings

[+ New](#) [Refresh](#) [Feedback](#) [Help](#) [Community Git repo](#) [Browse across galleries](#)[All](#) [Workbooks](#) [Public Templates](#) [My Templates](#)Subscription : **All**Resource Group : **All**[Reset filters](#)

### Quick start

**Empty**

A completely empty workbook.

### Recently modified workbooks (0)

No items found.

### Usage (10)



Sign-ins using Legacy Aut...



Sign-ins



Access Package Activity



App Consent Audit



SSPR Reset Funnel



Sign-In Analysis (Preview: ...)



Identity Protection Risk An...

Authentication Prompts A...  
Monitor authentication prompts ...

Tenant restriction insights



Cross-tenant access activity

### Conditional access (5)

Conditional Access Insight...  
Monitor the impact of your Cond...

Continuous access evaluat...



Sign-ins by Conditional Ac...



Sign-ins by Grant Controls...



Conditional Access Gap A...

### Troubleshoot (4)



- Overview
- Preview features
- Diagnose and solve problems

Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations
- Identity Governance
- Application proxy
- Custom security attributes (Preview)
- Licenses
- Azure AD Connect
- Custom domain names
- Mobility (MDM and MAM)
- Password reset
- Company branding
- User settings
- Properties
- Security

Monitoring

Authentication Prompts Summary

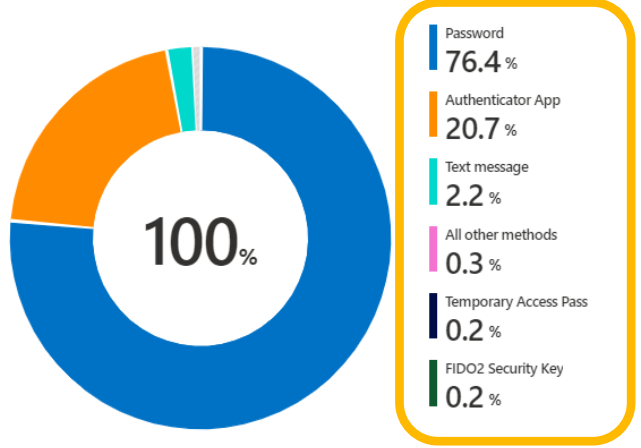


Count of authentication methods excludes "previously satisfied" and "unknown" methods.

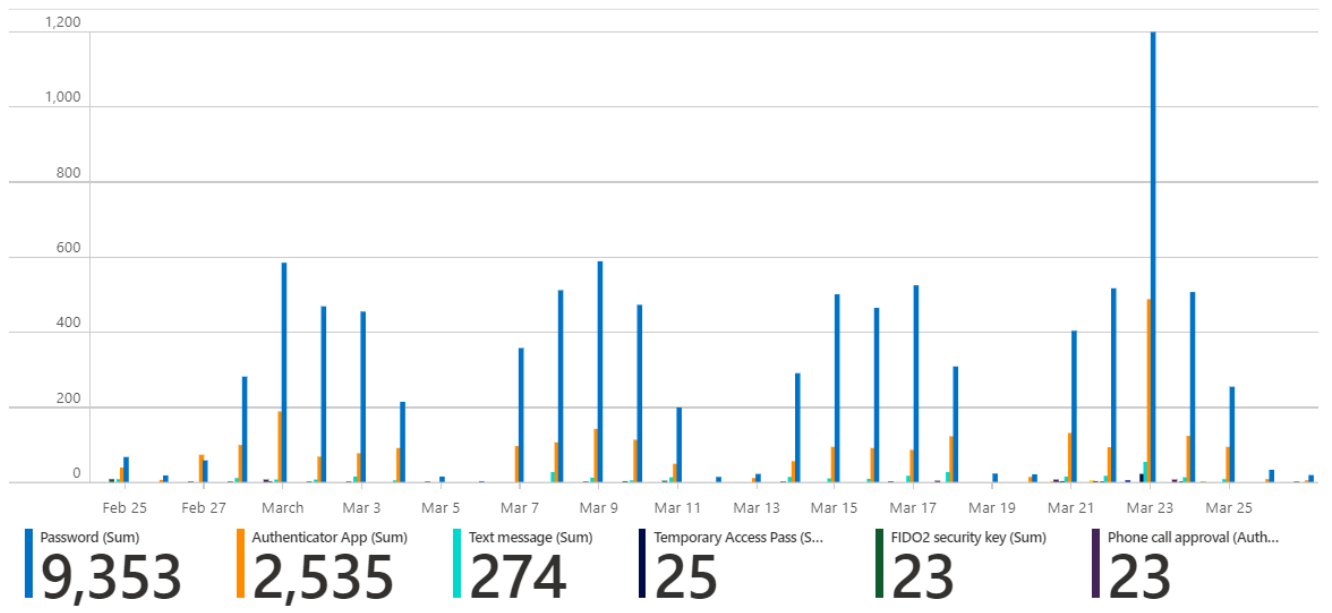
Authentication prompts by authentication method

To investigate methods causing the most prompts, filter this report by AuthMethod.

% Prompts by method



Daily prompts by method



- Overview
- Preview features
- Diagnose and solve problems

Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations
- Identity Governance
- Application proxy
- Custom security attributes (Preview)
- Licenses
- Azure AD Connect
- Custom domain names
- Mobility (MDM and MAM)
- Password reset
- Company branding
- User settings
- Properties
- Security

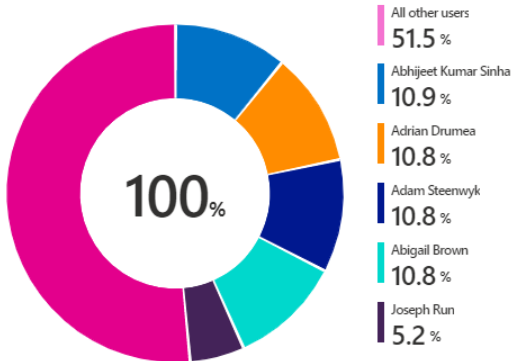
Monitoring

- Sign-in logs
- Audit logs
- Provisioning logs
- Log Analytics

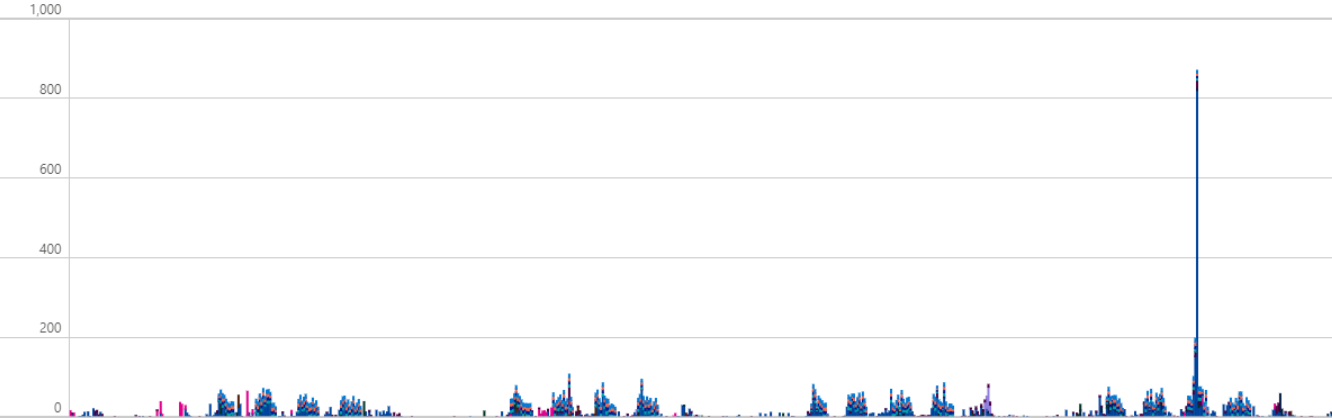
Authentication prompts by user

To investigate the top prompted users, filter this report by user.

% Prompts by user



Hourly prompts by user



Abhijeet Kumar Sinha (Sum)	Adrian Drumea (Sum)	Adam Steenwyk (Sum)	Abigail Brown (Sum)	Joseph Run (Sum)	Cedric Blomart (Sum)	Joey Cruz (Sum)	Vladimir Potiyevskiy
1.33 K	1.33 K	1.32 K	1.32 K	640	396	295	217

Average prompts for users

37.8

Median prompts for users

10

A high count of failures per user can skew the average so consider both the average and the median when analyzing the data. The median represents the 50th percentile, where 50% of the users have prompts equal to or less than the median value.

Prompts by user with additional meta data

Search

TimeGenerated	↑↓	UserDisplayName	↑↓	AppDisplayName	↑↓	AuthStatus	↑↓	OriginalRequestId	↑↓	CountPrompts	↑↓
3/3/2022, 9:19:31 PM		Joey Cruz		Azure Portal		Success		4c527966-d913-47d7-9b37-257f732c0600		15	
3/1/2022, 12:10:45 PM		Adell Evens (Project TNT)		OfficeHome		Success		860d4cac-80f1-46ee-8694-394a877efa00		12	
3/15/2022, 4:49:55 PM		JOKER PARKER		Azure Portal		Success		60ce45e9-5d2d-4917-9094-03896d249e00		12	
3/22/2022, 11:53:59 AM		Vishakha Goel		Microsoft Authenticator App		Success		0c6a0de8-350b-4ab3-a907-0659d3a91a00		11	
3/18/2022, 6:58:44 AM		Joey Cruz		Azure Active Directory PowerShell		Success		bh20e4d2-e127-4f46-aa1e-77b22f391000		10	

Time: Last 30 days ▾ AuthMethod: All ▾ DeviceState: All ▾ AppDisplayName: <unset> ▾ ⓘ UserDisplayName: joey cruz ⓘ ⓘ AuthStatus: All ▾ OS: <unset> ▾ ⓘ

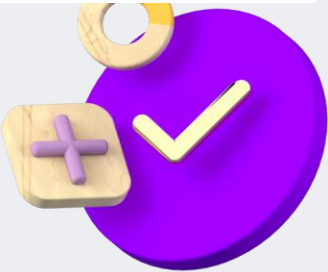
Authentication Prompts Summary



Prompts by user with additional meta data

Search

TimeGenerated	↑↓	UserDisplayName	↑↓	AppDisplayName	↑↓	AuthStatus	↑↓	OriginalRequestId	↑↓	CountPrompts	↑↓
3/3/2022, 9:19:31 PM		Joey Cruz		Azure Portal		Success		4c527966-d913-47d7-9b37-257f732c0600		15	
3/7/2022, 10:36:41 AM		Joey Cruz		Microsoft Azure Active Directory ...		Success		c3925805-7bab-4cbc-961e-31403f564400		10	
3/18/2022, 6:58:44 AM		Joey Cruz		Azure Active Directory PowerShell		Success		bb20e4d2-e127-4f46-aa1e-77b22f391000		10	
3/3/2022, 9:08:41 PM		Joey Cruz		My Apps		Success		cb30c0b9-4318-4aac-a8c9-f04265525000		9	
3/20/2022, 7:31:38 PM		Joey Cruz		Azure Active Directory PowerShell		Success		c4151298-ca9e-4e8c-8b98-fdd766bc0700		8	
3/21/2022, 11:38:49 AM		Joey Cruz (HE/HIM)		My Apps		Success		a59e2797-c362-4776-b7e9-3078c4ba0400		8	



- Overview
- Preview features
- Diagnose and solve problems

Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations
- Identity Governance
- Application proxy
- Custom security attributes (Preview)
- Licenses
- Azure AD Connect
- Custom domain names
- Mobility (MDM and MAM)

- Password reset
- Company branding
- User settings
- Properties
- Security

Monitoring

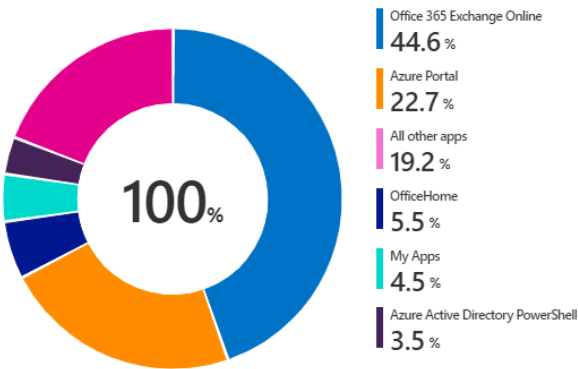
- Sign-in logs
- Audit logs
- Provisioning logs

Workbooks Edit Save Refresh Alerts Pins Smile Help ? Auto refresh: Off

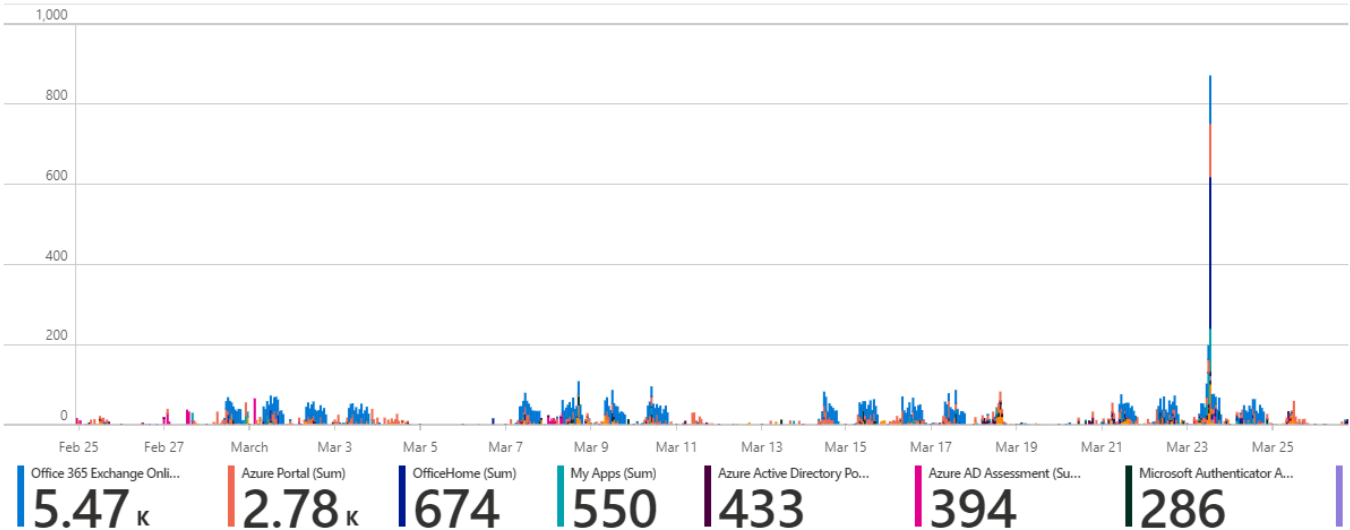
Authentication prompts by application

To investigate the top prompted apps, filter this report by AppID and look at "Authentication policy" graph to understand what triggers multiple authentication requests.

% Prompts by application



Hourly prompts by application



Prompts by application with additional meta data

Search

TimeGenerated	↑↓	Application display name	↑↓	Authentication status	↑↓	OriginalRequestId	↑↓	Count of prompts↑↓
3/1/2022, 12:07:38 PM		OfficeHome		Success		6c96a683-103a-4708-9a18-3a7cca18c700		9
3/23/2022, 10:53:47 PM		OfficeHome		Success		b1a766a6-0ee4-4eae-ae51-f2f358152b00		9
3/22/2022, 10:23:33 AM		Azure Portal		Success		4aa99254-9ec2-49bc-b2db-6e11c9b01500		9
3/22/2022, 8:05:50 AM		Azure Portal		Success		4d2de397-4d08-478b-a676-58ca9f612e00		9
3/1/2022, 3:04:54 AM		Azure AD Assessment		Success		db62e55-ca6c-40dd-ae3c-7a640e17ac00		9
3/1/2022, 11:14:56 PM		Azure Portal		Success		56a95ee0-af28-47c5-a569-1cbd41de1001		9

Results were limited to the first 250 rows.

- Overview
- Preview features
- Diagnose and solve problems

Manage

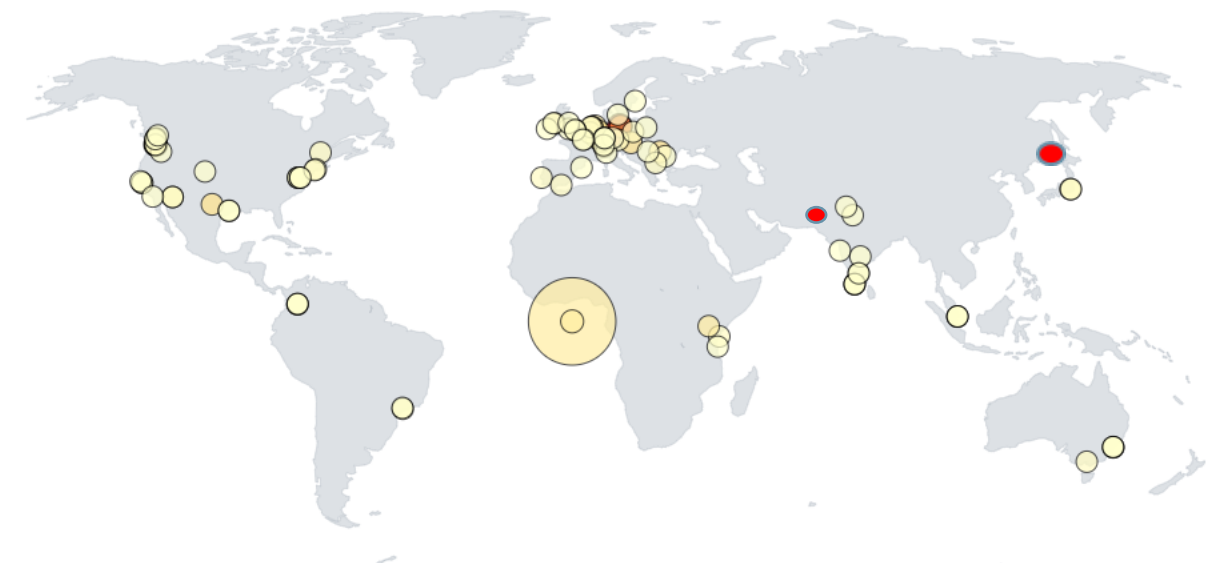
- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations
- Identity Governance
- Application proxy
- Custom security attributes (Preview)
- Licenses
- Azure AD Connect
- Custom domain names
- Mobility (MDM and MAM)
- Password reset
- Company branding
- User settings
- Properties
- Security

Heatmap of Risk Detections

Map location is based on longitude and latitude data. Size of the circles are based on number of risk detections. Color of circles are based on the Weighted Risk = Number of Detections x Weight

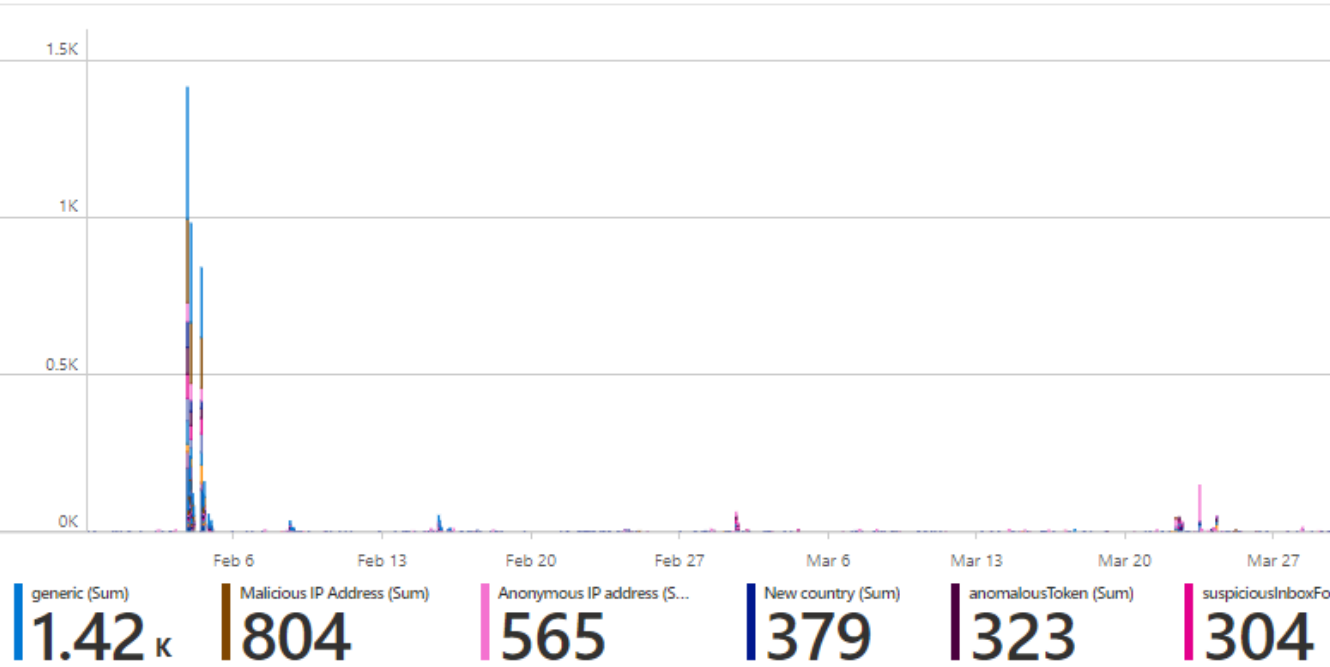
Weight: High Risk = 10, Medium Risk = 5, Low Risk = 1

Lower risk circles are yellow and higher risk circles are red.

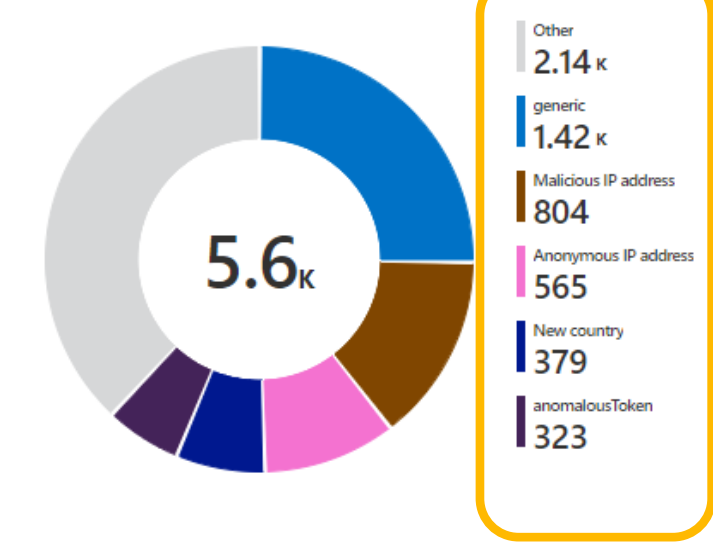


Unknown	Other	Schoenwalde-Glien	Hassfurt	Seattle	Dronen	Unknown	Berlin	Wieden	Duesseldorf	Victoria	Huenenberg	Strasbourg	Seattle	Bucuresti	Seattle	Luxembourg	Amsterdam	Karlsruhe
4.68k	128	118	114	83	57	57	37	30	25	20	20	18	15	14	14	14	14	14

Hourly Risk Detections Event Type



Total Count by Risk Event Type



Count of Risk Detections by User

UserDisplayName	↑↓	TotalCount↑↓	HighRiskCount↑↓	MediumRiskCount↑↓	LowRiskCount↑↓	NoR
Adell Evens (Project TNT)		553	553	0	0	
Joey Cruz		172	129	36	7	
Adell Evens		23	23	0	0	
Aritra Nirmal		16	16	0	0	
Joey Cruz (HE/HIM)		37	15	21	1	
Michael Wakahe		12	9	3	0	
Lanbo Fang		16	8	8	0	
Yugansh Arora		11	8	3	0	

Risk Detections by IP Address

IpAddress	↑↓	TotalCount↑↓	HighRiskCount↑↓
216.243.35.38		232	231
fde4:8dba:2600:6d0a:6c26:100:ac10:4205		100	100
109.70.100.83		58	58
83.97.20.189		50	50
185.220.100.252		50	50
185.220.101.136		42	42
89.163.252.230		42	42
185.220.100.250		25	20

Cross-tenant inbound and outbound access activity for Azure AD tenants

Workspace: All Guide: Off

Tenant administrators who are making changes to policies governing cross-tenant access can use this workbook to visualize and review existing access activity patterns before making policy changes.

Time range: Last 90 days External Tenant ID: All external tenants User principal name: All users Application: All applications Status: All

Number of external tenants with cross-tenant access activity

46

Search

External Tenant	↑↓	Outbound Sign-Ins...↑↓	Outbound Sign-In ...↑↓	Outbound Users	↑↓	Outbound Apps	↑↓	Inbound Sign-Ins T...↑↓	Inbound Sign-In Fa...↑↓	Inbound Users	↑↓	Inbound Apps	↑↓
72f988bf-86f1-41af-91ab-2d7cd011db47	839	372	161	15	810	512	6	13					
bdf3e0e4-70dd-43d1-88d0-5851cc79c3a2	62	47	1	2	73	10	1	3					
0817c655-a853-4d8f-9723-3a333b5b9235	248	35	1	25	0	0	0	0					
0e238151-6e74-4e33-89dd-1e4dd82358c1	22	19	2	3	0	0	0	0					
556b80b7-c9fc-41fd-92da-c3635f7918e5	37	17	3	3	0	0	0	0					
8c3af30a-0c63-43b4-8b5b-9888693e053f	37	11	2	3	5	3	2	3					
cc7d0b33-84c6-4368-a879-2e47139b7b1f	23	9	1	5	0	0	0	0					
b4c546a4-7dac-46a6-a7dd-ed822a11efd3	8	8	4	1	0	0	0	0					
a19f121d-81e1-4858-a9d8-736e267fd4c7	4	4	1	1	0	0	0	0					
8b8e4098-d482-4b93-aef1-872652f91559	14	4	1	3	0	0	0	0					
ffa26b7c-9f70-4518-ac4b-4a2bdc53404b	36	3	2	3	1	1	1	1					

Outbound activity Inbound activity

Inbound activity details for selected external tenant

Inbound sign-in status summary for selected external tenant

Search

Sign-In Status Count↑↓	Status	↑↓	Sign-In Status Code	↑↓	Sign-In Status Reason	↑↓
403	✓ Success		0		Success	
327	✗ Failure		50158		External security challenge was not satisfied.	
55	✗ Failure		50074		User did not pass the MFA challenge.	
11	✗ Failure		50072		Users' needs to enroll for second factor authentication (interactive).	
6	✗ Failure		530004		Other	
3	✗ Failure		50011		The reply address is missing, misconfigured, or does not match reply addresses configured for the application. Try out the resolution listed at <a href="https://docs.microsoft.com/en-us/azure/active-directory/develop/msal-authentication-errors">https://docs.microsoft.com/en-us/azure/active-directory/develop/msal-authentication-errors</a> .	
3	✗ Failure		65001		Application X doesn't have permission to access application Y or the permission has been revoked. Or The user or administrator has not consented to use the ap...	
3	✗ Failure		50203		Other	
1	✗ Failure		50140		This error occurred due to 'Keep me signed in' interrupt when the user was signing-in.	



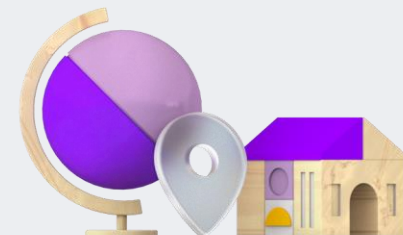
Outbound application activity by selected tenant

Search									
External Resource App Id	↑↓	External Resource App Display Name	↑↓	External Client App Id	↑↓	External Client App Display Name	↑↓	Outbound...	Outbound...
00000003-0000-0ff1-ce00-000000000000		Office 365 SharePoint Online		00000003-0000-0ff1-ce00-000000000000		Office 365 SharePoint Online		98	356
797f4846-ba00-4fd7-ba43-dac1f8f63013		Windows Azure Service Management API		499b84ac-1321-427f-aa17-267ca6975798		Azure DevOps		24	83
797f4846-ba00-4fd7-ba43-dac1f8f63013		Windows Azure Service Management API		c44b4083-3bb0-49c1-b47d-974e53cbdf3c		Azure Portal		5	37
00000002-0000-0000-c000-000000000000		Windows Azure Active Directory		ed563f8f-05dd-438b-9dce-61c5d2859cc4		spmanage-prod		4	26
00000002-0000-0000-c000-000000000000		Windows Azure Active Directory		5572c4c0-d078-44ce-b81c-6cbf8d3ed39e		Vortex [wsfed enabled]		3	25
00000003-0000-0ff1-ce00-000000000000		Office 365 SharePoint Online		08e18876-6177-487e-b8b5-cf950c1e598c		SharePoint Online Web Client Extensibility		1	11
00000003-0000-0000-c000-000000000000		Microsoft Graph		810dcf14-1858-4bf2-8134-4c369fa3235b		Azure AD Identity Governance - Entitlement Management		3	10
00000003-0000-0000-c000-000000000000		Microsoft Graph		8c59ead7-d703-4a27-9e55-c96a0054c8d2		My Profile		1	9
00000002-0000-0000-c000-000000000000		Windows Azure Active Directory		cd88b695-920c-4898-a6bb-36b3908a479c		IdentityIQ		5	8
00000003-0000-0000-c000-000000000000		Microsoft Graph		2793995e-0a7d-40d7-bd35-6968ba142197		My Apps		3	7
cc15fd57-2c6c-4117-a88c-83b1d56b4bbe		Microsoft Teams Services		1fec8e78-bce4-4aaf-ab1b-5451cc387264		Microsoft Teams		1	7
00000003-0000-0000-c000-000000000000		Windows Azure Active Directory		43d741d1-efb1-460a-b4da-f53e433e0644		...		2	7

Outbound user activity by selected application

Search			
User ID	↑↓	Outbound App Sign-I...	Outbound App Sign-i...
joeyc@woodgrove.ms		78	14
zegon@woodgrove.ms		32	3
vmahtani@woodgrove.ms		27	0
vnotivevskiy@woodgrove.ms		26	14

# Lessons learned in Log Analytics and KQL



# Percentages

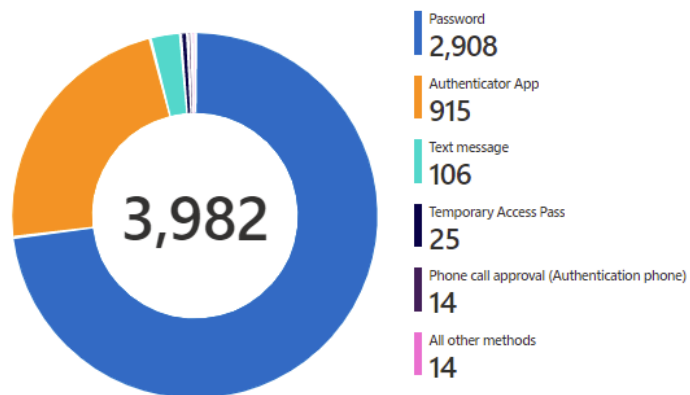


# Laying out your workbook

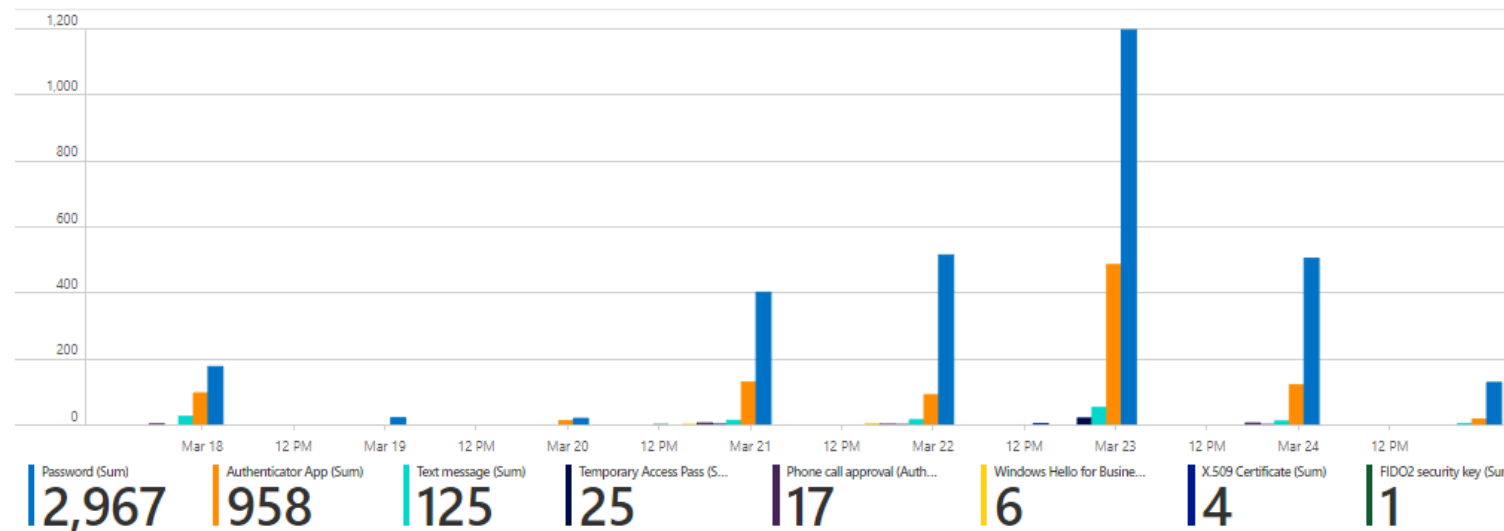
## Authentication prompts by authentication method

To investigate methods causing the most prompts, filter this report by AuthMethod.

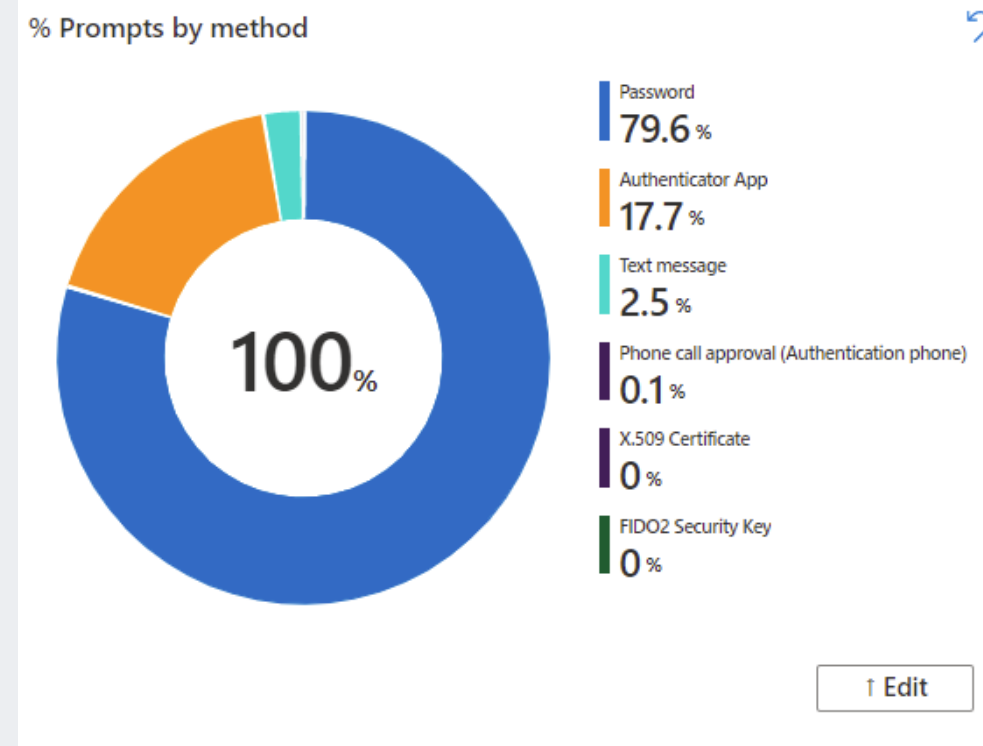
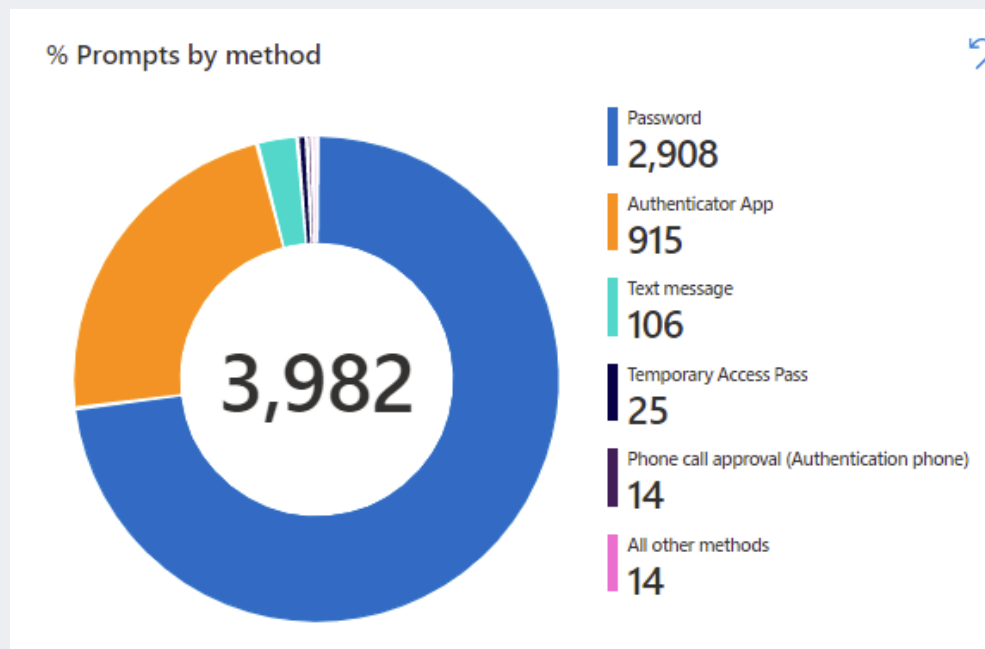
% Prompts by method



Daily prompts by method



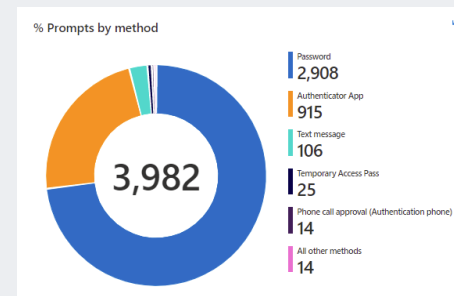
## Converting count to percentages



# Base query

SigninLogs

```
|mv-expand ParsedFields=parse_json(AuthenticationDetails) //parse out Authentication Details column and put in new column called "ParsedFields"
    |extend AuthenticationMethod = ParsedFields.authenticationMethod //create new column by extracting the authenticationMethod value out of the ParsedFields column
    |extend AuthMethod = toString(AuthenticationMethod) //convert to string format
    |extend ParsedFields2=parse_json(DeviceDetail) //parse out Device Details column and put in column called "ParsedFields2"
    |extend DeviceState = case(DeviceDetail["trustType"] == "", "Unmanaged", DeviceDetail["trustType"]) //convert to string
|extend OperatingSystem = ParsedFields2.operatingSystem //extract the operatingSystem value out of the ParsedFields2 column
|extend OS = toString(OperatingSystem) //convert to string format
|where AuthMethod != "Previously satisfied" and AuthMethod != "" //remove Previously Satisfied and blank AuthMethods from the query
|where UserDisplayName != "On-Premises Directory Synchronization Service Account" //remove this UserDisplayName
|where AuthMethod in ({AuthMethod}) or '*' in ({AuthMethod}) //AuthMethod parameter
|where DeviceState in ({DeviceState}) or '*' in ({DeviceState}) //DeviceState parameter
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}" //User parameter
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}" //App parameter
|where "{OS:escape}" == "All OS" or OS contains "{OS:escape}" //OS parameter
|extend Status = ParsedFields.succeeded //extract the succeeded value out of the ParsedFields2 column
|extend AuthStatus = case(Status == "true", "Success", "Failure") //label status as Success or Failure
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus}) //AuthStatus parameter
|summarize Count = count() by AuthMethod //summarize operator aggregates data and produces count by AuthMethod
```

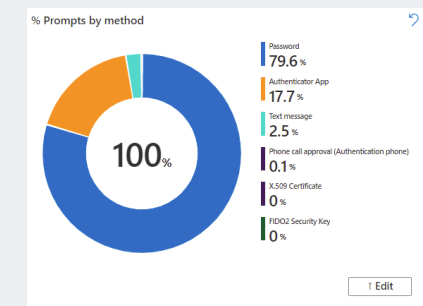


# Percentage

```

let baseQuery = //store a dataset in baseQuery
materialize(SigninLogs //materialize creates a subquery that other queries can reference
|mv-expand ParsedFields=parse_json(AuthenticationDetails) //parse out Authentication Details column and put in new column called
"ParsedFields"
    |extend AuthenticationMethod = ParsedFields.authenticationMethod //create new column by extracting the authenticationMethod value
out of the ParsedFields column
    |extend AuthMethod = toString(AuthenticationMethod) //convert to string format
    |extend ParsedFields2=parse_json(DeviceDetail) //parse out Device Details column and put in column called "ParsedFields2"
    |extend DeviceState = case(DeviceDetail["trustType"] == "", "Unmanaged", DeviceDetail["trustType"]) //convert to string
|extend OperatingSystem = ParsedFields2.operatingSystem //extract the operatingSystem value out of the ParsedFields2 column
|extend OS = toString(OperatingSystem) //convert to string format
|where AuthMethod != "Previously satisfied" and AuthMethod != "" //remove Previously Satisfied and blank AuthMethods from the query
|where UserDisplayName != "On-Premises Directory Synchronization Service Account" //remove this UserDisplayName
|where AuthMethod in ({AuthMethod}) or '*' in ({AuthMethod}) //AuthMethod parameter
|where DeviceState in ({DeviceState}) or '*' in ({DeviceState}) //DeviceState parameter
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}" //User parameter
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}" //App parameter
|where "{OS:escape}" == "All OS" or OS contains "{OS:escape}" //OS parameter
|extend Status = ParsedFields.succeeded //extract the succeeded value out of the ParsedFields2 column
|extend AuthStatus = case(Status == "true", "Success", "Failure") //label status as Success or Failure
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus}) //AuthStatus parameter
);
let totalCount = toscalar(baseQuery | count); //return a scalar constant value of the evaluated expression
baseQuery //reference the baseQuery
|summarize Count = count(AuthMethod) * 100.0 /totalCount by AuthMethod //aggregate count, divide by total count

```

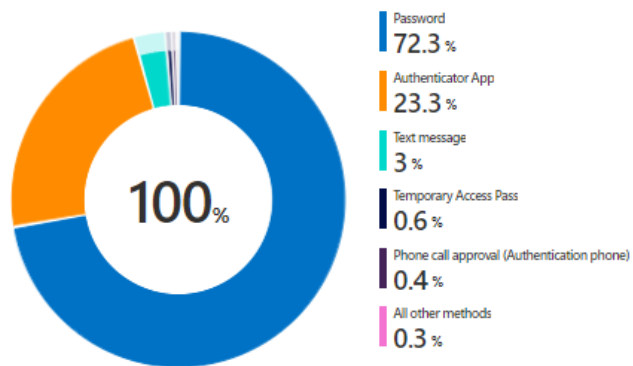


# Converting count to percentages

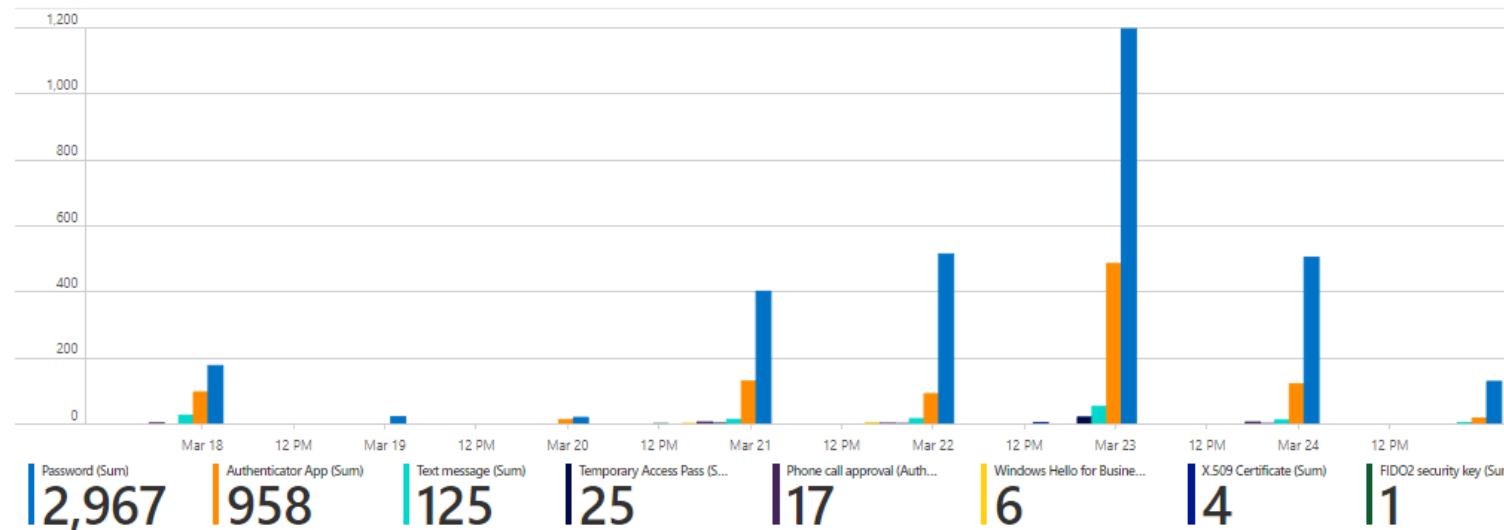
## Authentication prompts by authentication method

To investigate methods causing the most prompts, filter this report by AuthMethod.

% Prompts by method



Daily prompts by method

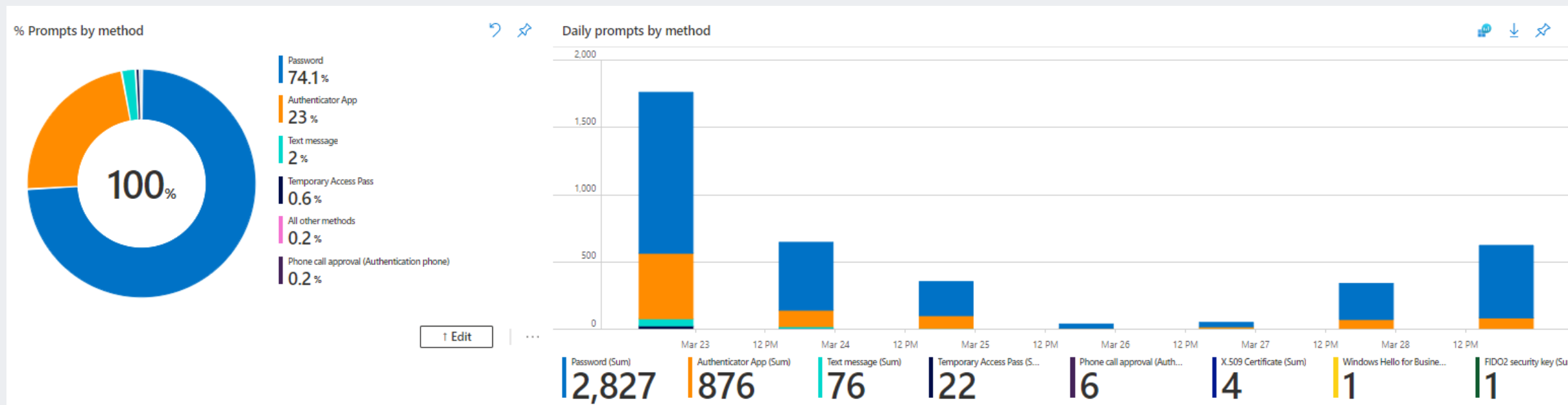




# Interactive Graphs

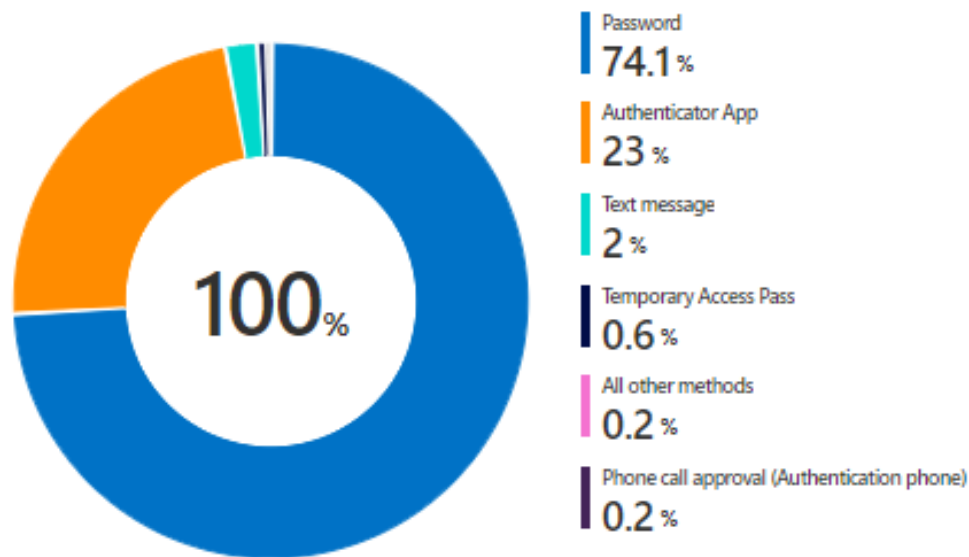


# Interactive Graphs



# Interactive graphs

% Prompts by method



Settings
Advanced Settings
Style
Advanced Editor

Step name ⓘ  
query - 1

☐ Make this item conditionally visible ⓘ  
☒ Always show the pin icon on this step ⓘ  
☒ When items are selected, export parameters ⓘ  
☐ Allow selection of multiple values ⓘ

All fields → SelectedAuthMethod × + Add parameter

☐ Show query when not editing  
☐ Show open external query button when not editing  
☐ Show refresh icon when not editing  
☐ Show Export to Excel button when not editing

Columns to Export  
Visible Columns All Columns

Chart title ⓘ  
% Prompts by method

No data message ⓘ  
The query returned no results.

No data message style ⓘ  
Info

☒ Done Editing ☐ Cancel +

When items are selected, export parameters ⓘ  
☐ Allow selection of multiple values ⓘ  
All fields → SelectedAuthMethod × + Add parameter

Exported Parameter Settings

Field to export ⓘ  
field name

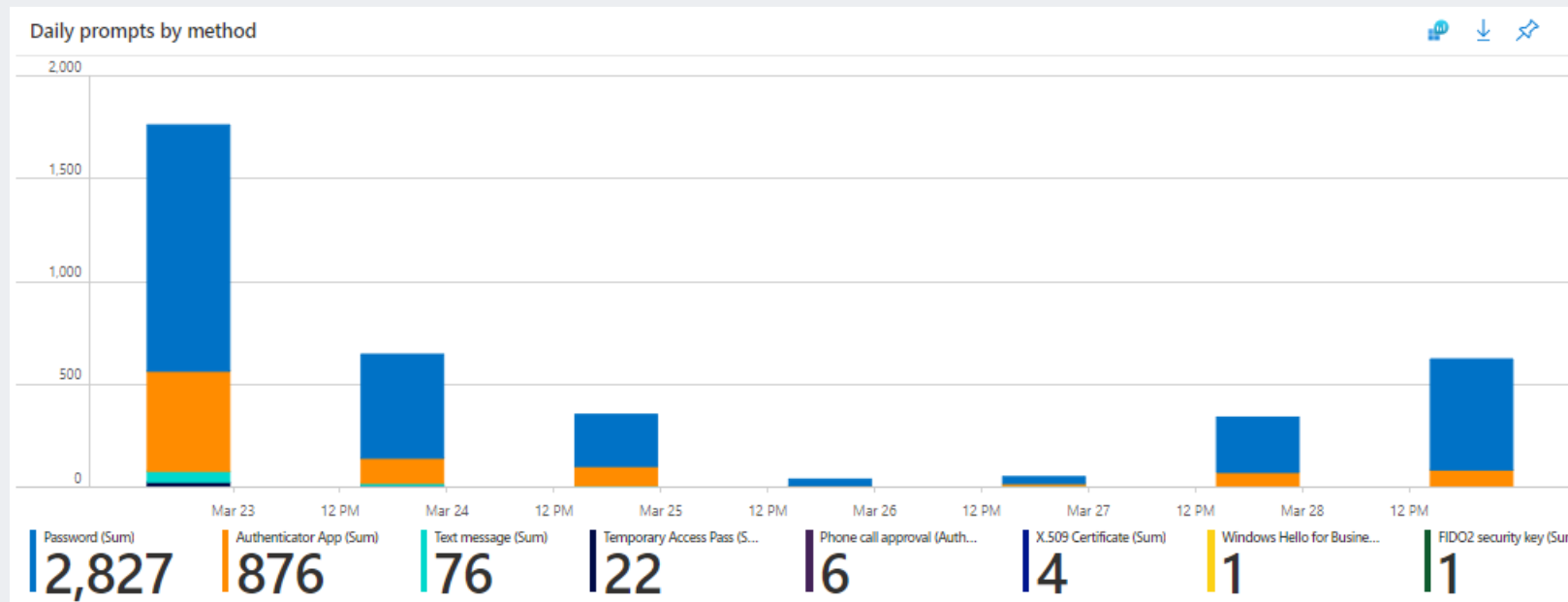
Parameter name ⓘ  
SelectedAuthMethod

Parameter type ⓘ  
▼

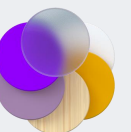
Default value ⓘ  
{"series": "All"}

Save Cancel

## Interactive graphs



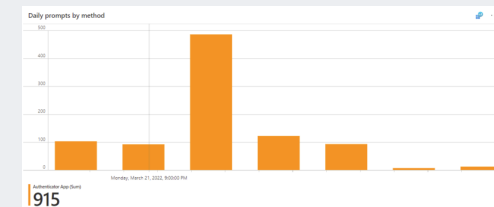
```
|mv-expand ParsedMethod = parse_json(tostring('{SelectedAuthMethod}'))
|extend SelectedAuthMethod = ParsedMethod.series
|where SelectedAuthMethod == 'All' or (AuthMethod == SelectedAuthMethod)
```



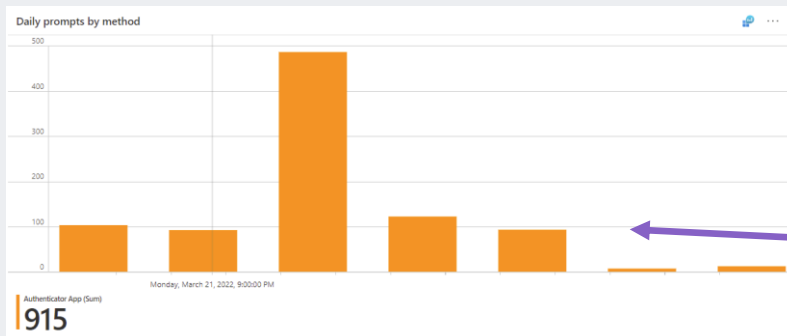
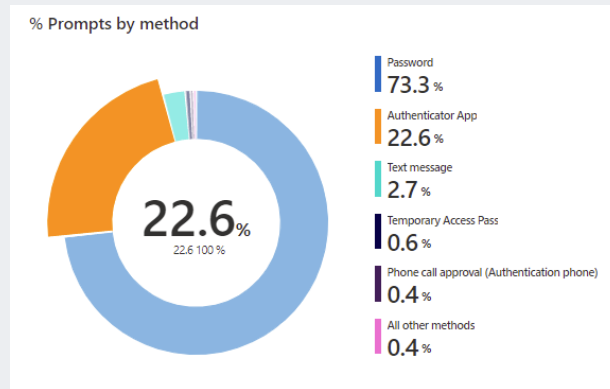
# Interactive graphs

SigninLogs

```
|mv-expand ParsedFields=parse_json(AuthenticationDetails) //parse out Authentication Details column and put in new column called "ParsedFields"
    |extend AuthenticationMethod = ParsedFields.authenticationMethod //create new column by extracting the authenticationMethod value out of the ParsedFields column
    |extend AuthMethod = toString(AuthenticationMethod) //convert to string format
|mv-expand ParsedMethod = parse_json(toString('{SelectedAuthMethod}')) //parse out SelectedAuthMethod
    |extend SelectedAuthMethod = ParsedMethod.series //create new column with by extracting series value
    |where SelectedAuthMethod == 'All' or (AuthMethod == SelectedAuthMethod) //parameters for SelectedAuthMethod
|extend ParsedFields2=parse_json(DeviceDetail) //parse out Device Details column and put in column called "ParsedFields2"
|extend DeviceState = case(DeviceDetail["trustType"] == "", "Unmanaged", DeviceDetail["trustType"]) //convert to string
|extend OperatingSystem = ParsedFields2.operatingSystem //extract the operatingSystem value out of the ParsedFields2 column
|extend OS = toString(OperatingSystem) //convert to string format
|where AuthMethod != "Previously satisfied" and AuthMethod != "" //remove Previously Satisfied and blank AuthMethods from the query
|where UserDisplayName != "On-Premises Directory Synchronization Service Account" //remove this UserDisplayName
|where AuthMethod in ({AuthMethod}) or '*' in ({AuthMethod}) //AuthMethod parameter
|where DeviceState in ({DeviceState}) or '*' in ({DeviceState}) //DeviceState parameter
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}" //User parameter
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}" //App parameter
|where "{OS:escape}" == "All OS" or OS contains "{OS:escape}" //OS parameter
|extend Status = ParsedFields.succeeded //extract the succeeded valueout of the ParsedFields2 column
|extend AuthStatus = case(Status == "true", "Success", "Failure") //label status as Success or Failure
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus}) //AuthStatus parameter
|summarize AuthMethCount = count() by bin (TimeGenerated, 1d), AuthMethod //aggregate to count the daily sum
By AuthMethod
```



# Interactive Graphs



Settings
Advanced Settings
Style
Advanced Editor

Run Query
Samples
woodgrove-loganalytic...
Time Range
Time
Visualization
Grid
Size
Medium
Column Settings

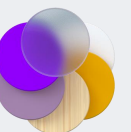
Log Analytics workspace Logs Query

```

|where SelectedAuthMethod == 'All' or (AuthMethod == SelectedAuthMethod)
|where AuthMethod != ''
|extend ParsedFields2=parse_json(DeviceDetail)
|extend DeviceState = case(DeviceDetail["trustType"] == "", "Unmanaged", DeviceDetail["trustType"])
|extend OperatingSystem = ParsedFields2.operatingSystem
|extend OS = tostring(OperatingSystem)
|where AuthMethod != "Previously satisfied"
|where UserDisplayName != "On-Premises Directory Synchronization Service Account"
|extend Status = ParsedFields.succeeded
|extend AuthStatus = case(Status == "true", "Success", "Failure")
|where AuthStatus in ((AuthStatus)) or '*' in ((AuthStatus))
    
```

Daily prompts by method

TimeGenerated	AuthMethod	SelectedAuthMethod	AuthMethCount
3/27/2022, 7:00:00 PM	Mobile app notification	Mobile app notification	13
3/20/2022, 7:00:00 PM	Mobile app notification	Mobile app notification	59
3/26/2022, 7:00:00 PM	Mobile app notification	Mobile app notification	12
3/25/2022, 7:00:00 PM	Mobile app notification	Mobile app notification	7
3/23/2022, 7:00:00 PM	Mobile app notification	Mobile app notification	122
3/24/2022, 7:00:00 PM	Mobile app notification	Mobile app notification	93
3/22/2022, 7:00:00 PM	Mobile app notification	Mobile app notification	486
3/21/2022, 7:00:00 PM	Mobile app notification	Mobile app notification	92



# Interactive Graphs

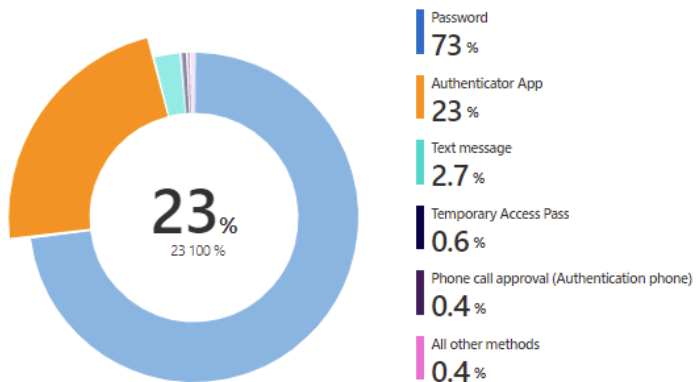
## Authentication prompts by authentication method

To investigate methods causing the most prompts, filter this report by AuthMethod.

[↑ Edit](#)

...

% Prompts by method

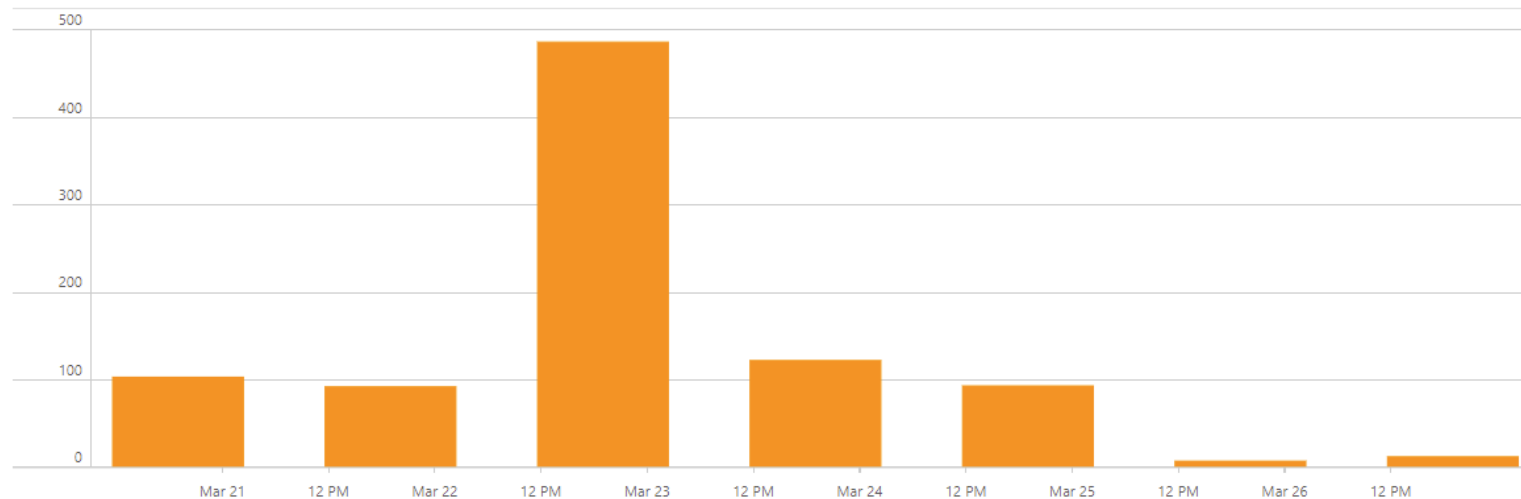

[↑ Edit](#)

...

Daily prompts by method



...



Authenticator App (Sum)

915



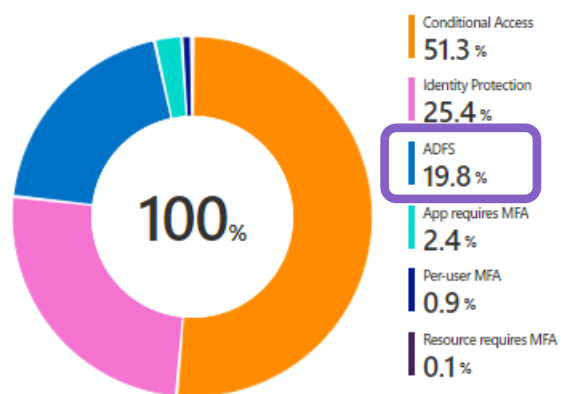
# Union



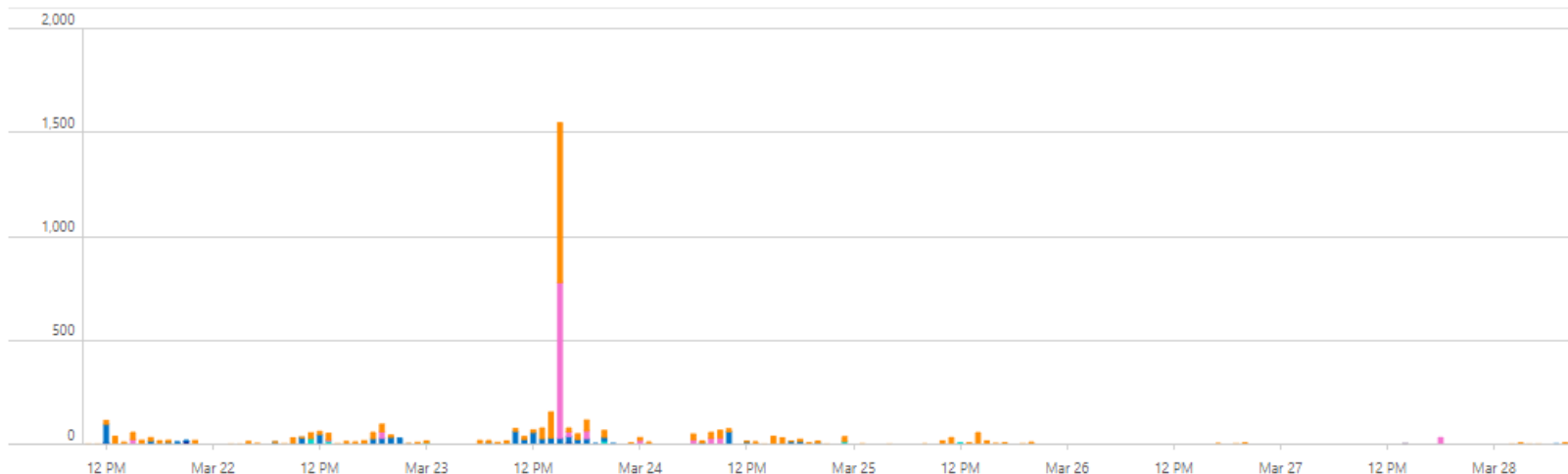


# Union of two data sets

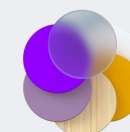
Prompts by authentication policy



Hourly prompts by authentication policy



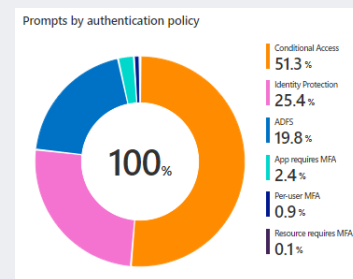
1 Edit



# Union of two data sets: the pie chart

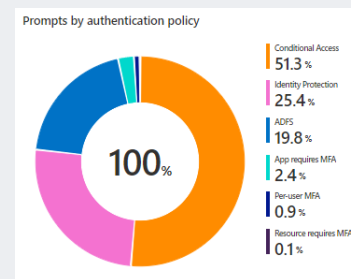
```
let policyData = SigninLogs
|mv-expand ParsedFields=parse_json(AuthenticationDetails)
|extend AuthenticationMethod = ParsedFields.authenticationMethod
|extend AuthMethod = toString(AuthenticationMethod)
|extend ParsedFields2=parse_json(DeviceDetail)
|extend DeviceState = case(DeviceDetail["trustType"] == "", "Unmanaged"
, DeviceDetail["trustType"])
|extend OperatingSystem = ParsedFields2.operatingSystem
|extend OS = toString(OperatingSystem)
|mv-
expand ParsedFields3 = parse_json(AuthenticationRequirementPolicies)
|extend AuthReqPolicy = ParsedFields3.detail
|extend AuthPolicy = toString(AuthReqPolicy)
|where AuthMethod != "Previously satisfied" and AuthMethod != ""
|where UserDisplayName != "On-
Premises Directory Synchronization Service Account"
|where AuthMethod in ({AuthMethod}) or '*' in ({AuthMethod})
|where DeviceState in ({DeviceState}) or '*' in ({DeviceState})
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}"
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}"
|where "{OS:escape}" == "All OS" or OS contains "{OS:escape}"
|extend Status = ParsedFields.succeeded
|extend AuthStatus = case(Status == "true", "Success", "Failure")
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus})
|project AuthPolicy, UserDisplayName, AppDisplayName;
```

```
let adfsData = ADFSSignInLogs
|mv-expand PF=parse_json(AuthenticationDetails)
|extend AuthDetail = PF.authenticationMethodDetail
|extend AuthPolicy = toString(AuthDetail)
|where UserDisplayName != "On-Premises Directory Synchronization Service Account"
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}"
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}"
|extend Status = PF.succeeded
|extend AuthStatus = case(Status == "true", "Success", "Failure")
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus})
|project AuthPolicy, AuthStatus, AppDisplayName, UserDisplayName;
let basequery = materialize(
union policyData, adfsData
|summarize totalCount = count() by AuthPolicy
);
let total = toscalar(
basequery
| summarize sum(totalCount)
);
basequery
| extend pct = totalCount*100.0/total
```



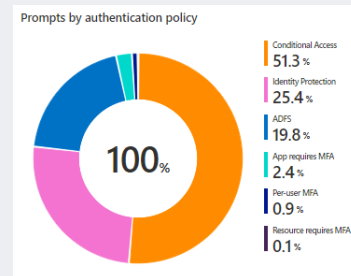
## Break it down: First part of the query

```
let policyData = SigninLogs
|mv-expand ParsedFields=parse_json(AuthenticationDetails) //parse out Authentication Details column and put in new column called "ParsedFields"
    |extend AuthenticationMethod = ParsedFields.authenticationMethod //create new column by extracting the authenticationMethod value out of the ParsedFields column
    |extend AuthMethod = toString(AuthenticationMethod) //convert to string format
    |extend ParsedFields2=parse_json(DeviceDetail) //parse out Device Details column and put in column called "ParsedFields2"
    |extend DeviceState = case(DeviceDetail["trustType"] == "", "Unmanaged", DeviceDetail["trustType"]) //convert to string
|extend OperatingSystem = ParsedFields2.operatingSystem //extract the operatingSystem value out of the ParsedFields2 column
|extend OS = toString(OperatingSystem) //convert to string format
|mv-expand ParsedFields3 = parse_json(AuthenticationRequirementPolicies)
    |extend AuthReqPolicy = ParsedFields3.detail
    |extend AuthPolicy = toString(AuthReqPolicy)
|where AuthMethod != "Previously satisfied" and AuthMethod != "" //remove Previously Satisfied and blank AuthMethods from the query
|where UserDisplayName != "On-Premises Directory Synchronization Service Account" //remove this UserDisplayName
|where AuthMethod in ({AuthMethod}) or '*' in ({AuthMethod}) //AuthMethod parameter
|where DeviceState in ({DeviceState}) or '*' in ({DeviceState}) //DeviceState parameter
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}" //User parameter
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}" //App parameter
|where "{OS:escape}" == "All OS" or OS contains "{OS:escape}" //OS parameter
|extend Status = ParsedFields.succeeded //extract the succeeded value out of the ParsedFields2 column
|extend AuthStatus = case(Status == "true", "Success", "Failure") //label status as Success or Failure
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus}) //AuthStatus parameter
|project TimeGenerated, AuthPolicy, UserDisplayName, AppDisplayName; //Select columns
```



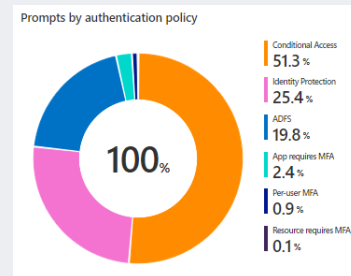
## Break it down: Second part of the query

```
let adfsData = ADFSSignInLogs //store a dataset in adfsData
|mv-expand PF=parse_json(AuthenticationDetails) //parse out AuthenticationDetails column and put in new column called "PF"
|extend AuthDetail = PF.authenticationMethodDetail //create new column by extracting the authenticationMethodDetail value out of the PF column
|extend AuthPolicy = tostring(AuthDetail) //convert to string format
|where UserDisplayName != "On-Premises Directory Synchronization Service Account"
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}" //User parameter
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}" //App parameter
|extend Status = PF.succeeded //extract status from the succeeded value in the PF column
|extend AuthStatus = case(Status == "true", "Success", "Failure") // if true, return "Success", if not, return Failure
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus}) //AuthStatus parameter
|project TimeGenerated,AuthPolicy, AuthStatus, AppDisplayName, UserDisplayName; //Select columns
```



## Break it down: Third part of the query

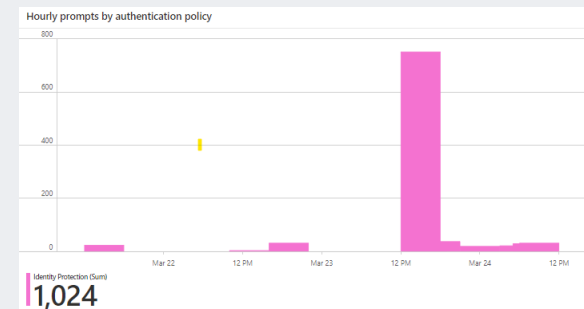
```
let baseQuery = materialize( //store a dataset in baseQuery
union policyData, adfsData //bring together the first and second query
| summarize totalCount = count() by AuthPolicy); //aggregate count by Auth Policy and close the let statement
let total = //calculate total of baseQuery
toscalar(baseQuery //toscalar will return a constant value of the evaluated expression
| summarize sum(totalCount)); //the evaluated expression totalCount by Auth Policy and close the let statement
baseQuery
| extend pct = totalCount*100.0/total //create new column showing percentage of totalCount
```



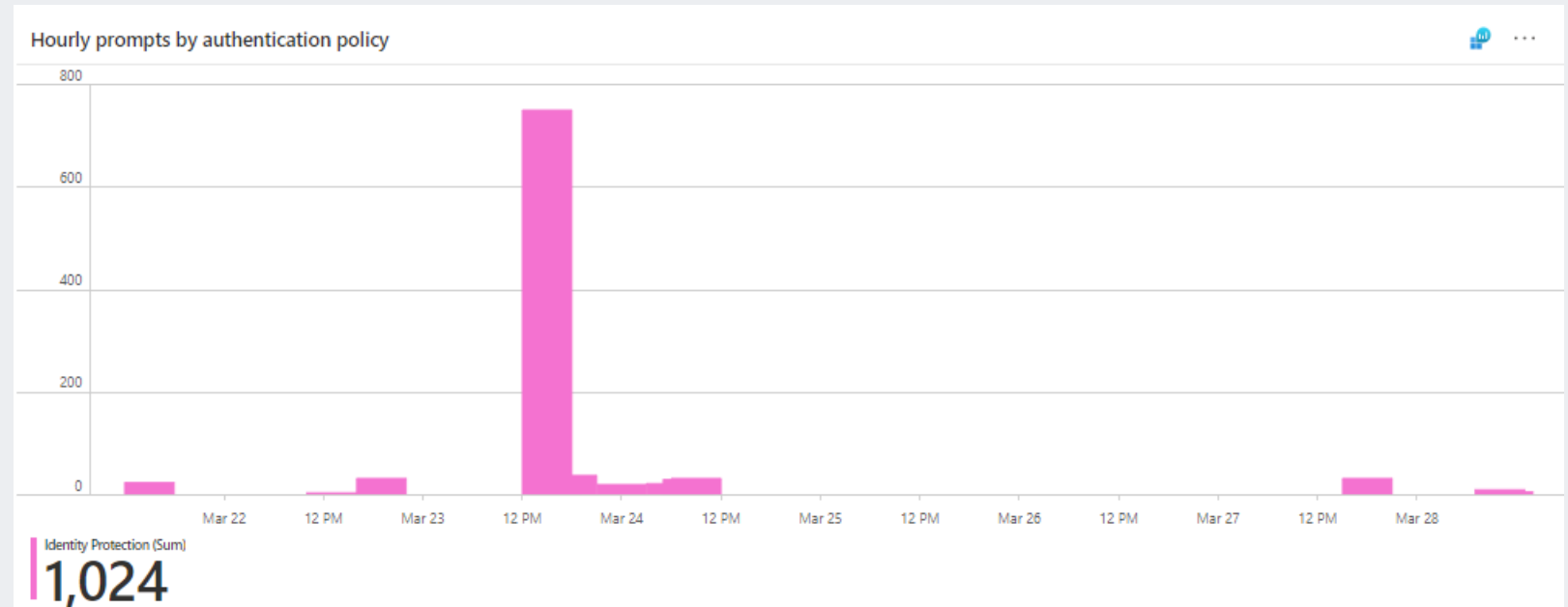
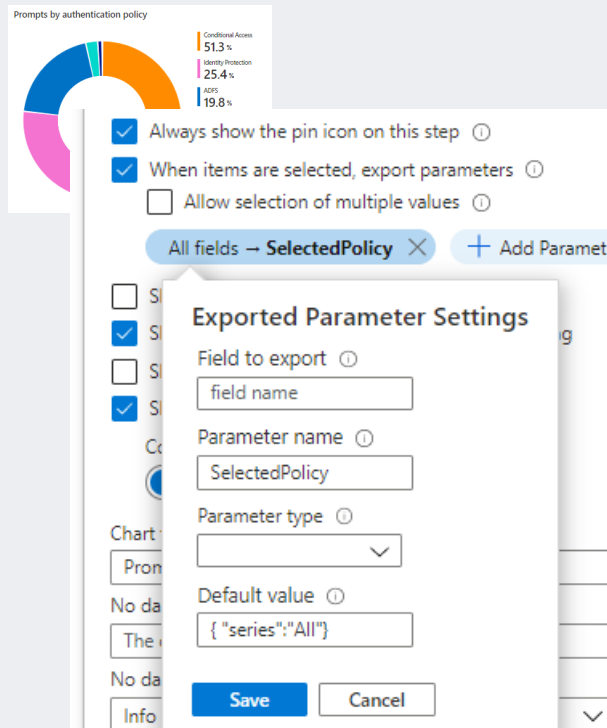
# Union of two data sets: the bar chart

```
let policyData = SigninLogs
|mv-expand ParsedFields=parse_json(AuthenticationDetails)
|extend AuthenticationMethod = ParsedFields.authenticationMethod
|extend AuthMethod = toString(AuthenticationMethod)
|extend ParsedFields2=parse_json(DeviceDetail)
|extend DeviceState = case(DeviceDetail["trustType"] == "", "Unmanaged", DeviceDetail["trustType"])
|extend OperatingSystem = ParsedFields2.operatingSystem
|extend OS = toString(OperatingSystem)
|mv-expand ParsedFields3 = parse_json(AuthenticationRequirementPolicies)
|extend AuthReqPolicy = ParsedFields3.detail
|extend AuthPolicy = toString(AuthReqPolicy)
|mv-expand ParsedPolicy = parse_json(tostring('{SelectedPolicy}'))
|extend SelectedPolicy = ParsedPolicy.series
|where SelectedPolicy == 'All' or (AuthPolicy == SelectedPolicy)
|where AuthMethod != "Previously satisfied" and AuthMethod != ""
|where UserDisplayName != "On-Premises Directory Synchronization Service Account"
|where AuthMethod in ({AuthMethod}) or '*' in ({AuthMethod})
|where DeviceState in ({DeviceState}) or '*' in ({DeviceState})
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}"
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}"
|where "{OS:escape}" == "All OS" or OS contains "{OS:escape}"
|extend Status = ParsedFields.succeeded
|extend AuthStatus = case(Status == "true", "Success", "Failure")
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus})
|project TimeGenerated, AuthPolicy, UserDisplayName, AppDisplayName;
```

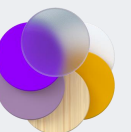
```
let adfsData = ADFSSignInLogs
|mv-expand PF=parse_json(AuthenticationDetails)
|extend AuthDetail = PF.authenticationMethodDetail
|extend AuthPolicy = toString(AuthDetail)
|mv-expand ParsedPolicy = parse_json(tostring('{SelectedPolicy}'))
|extend SelectedPolicy = ParsedPolicy.series
|where SelectedPolicy == 'All' or (AuthPolicy == SelectedPolicy)
|where UserDisplayName != "On-Premises Directory Synchronization Service Account"
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}"
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}"
|extend Status = PF.succeeded
|extend AuthStatus = case(Status == "true", "Success", "Failure")
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus})
|project TimeGenerated, AuthPolicy, AuthStatus, AppDisplayName, UserDisplayName;
union policyData, adfsData
|summarize Count = count() by bin(TimeGenerated,1h),AuthPolicy
```



## The bar chart is still interactive...

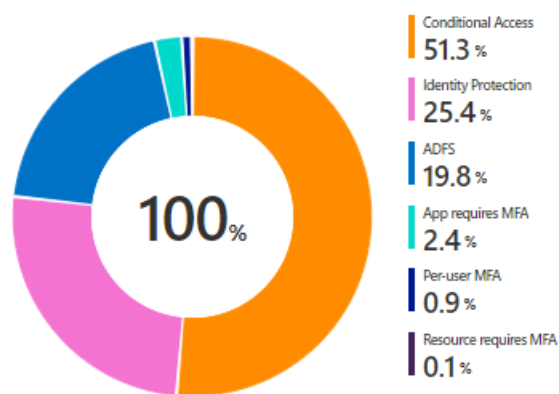


```
| mv-expand ParsedPolicy = parse_json(tostring('{SelectedPolicy}'))
| extend SelectedPolicy = ParsedPolicy.series
| where SelectedPolicy == 'All' or (AuthPolicy == SelectedPolicy)
```

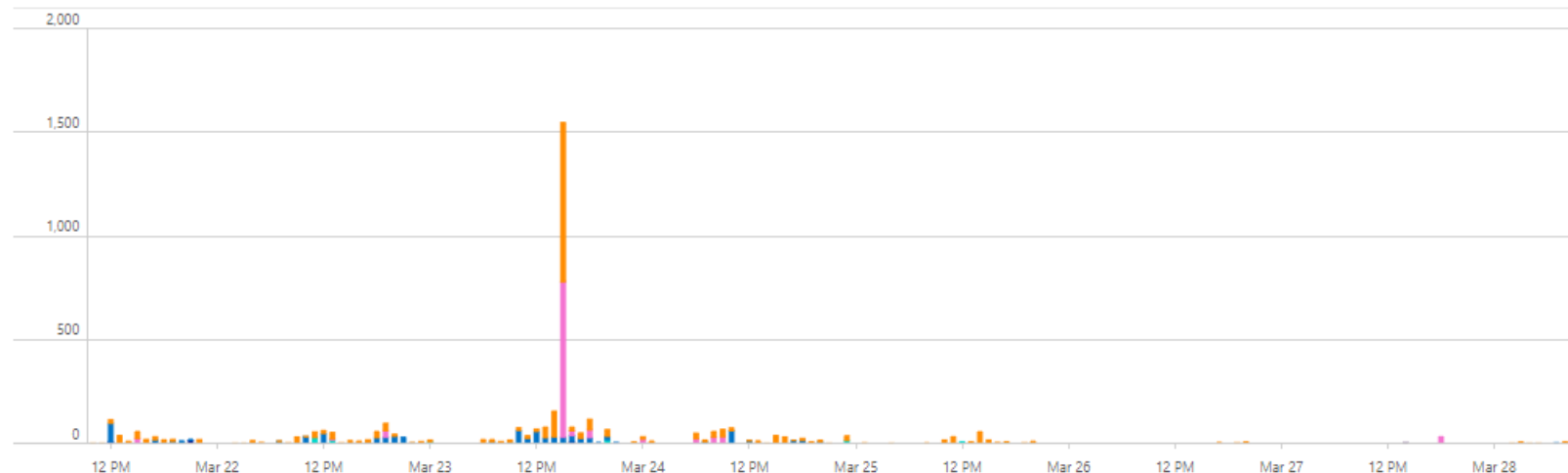


# Union of two data sets

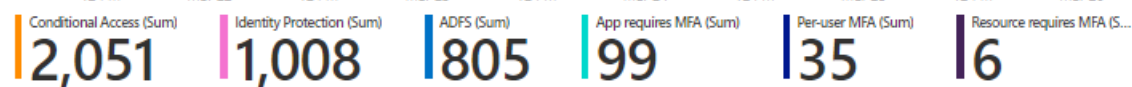
Prompts by authentication policy



Hourly prompts by authentication policy



1 Edit





# Recommendations



# Recommendations

## Managed devices

For improved user experience, we recommend enabling single sign-on using managed devices.

Learn more: [Plan your Azure Active Directory device deployment](#)

% | Count of Managed Devices

0% 10

## Windows Hello for Business

To reduce authentication prompts and improve user experience on Windows 10 and above, we recommend using Windows Hello for Business.

Learn more: [Windows Hello for Business Overview - Microsoft 365 Security](#)

% WHFB | Count of Windows Devices

0.2% 2,825

## Mobile Authentications

The Authenticator app serves as a token broker and can reduce authentication prompts on Android and iOS.

For improved user experience and reduce prompts on mobile devices, we recommend using the Microsoft Authenticator.

Learn more: [Microsoft Authenticator app authentication method - Azure Active Directory](#)

% Auth App Authentications | Count of Android/iOS Devices

88% 3,887

## Mac OS Authentications

On MacOs we recommend the Microsoft Enterprise SSO plug-in for Apple devices.

Learn more: <http://aka.ms/AADAppleSSO>

% | Count of Mac Authentications

1.6% 76

## Recommendations: Mobile Authentications

### Mobile Authentications

The Authenticator app serves as a token broker and can reduce authentication prompts on Android and iOS.

For improved user experience and reduce prompts on mobile devices, we recommend using the Microsoft Authenticator.

Learn more: [Microsoft Authenticator app authentication method - Azure Active Directory](#)

% Auth App Authentications | Count of Android/iOS Devices

| 88% | 3,887



# Recommendations: Mobile Authentications

SigninLogs

```

|mv-expand ParsedFields=parse_json(AuthenticationDetails) //parse out Authentication Details column and put in new column called
"ParsedFields"
|extend AuthenticationMethod = ParsedFields.authenticationMethod //create new column by extracting the authenticationMethod
value out of the ParsedFields column
|extend AuthMethod = tostring(AuthenticationMethod) //convert to string format
|extend ParsedFields2=parse_json(DeviceDetail) //parse out Device Details column and put in column called "ParsedFields2"
|extend DeviceState = case(DeviceDetail["trustType"] == "", "Unmanaged", DeviceDetail["trustType"]) //convert to string
|extend OperatingSystem = ParsedFields2.operatingSystem //extract the operatingSystem value out of the ParsedFields2 column
|extend OS = tostring(OperatingSystem) //convert to string format
|where AuthMethod != "Previously satisfied" and AuthMethod != "" //remove Previously Satisfied and blank AuthMethods from the
query
|where UserDisplayName != "On-Premises Directory Synchronization Service Account" //remove this UserDisplayName
|where AuthMethod in ({AuthMethod}) or '*' in ({AuthMethod}) //AuthMethod parameter
|where DeviceState in ({DeviceState}) or '*' in ({DeviceState}) //DeviceState parameter
|where "{User:escape}" == "All users" or UserDisplayName contains "{User:escape}" //User parameter
|where "{App:escape}" == "All apps" or AppDisplayName contains "{App:escape}" //App parameter
|where "{OS:escape}" == "All OS" or OS contains "{OS:escape}" //OS parameter
|extend Status = ParsedFields.succeeded //extract the succeeded value out of the ParsedFields2 column
|extend AuthStatus = case(Status == "true", "Success", "Failure") //label status as Success or Failure
|where AuthStatus in ({AuthStatus}) or '*' in ({AuthStatus}) //AuthStatus parameter
|extend AuthAppOpp = case(AuthMethod == "Phone call approval (Authentication phone)", "Opportunity", AuthMethod == "Text message",
"Opportunity", AuthMethod == "Mobile app notification", "Opportunity", "StrongAuth")
|summarize TotalCount = count(), TotalAuthAppOpp = countif(AuthAppOpp != "StrongAuth"), AuthenticatorApp = countif(AuthMethod cont
ains "Mobile app notification")
|project ['% Authenticator App Sign-Ins'] = (1.0 * AuthenticatorApp/TotalAuthAppOpp), TotalCount

```

## Mobile Authentications

The Authenticator app serves as a token broker and can reduce authentication prompts on Android and iOS.

For improved user experience and reduce prompts on mobile devices, we recommend using the Microsoft Authenticator.

Learn more: Microsoft Authenticator app authentication method - Azure Active Directory

% Auth App Authentications | Count of Android/iOS Devices

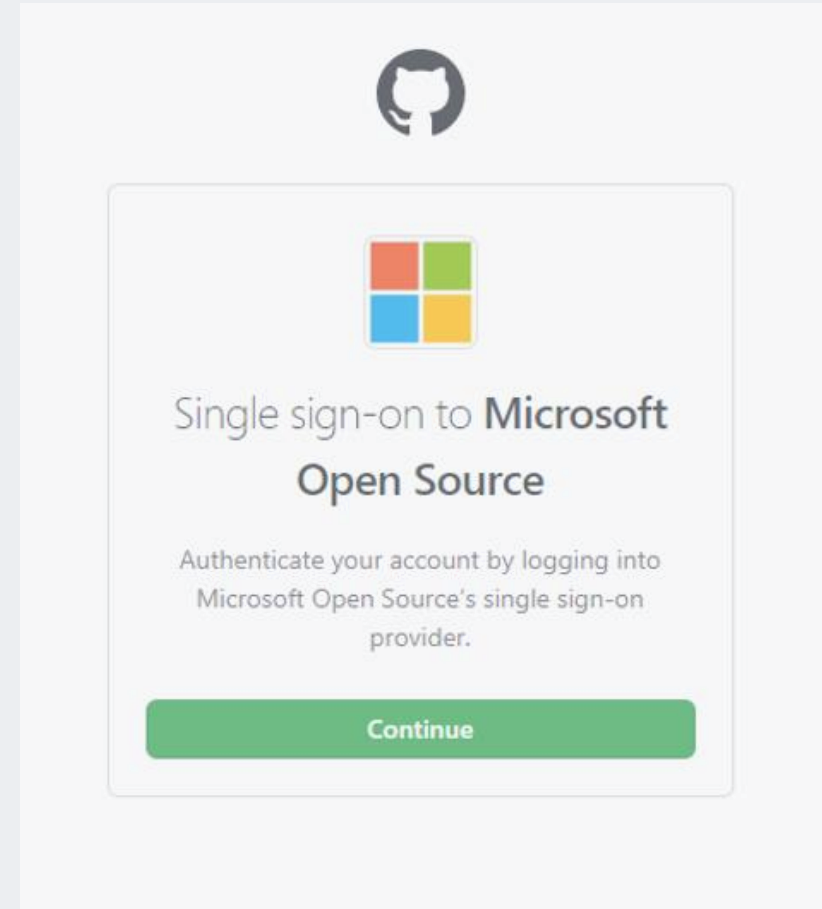
88% 3,887

# Our ask to the MVP Community...



## We want you to contribute!

- Join the [Azure AD Workbooks](#) community on GitHub
- Think about the questions you have heard from your customers/partners/associates repeatedly. Could a workbook help address this?
- Help others improve their workbooks
- Create and submit your own workbooks

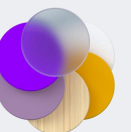


All resources shared in the presentation can be found at [aka.ms/KQLMVP](https://aka.ms/KQLMVP)

## Resources ([aka.ms/KQLMVP](https://aka.ms/KQLMVP))

- [Azure Monitor Workbooks Overview](#) – great starting place when learning about workbooks
- [Configure a Log Analytics Workspace](#) – if you don't already have a LA workspace, use this tutorial to set one up
- [How to use Azure AD workbooks](#) - a brief write-up about most of the workbooks in the gallery
- [Microsoft Organization in GitHub](#) and [@azure-ad-workbooks](#) – join this GitHub Repo and team to learn much more about Azure AD Workbooks as well as to contribute.
- [Kusto Query Language \(KQL\) from Scratch](#) – great Pluralsight Tutorial for learning KQL
- [aka.ms/PreviewWorkbooks](https://aka.ms/PreviewWorkbooks) – see Azure AD Workbooks in private preview!

All resources shared in the presentation can be found at [aka.ms/KQLMVP](https://aka.ms/KQLMVP)



# Questions and (hopefully) Answers






## We value your feedback

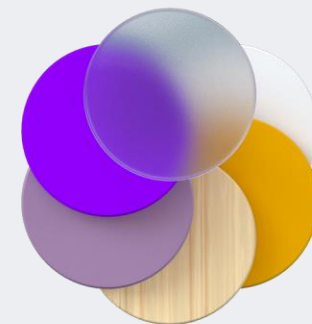
Please complete the short session survey

## Stay in touch:

Corissa Koopmans |  @Corissalea

Tosin Lufadeju |  @TosinLuf\_PM

All resources shared in the presentation can be found at [aka.ms/KQLMVP](https://aka.ms/KQLMVP)



Thank You!



# Appendix



# Scalars and Vectors

A **scalar quantity** is a quantity that has only **magnitude**.

A **vector quantity** is a quantity that has both a **magnitude** and a **direction**.

## Scalar quantities

Length, Area, Volume,  
Speed,  
Mass, Density  
Temperature, Pressure  
Energy, Entropy  
Work, Power



## Vector quantities

Displacement, Direction,  
Velocity, Acceleration,  
Momentum, Force,  
Electric field, Magnetic field

