Secureworks®

Public presentation - Detecting gaps in Azure and Azure AD Security

Azure Cloud Security Group meetup – 2. May 2023 –

Joosua Santasalo

Secureworks®

Due to the ever evolving nature of these services the information in this presentation is provided "AS IS" with no warranties and confers no rights. All opinions expressed here are my personal

Base facts: two kids, wife and dog. Lives in Helsinki

Formerly: 2017-2022 Nixu

Current: 2022 -> Secureworks

Doing:

Security research - Security tool development, to some extent offensive tooling. Tooling Mostly written in Node.js

Published research 2022-2023

- 2023 -> Stay tuned for multiple new disclosures
- MSRC: Public Disclosure AAD Privilege Escalation on SAML apps – Rating: Important
- MSRC: Public Disclosure Databricks Admin Isolation
- MSRC: Public Disclosure – Gaining Unlimited access to graph AuditLogs endpoint using complex filters with non-privileged user account
- MSRC:Public Disclosure - Joint security research write up – Azure AD Consent bypass disclosure with Kim Jamia – MSRC - Public Disclosure - Azure Monitor – Malicious KQL Query(By design... But was fixed week after disclosure went public)

- **Microsoft MVP since 2020**
- **MSRC (**Microsoft Security Response Center) - multiple Hall-of-fame mentions

Admin Isolation on Shared Clusters

by David Meyer and Joosua Santasalo

October 10, 2022 in Engineering Blog

Share this post

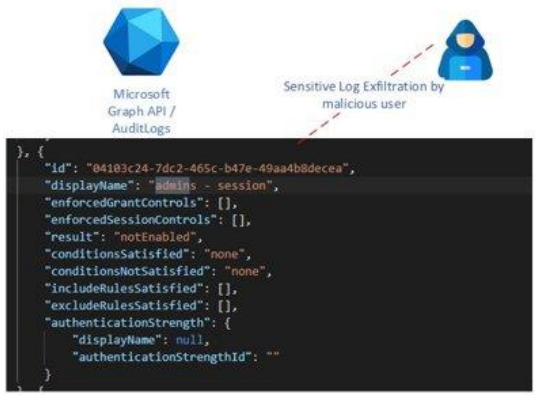
This blog was co-authored by David Meyer, SVP Product Management at Databricks and Joosua Santasalo, a security researcher with Secureworks.

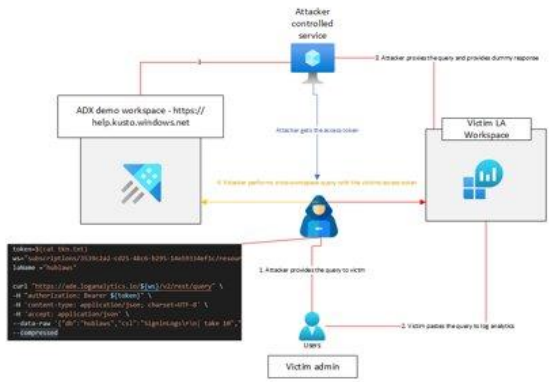
Microsoft Azure

Action Recommended: Disclosure of Basic Profile Information during Sign-in

Microsoft recently mitigated a user consent prompt issue in an uncommon configuration for certain Azure Active Directory (AAD)-registered applications. This inadvertently allowed application publishers to obtain your users' basic profile information (personal data) during the authentication process without proper consent.

The information available to applications without user consent was limited to the following personal data:





What does top gaps mean in the context of this session

In this session I highlight by demoing some very specific areas of gaps, but also show a larger more high-level area for gaps which is any initial authentication outside Azure AD

1. Top detection sources
2. Very narrow CA gaps
3. Show areas which single Azure Resource can access via managed identity
4. Key takeaways

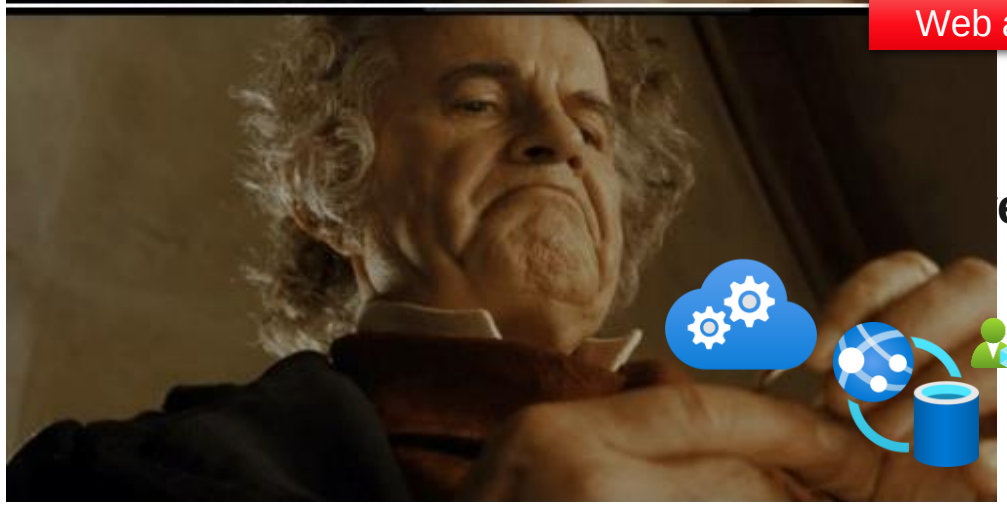
Sources for this session

sessions/vmdemo.sh at Azure_security_ug · jsa2/sessions · GitHub

sessions/useMIoutside.md at Azure_security_ug · jsa2/sessions · GitHub

jsa2/caOptics: CA Optics - Azure AD Conditional Access Gap Analyzer (github.com)

Get the basics right before worrying about detection



elop and **harden** is the one that does not exist .

Start by addressing the 98%

Cyber Signals, a cyberthreat intelligence brief informed by the latest Microsoft threat data and research - [Cyber Signals (microsoft.com)]

Overview for detecting gaps

Azure AD
Initial/ refresh authentication and authorization

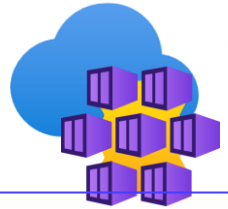
Azure RBAC assignments

Control plane

Activity logs

All requests are made thru Azure Resource Manager URL (e.g. POST management.azure.com/subscriptions/…)

Azure AD Authenticating principal



Data plane

Diagnostic logs

Service specific access URL to service runtime (e.g POST /endpoint/v1.0/create)

Sign-in logs

Audit logs

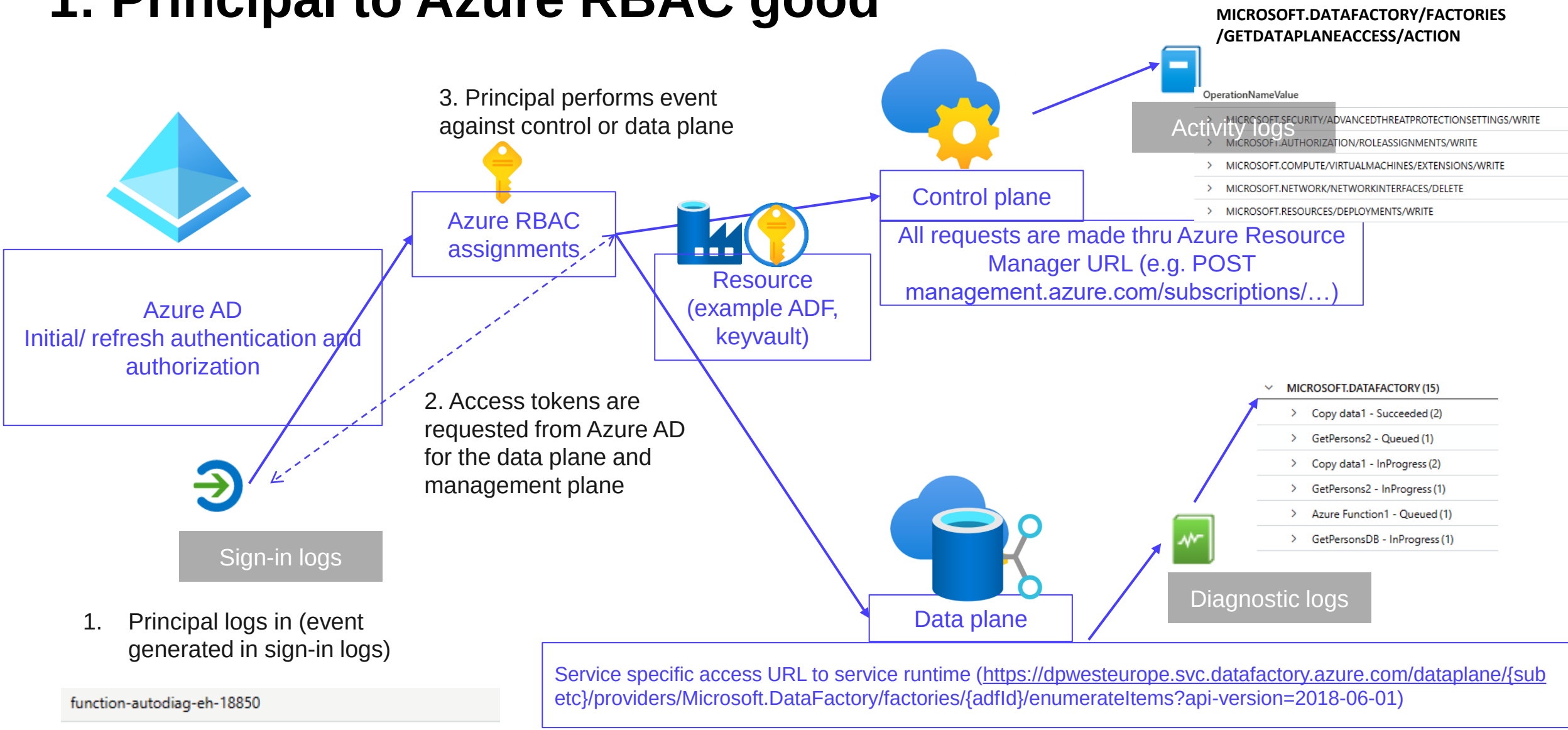
IaaS workloads

Azure Monitor Agent

SysLog / Windows logs

Secureworks

1: Principal to Azure RBAC good



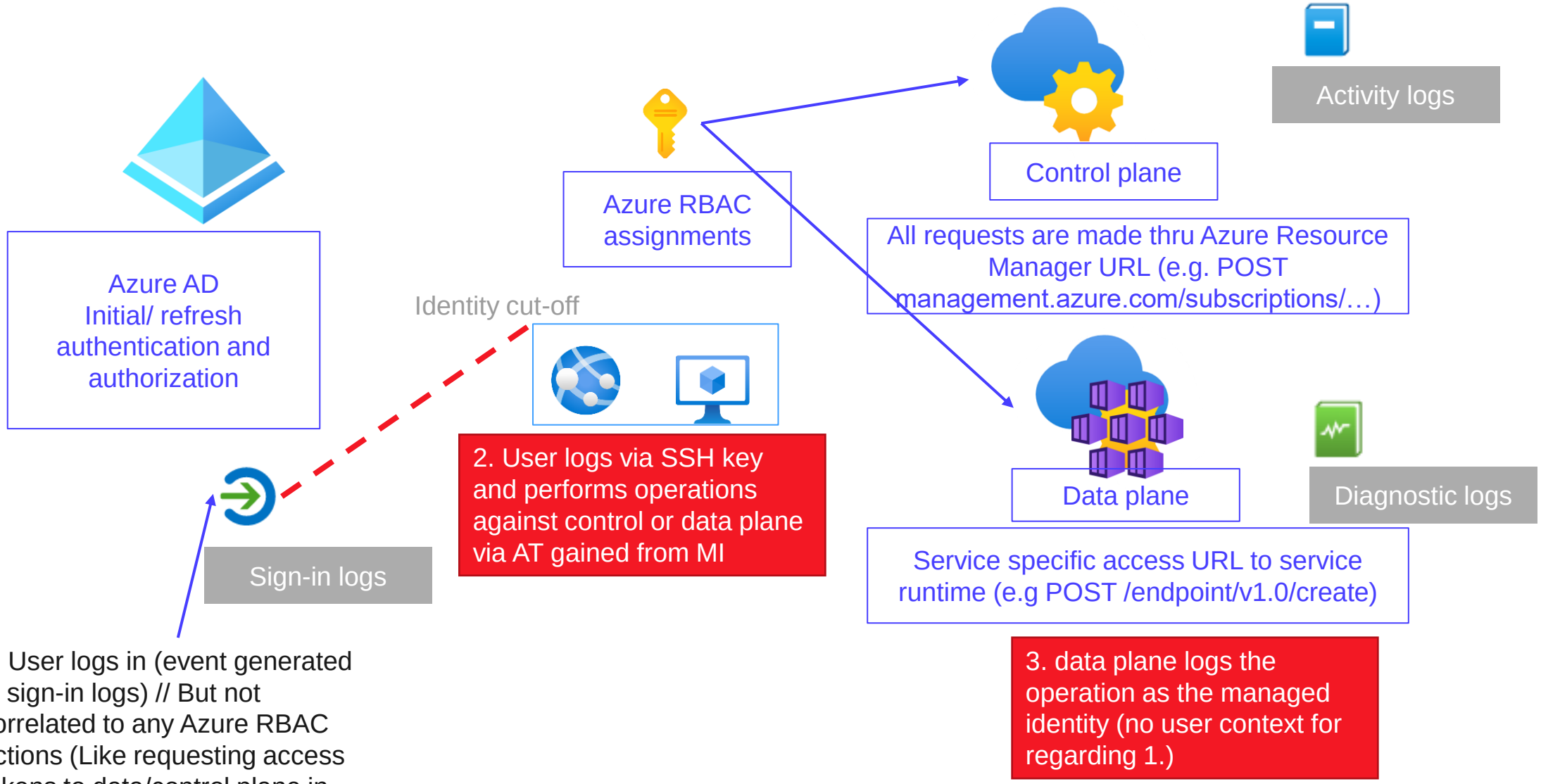
1. Principal logs in (event generated in sign-in logs)

function-autodiag-eh-18850

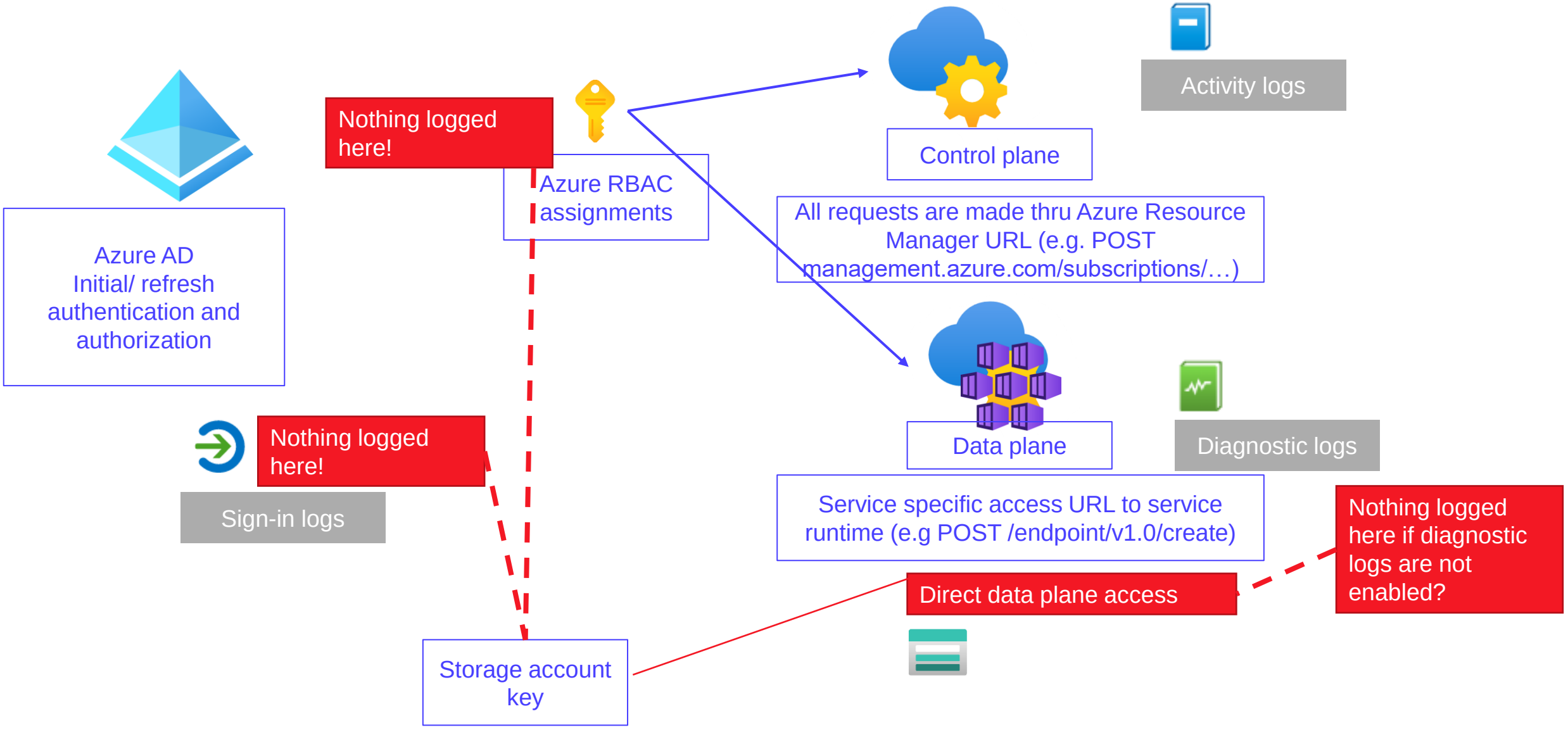
["Azure Key Vault","Windows Azure Service Management API"]

Secureworks

2: User Identity to Azure RBAC: Identity cut-off



3: No AAD, no RBAC



Secureworks

Read about various resources on ensuring data and control plane access is backed by AAD

Notes regarding non AAD-based authentication options

- All string-based authentication methods in table below are considered **passwords**.

❌Password/String-based authentication is not considered strong in terms of strength, as shown in the table below. Even though security can be increased with password length, and password rotation.

- The table assumes rotation because password can be leaked

❌Bypasses Azure AD logs means that no events are produced for the resource type in Azure AD logs when the authentication mechanism is used.

❌ Susceptible to sharing across multiple targets in the table means that the service allows human defined password (monkey/dog/cat/birthday) creation, thus allowing the admin to re-use passwords across systems

- Services which generate these passwords, and don't allow admins to input passwords are considered not shareable in the same sense (not all services in the table can be shared across systems, but can be shared across clients still)

  Note: Even if the key is system generated, it can be obviously leaked If a SAS is leaked, it can be used by anyone who obtains it, which can potentially compromise resource utilizing SAS scheme

Notes regarding AAD-based authentication options

✔ Can be authorized based on Azure RBAC settings on services, and Azure AD roles granted for the API permissions

✔ Are logged in both Azure AD and service specific logs

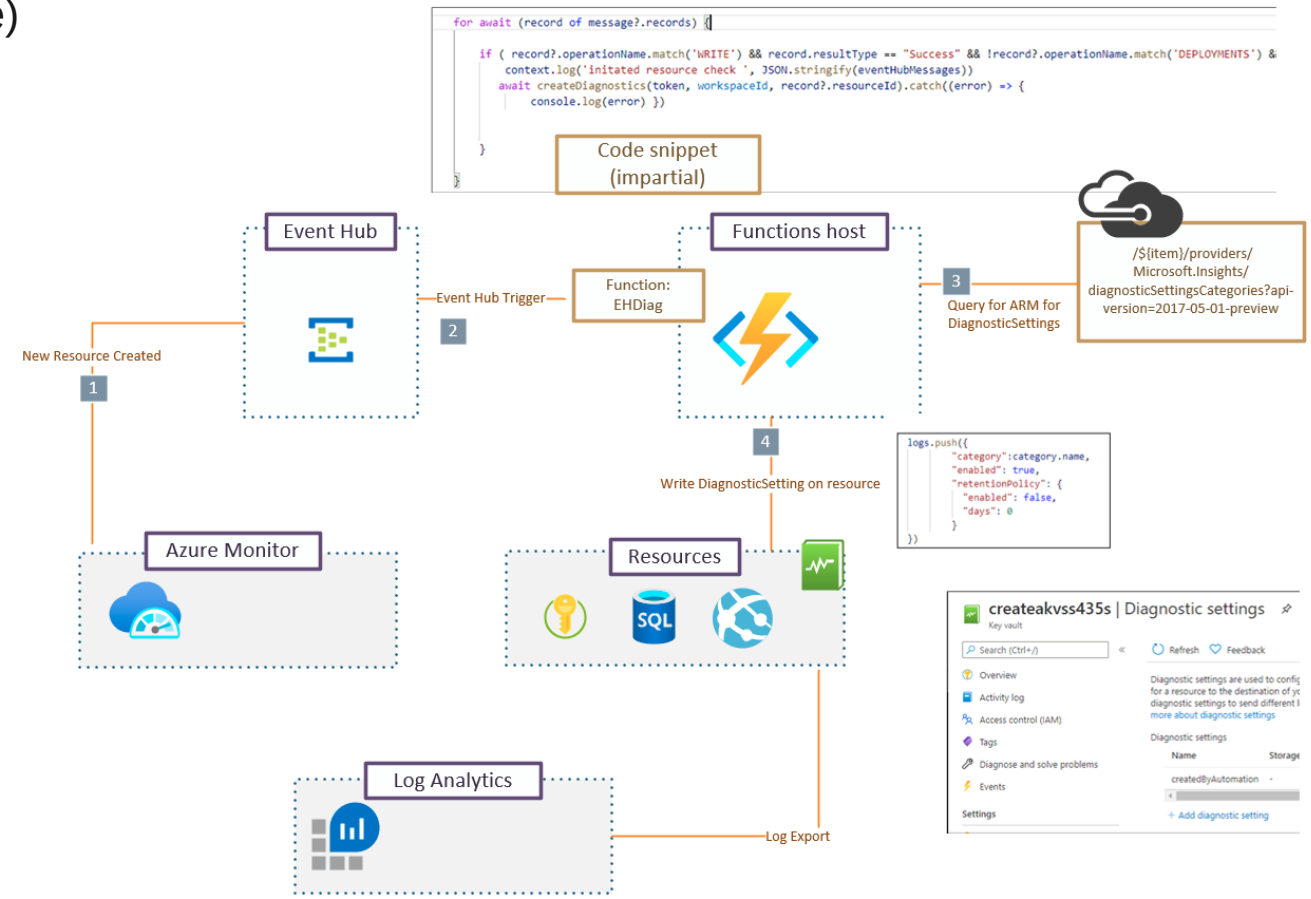
✔ Can be managed as objects exposed in Azure AD and Azure RBAC (supports listing, filtering, policies and have specific properties which are exposed to configuration plane)

Service Table

- Column *Service logs* in table mean that there is logging option outside Azure, which typically includes the authentication information.

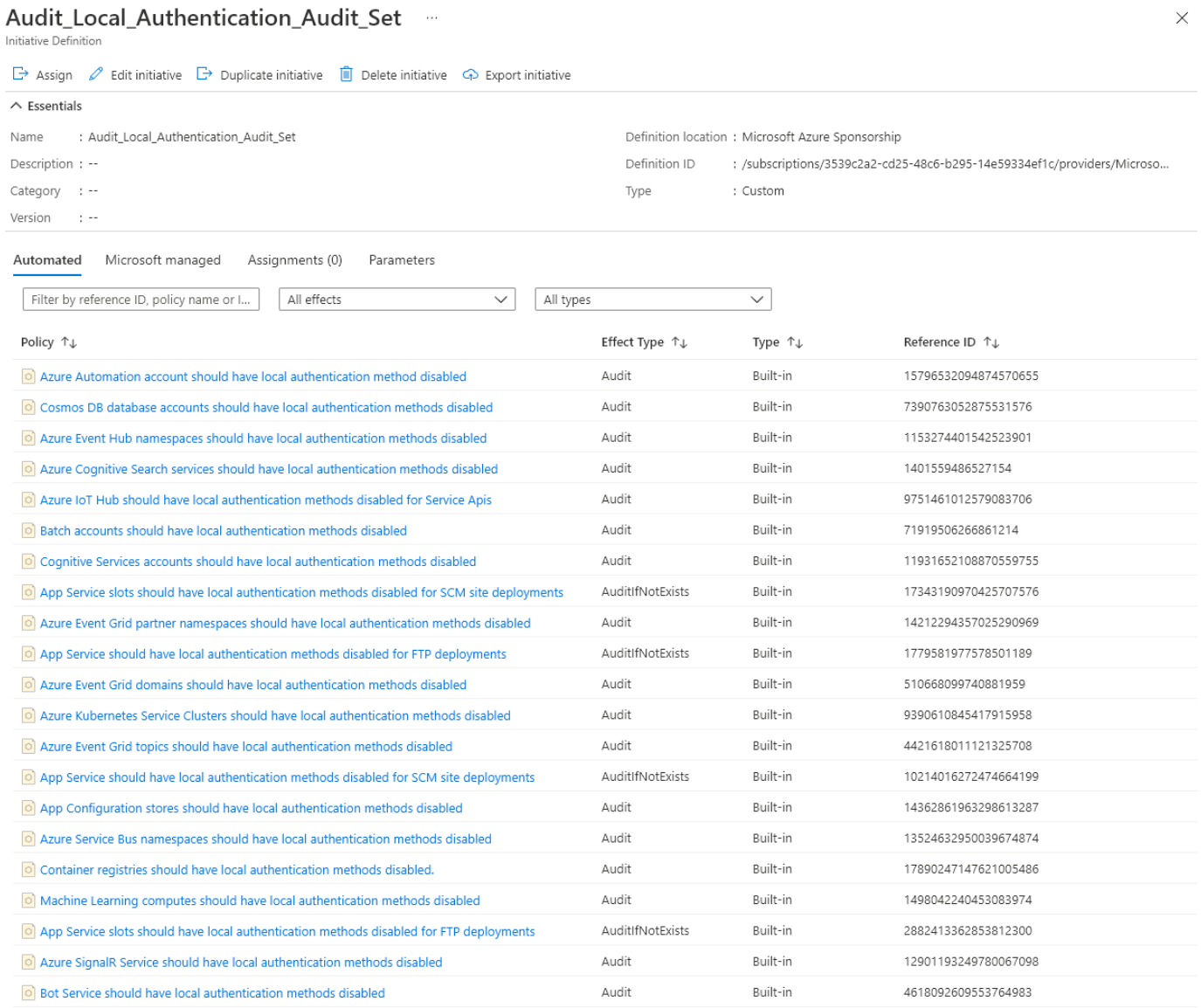
Important about Azure Logging

- Diagnostic logs are not enabled by default
  - Use Azure Policy, or some other form of automation at suitable stage (deployment pipeline, or post deployment pipeline)



Secureworks

Audit where local auth is used



Audit_Local_Authentication_Audit_Set

Initiative Definition

Assign | Edit initiative | Duplicate initiative | Delete initiative | Export initiative

Essentials

Name : Audit_Local_Authentication_Audit_Set
Description : --
Category : --
Version : --

Definition location : Microsoft Azure Sponsorship
Definition ID : /subscriptions/3539c2a2-cd25-48c6-b295-14e59334ef1c/providers/Microso...
Type : Custom

Automated | Microsoft managed | Assignments (0) | Parameters

Filter by reference ID, policy name or I... | All effects | All types

Policy	Effect Type	Type	Reference ID
Azure Automation account should have local authentication method disabled	Audit	Built-in	15796532094874570655
Cosmos DB database accounts should have local authentication methods disabled	Audit	Built-in	7390763052875531576
Azure Event Hub namespaces should have local authentication methods disabled	Audit	Built-in	1153274401542523901
Azure Cognitive Search services should have local authentication methods disabled	Audit	Built-in	1401559486527154
Azure IoT Hub should have local authentication methods disabled for Service Apis	Audit	Built-in	9751461012579083706
Batch accounts should have local authentication methods disabled	Audit	Built-in	71919506266861214
Cognitive Services accounts should have local authentication methods disabled	Audit	Built-in	11931652108870559755
App Service slots should have local authentication methods disabled for SCM site deployments	AuditIfNotExists	Built-in	17343190970425707576
Azure Event Grid partner namespaces should have local authentication methods disabled	Audit	Built-in	14212294357025290969
App Service should have local authentication methods disabled for FTP deployments	AuditIfNotExists	Built-in	1779581977578501189
Azure Event Grid domains should have local authentication methods disabled	Audit	Built-in	510668099740881959
Azure Kubernetes Service Clusters should have local authentication methods disabled	Audit	Built-in	9390610845417915958
Azure Event Grid topics should have local authentication methods disabled	Audit	Built-in	4421618011121325708
App Service should have local authentication methods disabled for SCM site deployments	AuditIfNotExists	Built-in	10214016272474664199
App Configuration stores should have local authentication methods disabled	Audit	Built-in	14362861963298613287
Azure Service Bus namespaces should have local authentication methods disabled	Audit	Built-in	13524632950039674874
Container registries should have local authentication methods disabled.	Audit	Built-in	17890247147621005486
Machine Learning computes should have local authentication methods disabled	Audit	Built-in	1498042240453083974
App Service slots should have local authentication methods disabled for FTP deployments	AuditIfNotExists	Built-in	2882413362853812300
Azure SignalR Service should have local authentication methods disabled	Audit	Built-in	12901193249780067098
Bot Service should have local authentication methods disabled	Audit	Built-in	4618092609553764983

[jsa2/aad-auth-n-z (github.com)]

Secureworks

Important about Azure Logging

- Azure Activity and and Azure AD Audit Logging is focused on write operations

What is it?

Audit logs in Azure AD provide access to system activity records, often needed for compliance. This log is categorized by user, group, and application management.

With a user-centric view, you can get answers to questions such as:

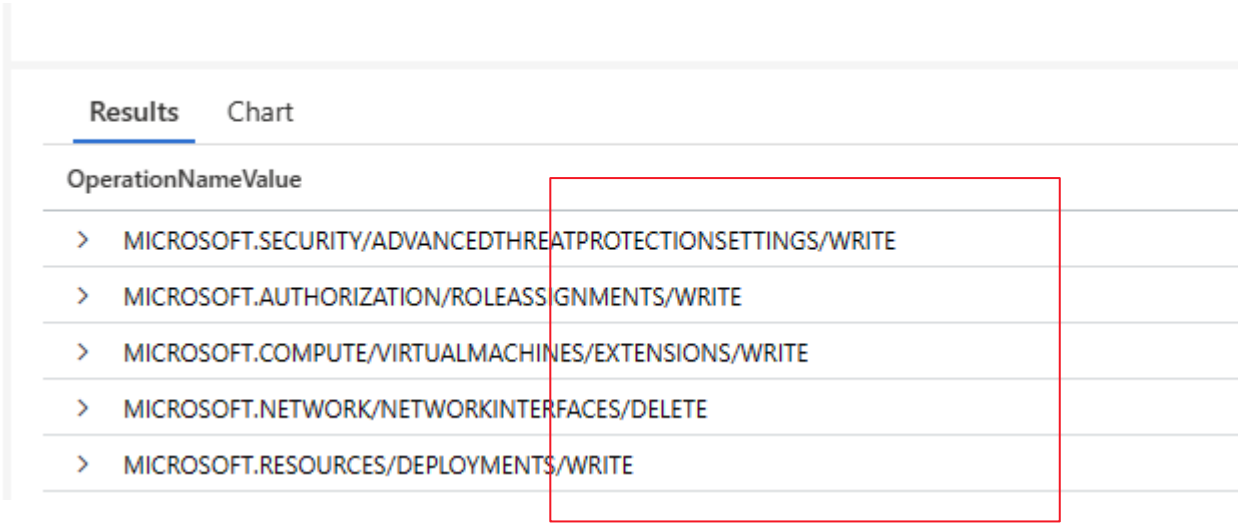
- What types of updates have been applied to users?
- How many users were changed?
- How many passwords were changed?
- What has an administrator done in a directory?

With a group-centric view, you can get answers to questions such as:

- What are the groups that have been added?
- Are there groups with membership changes?
- Have the owners of group been changed?
- What licenses have been assigned to a group or a user?

With an application-centric view, you can get answers to questions such as:

- What applications have been added or updated?
- What applications have been removed?
- Has a service principal for an application changed?
- Have the names of applications been changed?
- Who gave consent to an application?



Secureworks

Important about Azure Logging

- Many external integrations just read Office 365 Management API and Azure Activity logs
  - This does not replace the need for Azure AD & Azure Diagnostic logs (especially Audit, non user sign-ins and Non-interactive), or need for data plane logs in Azure (Diagnostic settings)

Click 'Add Diagnostic setting' above to configure the collection of the following data:

- AuditLogs
- SignInLogs
- NonInteractiveUserSignInLogs
- ServicePrincipalSignInLogs
- ManagedIdentitySignInLogs
- ProvisioningLogs
- ADFSSignInLogs
- RiskyUsers
- UserRiskEvents
- NetworkAccessTrafficLogs
- RiskyServicePrincipals
- ServicePrincipalRiskEvents
- EnrichedOffice365AuditLogs
- MicrosoftGraphActivityLogs

ResourceProvider
MICROSOFT.NETWORK
MICROSOFT.LOGIC
MICROSOFT.SQL
MICROSOFT.COGNITIVESERVICES
MICROSOFT.KEYVAULT
MICROSOFT.SERVICEBUS
MICROSOFT.CONTAINERSERVICE
MICROSOFT.RELAY
MICROSOFT.DATAFACTORY

Importance of using Azure AD for authentication and authorization across any service deployed to Azure

- Operations and authorizations to services completed via local authentication methods (SSH, PAT, Storage Account Keys, Deployment Keys etc) are not logged in
  - Azure AD Audit logs
  - Azure Activity logs
  - Azure AD Sign-in logs

Diagnostic settings

Besides this logs indicating data plane access for example reading blob from storage account with SAS token or storage account, will lack identity context.

Important about Azure Logging

- Authorization Failures against Azure Resource Manager Control Plane in most cases are not logged

```
curl -s -d "{}" $storagePath  -H "Authorization: Bearer $at2" | jq .
{
  "error": {
    "code": "AuthorizationFailed",
    "message": "The client 'a167d919-8ff7-4136-a6b9-59df089fd289' with object id 'a167d919-8ff7-4136-a6b9-59df089fd289' does not ha
ve authorization to perform action 'Microsoft.Storage/storageAccounts/listKeys/action' over scope '/subscriptions/3539c2a2-cd25-48c
6-b295-14e59334ef1c/resourceGroups/rg-eastdemovm31666/providers/Microsoft.Storage/storageAccounts/storagekgczvmlfyb' or the scope i
s invalid. If access was recently granted, please refresh your credentials."
  }
}
```

Secureworks®

Policy Name	State	Creation Date	Modified Date
Require multifactor authentication for admins	On	4/27/2023, 5:08:16 PM	
CAP-android-mobileAppsAndDesktopClients-O365	On	4/27/2023, 5:08:18 PM	
CAP-Guest Inviter-management	On	4/27/2023, 5:08:19 PM	
wp close all for browser	On	4/27/2023, 5:08:21 PM	4/27/2023, 5:48:10 PM
CAP-Service Support Administrator-management	On	4/27/2023, 5:08:22 PM	
CAP-Billing Administrator-management	On	4/27/2023, 5:08:22 PM	
App admin browser	On	4/27/2023, 5:08:20 PM	
CAP-linux-browser-O365	On	4/27/2023, 5:08:24 PM	
CAP-windows-browser-O365	On	4/27/2023, 5:08:25 PM	
WindowsPhone-fix	On	4/27/2023, 5:08:24 PM	
CAP-macOS-browser-O365	On	4/27/2023, 5:08:23 PM	
CAP-Helpdesk Administrator-management	On	4/27/2023, 5:08:27 PM	
CAP-Guest User-management	On	4/27/2023, 5:08:26 PM	
CAP-Restricted Guest User-management	On	4/27/2023, 5:08:26 PM	
CAP-windows-mobileAppsAndDesktopClients-O365	On	4/27/2023, 5:08:28 PM	
CAP-Global Administrator-management	On	4/27/2023, 5:08:29 PM	
CAP-User Administrator-management	On	4/27/2023, 5:08:29 PM	
CAP-linux-mobileAppsAndDesktopClients-O365	On	4/27/2023, 5:08:31 PM	
	On	4/27/2023, 5:08:31 PM	
	On	4/27/2023, 5:08:32 PM	
CAP-macOS-mobileAppsAndDesktopClients-O365	On	4/27/2023, 5:08:33 PM	
		4/27/2023, 5:08:33 PM	
		4/27/2023, 5:08:34 PM	
		4/27/2023, 5:08:36 PM	
		4/27/2023, 5:08:36 PM	4/27/2023, 6:01:19 PM
		4/27/2023, 5:08:37 PM	
CAP-GA-Admins-Azure	On	4/27/2023, 5:08:38 PM	
CAP-android-browser-O365	On	4/27/2023, 5:08:38 PM	
CAP-iOS-mobileAppsAndDesktopClients-O365	On	4/27/2023, 5:08:39 PM	
app admin	On	4/27/2023, 5:53:20 PM	
guests all but OVPN	On	4/27/2023, 5:54:09 PM	4/27/2023, 5:55:40 PM
ovpn enforce all platforms	On	4/27/2023, 5:55:11 PM	4/27/2023, 5:56:30 PM
guest all locations	On	4/27/2023, 5:57:18 PM	

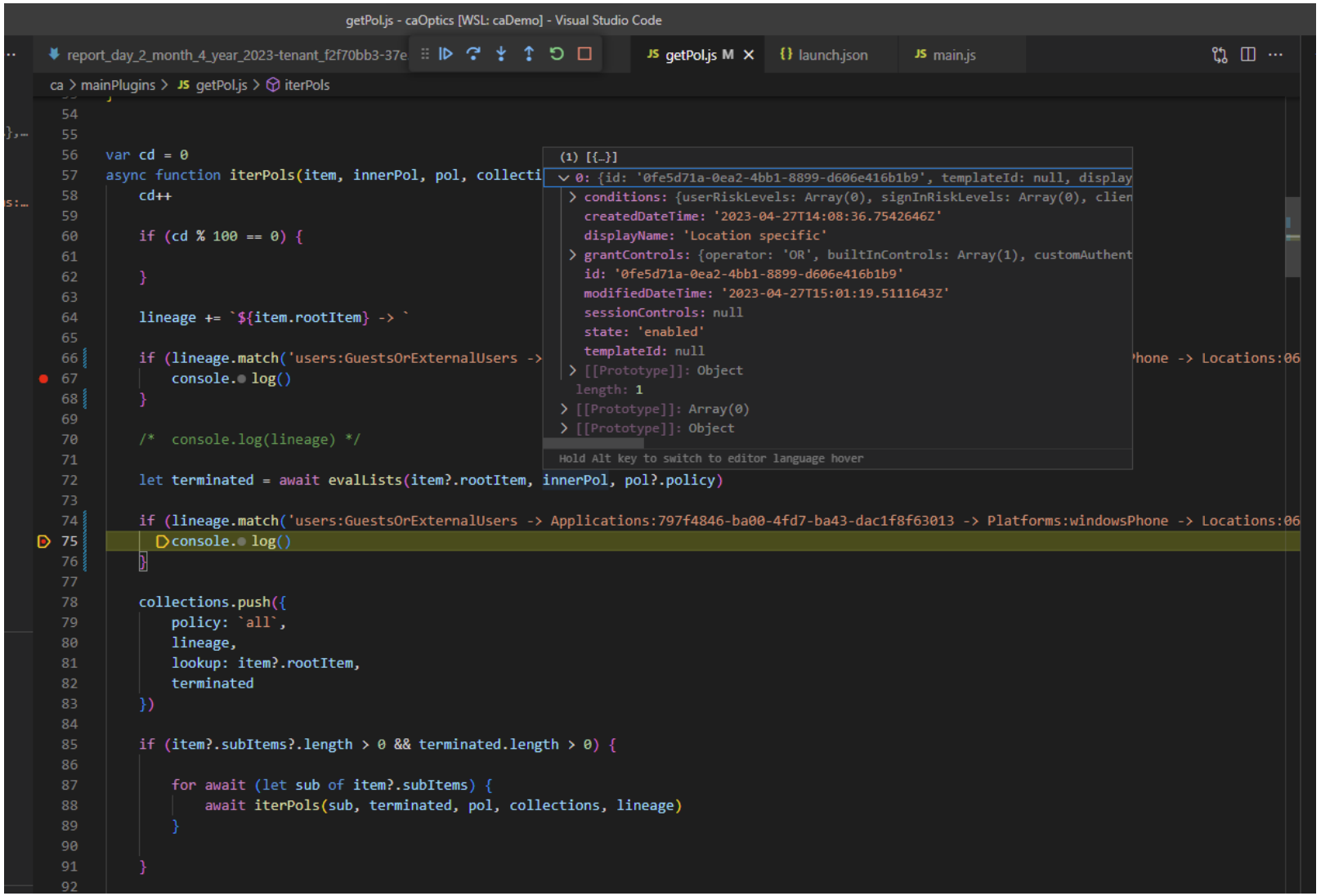
Detecting a needle in a haystack

Policy	Terminations	lookup
all	0	users:GuestsOrExternalUsers -> Applications:Windows Azure Service Management API -> Platforms:windowsPhone -> Locations:Netherlands -> clientAppTypes:browser ->

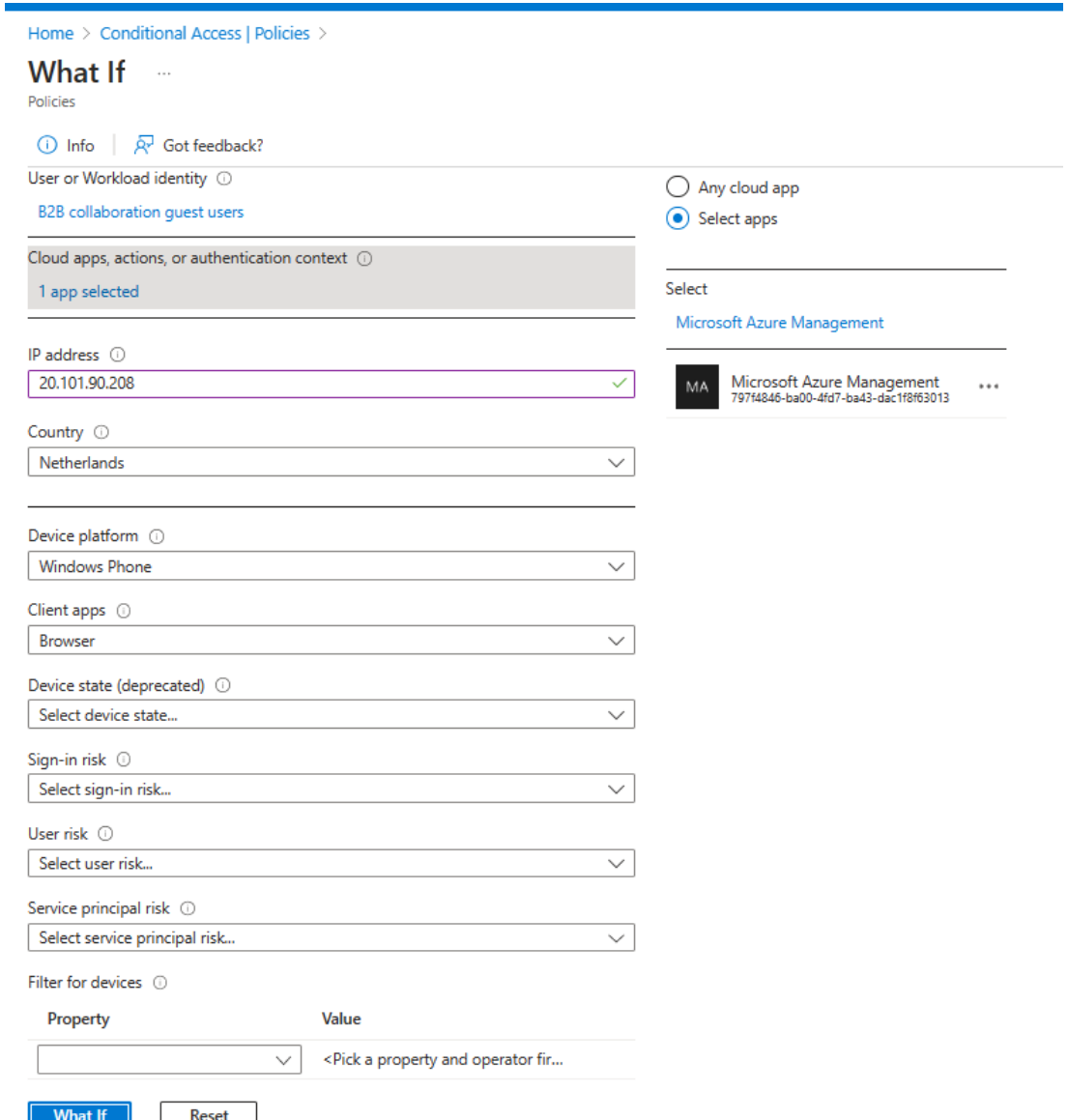
Detecting a needle in a haystack

Video https://youtu.be/l2599nha0M4?t=1835

Detecting a needle in a haystack



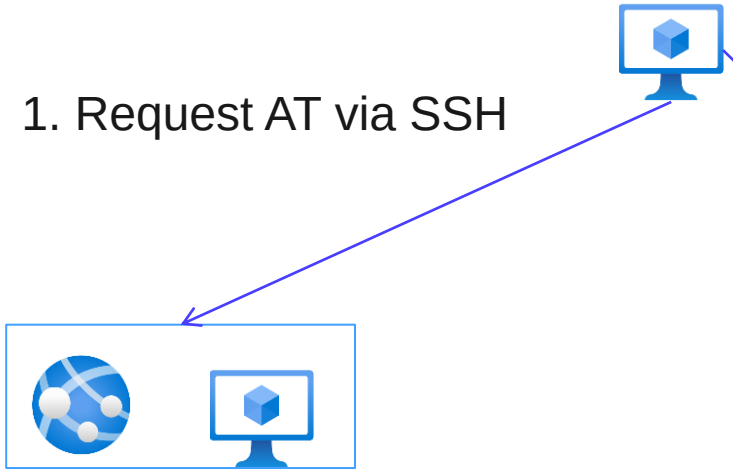
Detecting a needle in a haystack



Detecting MI access token used outside of Azure

ssh -i tempkeys/tempkey azureuser@$ip "curl -s 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net' -H Metadata:true | jq .access_token "

1. Request AT via SSH



Multiple Azure Services supporting managed identity's

2. Use AT outside of Azure

curl -s 'https://eastdemokv.vault.azure.net/secrets/vmsecret/?api-version=2016-10-01' -H 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ii1LSTNROW5OUjdiUm9m

Azure Key Vault access

ServicePrincipalName	eastdemovm8943
appId	a87ace34-d027-440c-8218-ac78bfd33567

union AADServicePrincipalSignInLogs, AADManagedIdentitySignInLogs

mass	{"TenantId":"f8bbcf06-6cd4-45d9-a0a1-c1f89e40d7b9","Time
AdditionalFields	{"identity_claim_xms_az_nwperimid":"[]"}
CallerIPAddress	87.92.21.175
Category	AuditEvent
CorrelationId	885f3541-859f-4e13-b197-00c0bab7847f
DurationMs	95
OperationName	SecretGet

union AzureDiagnostics, AzureActivity

Methods - Compare against AAD log ip to IP in Azure logs

For SPNLogs This is simple, but does not work against MI logs. (below is a simple example)

```
AzureActivity
| where TimeGenerated > now() -180d
| extend parsedClaims = parse_json(Claims_d)
| extend appid = tostring(parsedClaims.appid)
| join kind=inner AADServicePrincipalSignInLogs on $left.appid == $right.AppId
| project TimeGenerated, TimeGenerated1, OperationNameValue, IPfromActivityLogs = CallerIpAddress, IPfromSPNLogs = IPAddress, appid, HTTPRequest, parsedClaims
| sort by TimeGenerated, TimeGenerated1 desc
| where IPfromActivityLogs != IPfromSPNLogs
```

- MI logs don't contain IP

ServicePrincipalName	IPA...	Type
fn-roleMon-22114		AADManagedIdentitySignInLogs
storagebinding		AADManagedIdentitySignInLogs

- SPN logs do contain IP

ServicePrincipalName	IPAddress	Type
withFedCred		AADManagedIdentitySignInLogs
thx138-kvref-6193053b-408b-4...	34.93.135.86	AADServicePrincipalSignInLogs

Methods - Compare against Azure IP's

Automatic updates with Azure Function

```
let lookBack = 90d;
let spns = AADServicePrincipalSignInLogs
| where TimeGenerated > ago(lookBack)
| project authTime=TimeGenerated, AppId, ServicePrincipalName, aadIp = IPAddress
| summarize count() by bin(authTime, 1d), AppId, aadIp, ServicePrincipalName
| summarize  make_list(pack_all(true));
let azIp =externaldata(changeNumber: string, cloud: string, values: dynamic)
["https://fn-azip-28667.azurewebsites.net/api/content/blob.json"
] with(format='multijson')
| mv-expand values
| project  aId =values.id, prefix =values.properties.addressPrefixes
| mv-expand prefix
| project aId, prefix;
azIp
| mv-apply spn= toscalar(spns) to typeof(dynamic) on
    (
extend matc = ipv4_is_in_range(tostring(spn.aadIp),tostring(prefix))
    )
| where matc == true
| evaluate bag_unpack(spn)
```

Map Azure IP list to SPN sign-ins

✅ Maps Azure IP range & service to SPN signins counted per day

- Update download link from Azure IP Ranges and Service Tags – Public Cloud to url in `externaldata()`
- Inspired by similar sentinel query for tenant sign-ins `Azure Portal Signin from another Azure Tenant`
- The use of evaluate bag_unpack() was heavily inspired by work of Matthew Zorich

Query

```
let lookBack = 180d;
let spns = AADServicePrincipalSignInLogs
| where TimeGenerated > ago(lookBack)
| project authTime=TimeGenerated, AppId, ServicePrincipalName, aadIp = IPAddress
| summarize count() by bin(authTime, 1d), AppId, aadIp, ServicePrincipalName
| summarize  make_list(pack_all(true));
let azIp =externaldata(changeNumber: string, cloud: string, values: dynamic)
["https://download.microsoft.com/download/7/1/D/71D86715-5596-4529-9B13-DA13A5DE5B63/ServiceTags_Public_20220613.json"]
with(format='multijson')
| mv-expand values
| project  aId =values.id, prefix =values.properties.addressPrefixes
| mv-expand prefix
| project aId, prefix;
azIp
| mv-apply spn= toscalar(spns) to typeof(dynamic) on
    (
extend matc = ipv4_is_in_range(tostring(spn.aadIp),tostring(prefix))
    )
| where matc == true
| evaluate bag_unpack(spn)
```

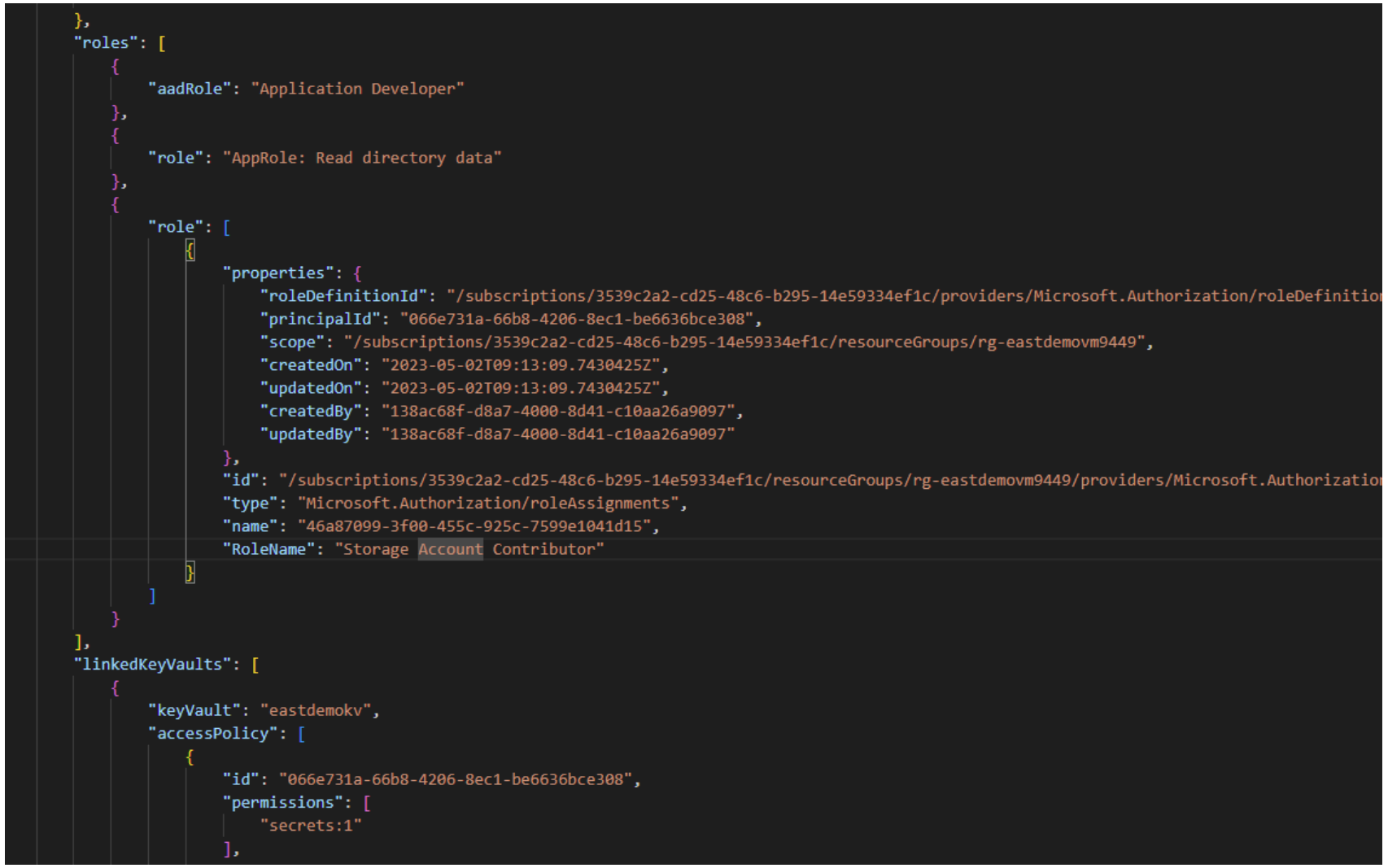
Result _ Compare against Azure IP's

ServicePrincipalName	combOp	ipByAsIdentifiedByAzure	Type	matchF...	ald	prefix
eastdemovm8943	SecretGet	87.92.21.175	AzureDiagnostics	false		
fn-roleMon-22114	SecretGet	52.178.79.163	AzureDiagnostics	true	AppService	52.178.79.163/32
function-autodiag-eh-18850	SecretGet	52.178.92.96	AzureDiagnostics	true	AppService	52.178.92.96/32
fn-roleMon-22114	SecretGet	52.178.79.163	AzureDiagnostics	true	AppService.WestEurope	52.178.79.163/32
function-autodiag-eh-18850	SecretGet	52.178.92.96	AzureDiagnostics	true	AppService.WestEurope	52.178.92.96/32
fn-roleMon-22114	MICROSOFT.STORAGE/STORAGEACCOUNTS/REGENER...	4.175.80.61	AzureActivity	true	AzureCloud.westeurope	4.175.0.0/16
fn-roleMon-22114	MICROSOFT.STORAGE/STORAGEACCOUNTS/REGENER...	20.8.73.154	AzureActivity	true	AzureCloud.westeurope	20.8.0.0/16

No match in Azure IP ranges

Match ok!

How about reviewing all data related to this service?



Secureworks

Checklist

1. Keep admin access separate and access policies simple (Separate admin accounts, and no complex CA policies, complex policies create gaps)
    1. When possible, avoid using synchronized (on-prem identities) for administrative purposes
2. Use as much as possible Managed Identities for outbound auth (no need to rotate and manage credentials, but you still need to secure access to them)
    1. When managed identity is not possible use certificate based authentication for flows using Client_Credentials
3. Use as much as possible Azure AD for inbound auth
4. disable / audit local auth with Azure Policy
5. Export critical logs at minimum (recommended is also evaluation (SIEM/XDR))
6. Lock down the possibility of transferring subscriptions
7. For any service mostly working with inbound traffic, limit the outbound calls (most C2 comms happen on reverseShells, thus relying on outbound)
8. Prevent users from registering and consenting applications themselves
9. Block users from Accessing Azure Resource Manager by default
10. Tag resources and delete unused ones. The least attack surface resource can have is the attack surface of deleted resource

Identity and access management checklist - Microsoft Azure Well-Architected Framework | Microsoft Learn

Secureworks®