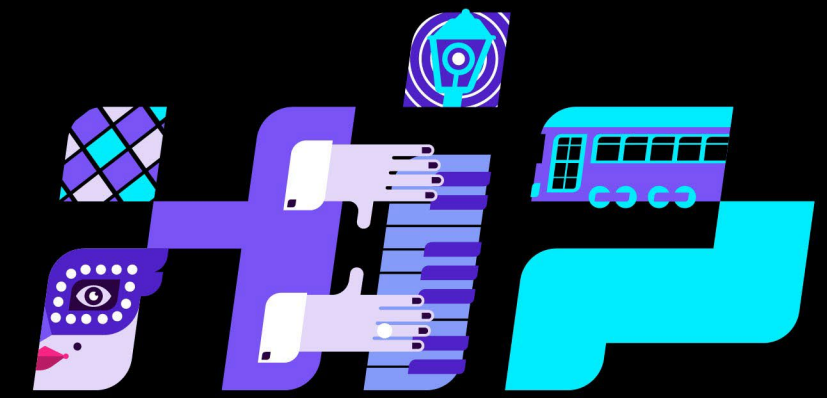
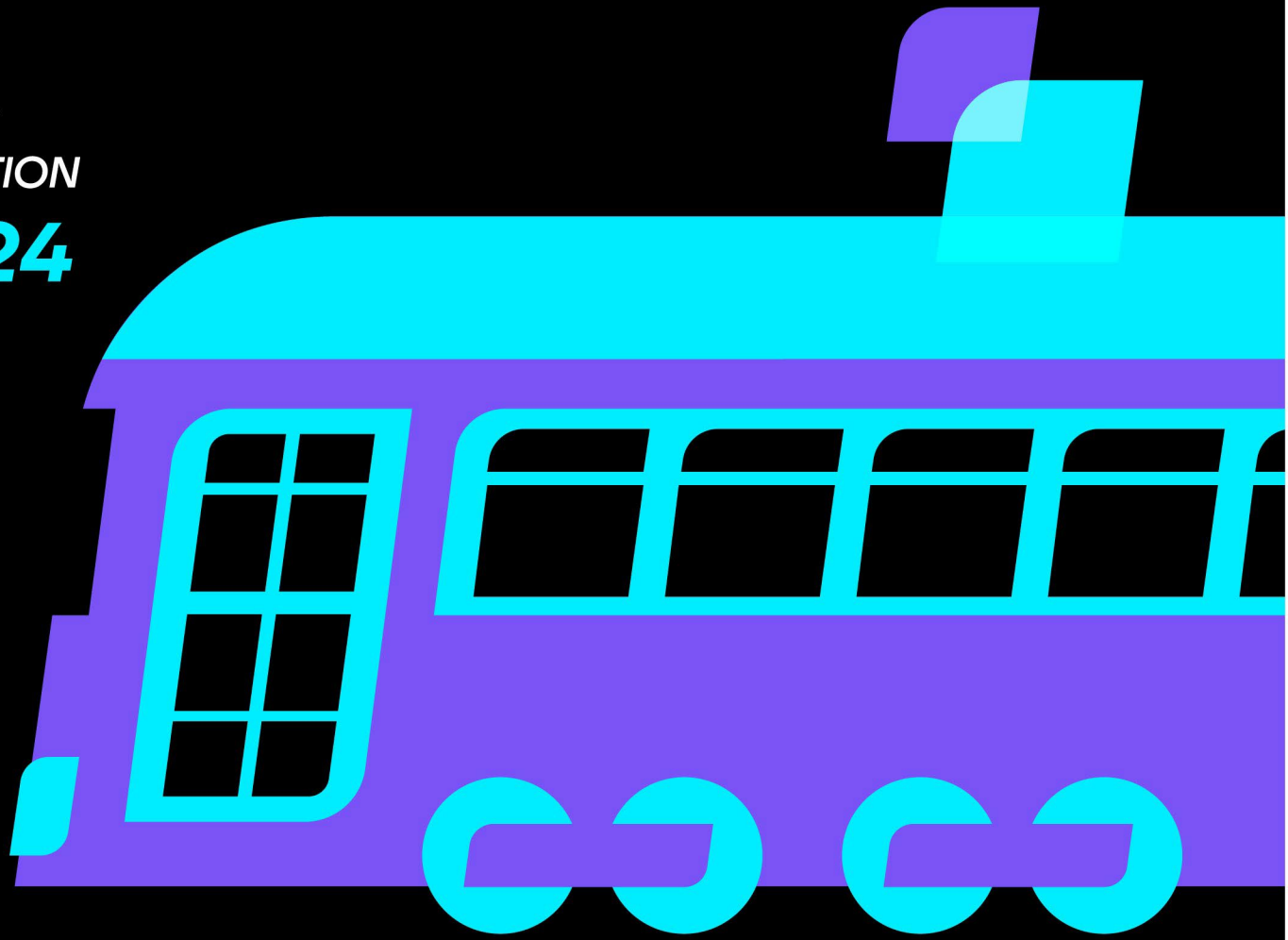
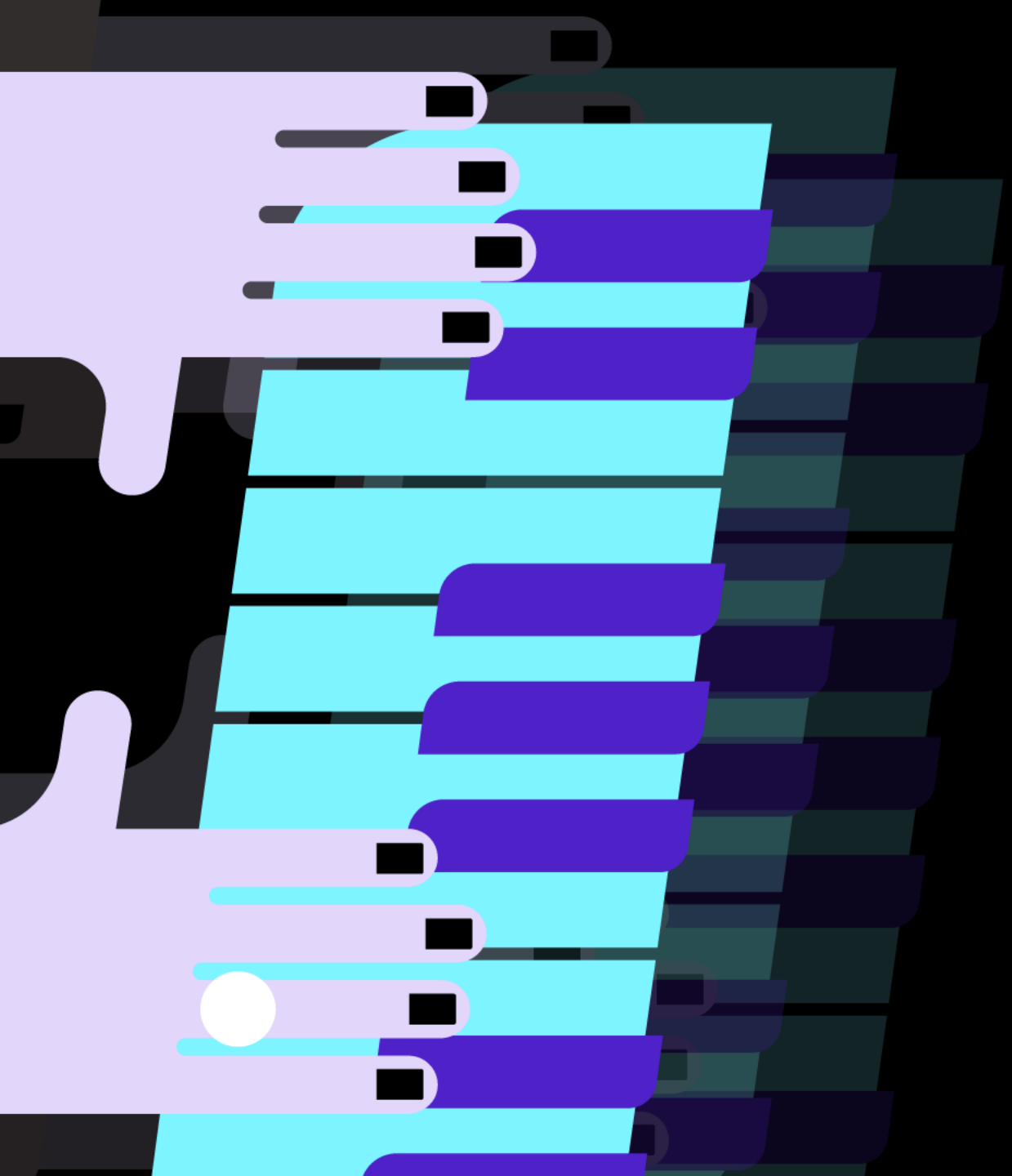


HYBRID
IDENTITY
PROTECTION

conf24

NEW ORLEANS





HYBRID IDENTITY PROTECTION conf24

NEW ORLEANS

Identity Theft is not a Joke, Azure

Karl Fosaaen

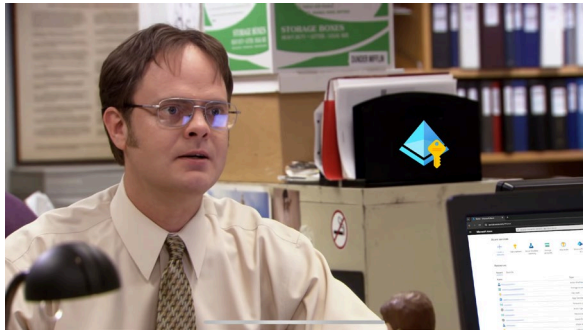
VP of Research at NetSPI

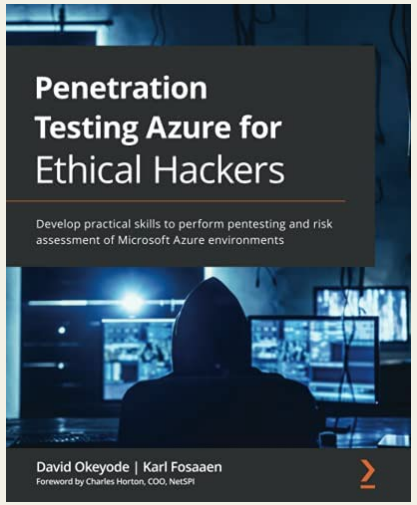
- Co-Author – Penetration Testing Azure for Ethical Hackers
- GitHub - https://github.com/NetSPI/
    - Creator – MicroBurst
    - Co-Creator – FuncoPop
- Private Pilot
- Managed Identity Thief

twitter.com/kfosaaen

linkedin.com/in/karl-fosaaen

LinkTree QR Code





FuncoPOP

MicroBurst
TOOLS.NETSPI.COM

Previous Research

Rogier Dijkman – *Project Miaow (Privilege Escalation from an ARM template)*

Rogier Dijkman – *Why you should avoid using federated credentials*

Andy Robbins – *Managed Identity Attack Paths, Part 1: Automation Accounts*

Andy Robbins – *Managed Identity Attack Paths, Part 2: Logic Apps*

Andy Robbins – *Managed Identity Attack Paths, Part 3: Function Apps*

Andy Robbins – *Abusing Azure App Service Managed Identity Assignments*

Andy Robbins – *Automating Azure Abuse Research — Part 1*

Yanir Tsarimi – *AutoWarp: Critical Cross-Account Vulnerability in Microsoft Azure Automation Service*

Huy Kha – *Lateral Movement With Managed Identities Of Azure Virtual Machines*

David Fiser – *An Analysis of Azure Managed Identities Within Serverless Environments*

Dirk-jan Mollema – *Persisting on Entra ID applications and User Managed Identities with Federated Credentials*

Today's Agenda

- Managed Identities Overview
- Managed Identity Attacks by Service
- Abusing Deployment Scripts
- Finding Managed Identity Certificates
- Abusing Tokens and Certificates
- Conclusions & Questions

Managed Identities Overview

What are Managed Identities?

Method for Authenticating an Azure Resource as an Entra ID Principal

- Utilized for granting access to other Azure/Entra ID resources
- Example:
  - Virtual Machine needs to read a secret from a Key Vault
    - Assign a Managed Identity to the VM
    - Get a Key Vault token from the IMDS
    - Use the token with REST APIs to access the Key Vault
- Eliminates the need for stored credentials on the Resource

Types of Managed Identities

- System-Assigned
  - Only associated with one resource
- User-Assigned
  - Can be shared across resources
  - Subscription level resource

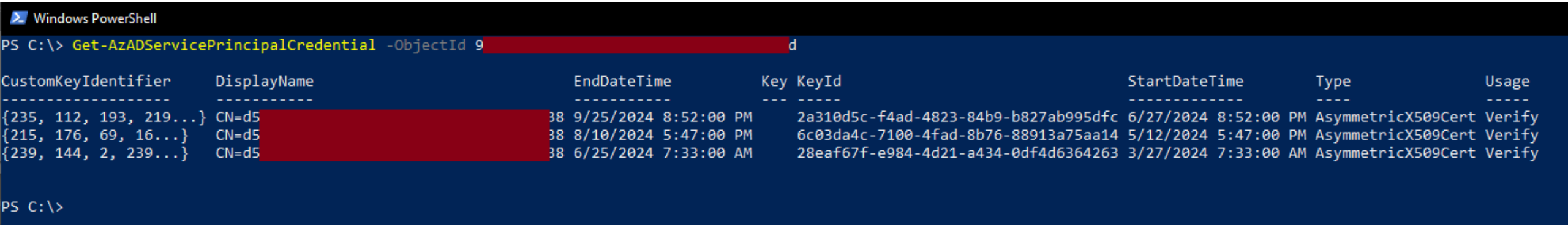
Supported by a Wide Variety of Services

- Virtual Machines, App Services, AKS, ACR, etc.

What are Managed Identities?

Underlying Credential Mechanism

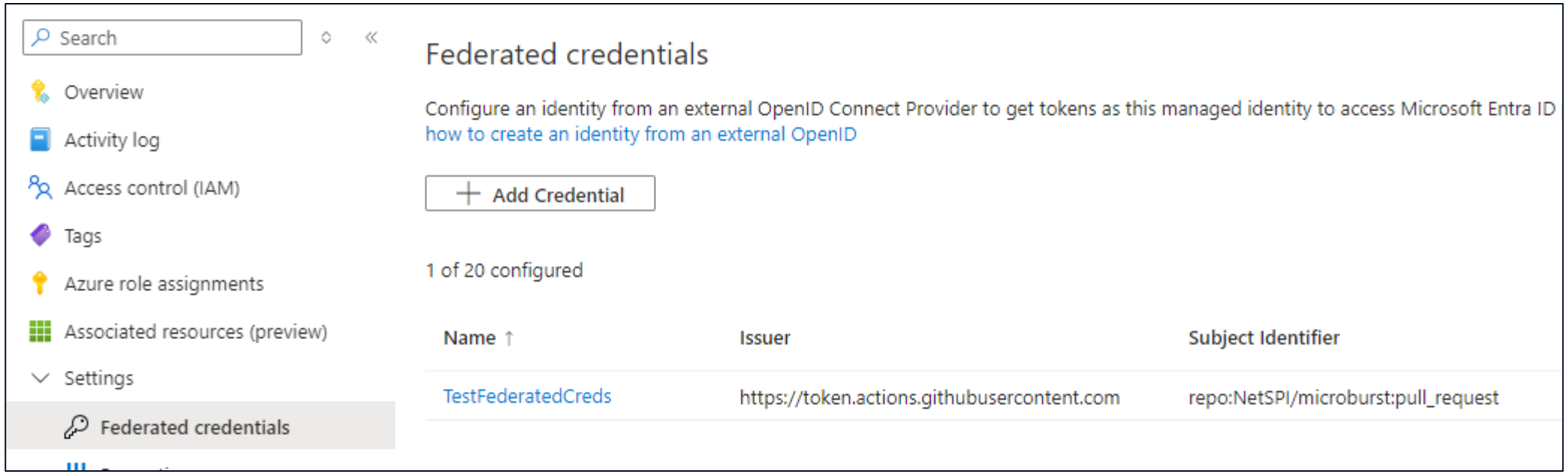
- Azure Service Principal
  - Client/Application ID
    - App Registration is visible in the APIs, not in the Portal
    - **Get-AzADServicePrincipalCredential –ObjectId $OBJ_ID**
  - Certificate
    - User-Assigned – 3-month expiration
    - Users are never supposed to have access to these credentials



What are Managed Identities?

Underlying Credential Mechanism

- Azure Service Principal
  - Federated Credentials
    - A method for authenticating as the User-Assigned Managed Identity outside of Azure
    - See roadoidc in ROADtools by Dirk-jan Mollema



Original research: Rogier Dijkman & Dirk-jan Mollema

What are Managed Identities?

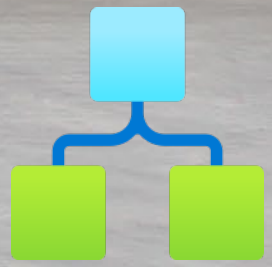
Temporary Credential Mechanism

- Token services
  - IMDS and Side Car Token issuing services – Web Request
  - Passing a User-Assigned Managed Identity to a service (Deployment Scripts)
- Generates a token for a specific scope:
  - https://management.azure.com/
  - https://graph.microsoft.com/
  - https://vault.azure.net
  - Several more options…

Authenticating with Az PowerShell and CLI

- PowerShell Authentication
  - Connect-AzAccount -identity
- CLI Authentication
  - az login --identity

Managed Identity Attacks by Service



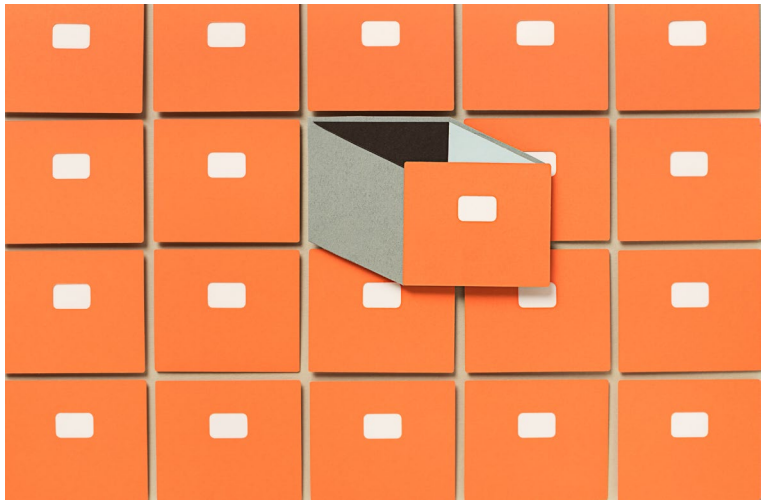
Attacks by Service – Overview

Initial Access Method

- What access do we need to start generating tokens?
- Token Generation Command/Code Examples

Tooling Automation Options

- Tools that can automate the collection



***All examples will focus on System-Assigned Managed Identities**

- User-Assigned will require you to specify a Client or Object ID

Attacks by Service – Virtual Machines*

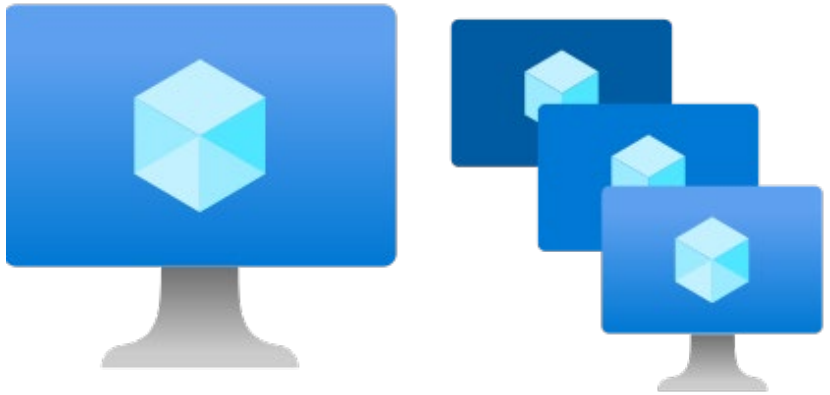
Initial Access Method

- Command Execution
  - Command Injection
  - Execution with Azure Credentials
  - Authenticated Access (RDP, SSH, etc.)
- Requires a “Metadata: true” header
- Windows
  - (Invoke-WebRequest -Uri 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https://management.azure.com/' -Method GET -Headers @{Metadata="true"} -UseBasicParsing).Content
- Linux
  - curl -s 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https://management.azure.com/' --header "Metadata: true"

Tooling Automation Options

- MicroBurst - Invoke-AzVMCommandREST and Invoke-AzVMBulkCMD

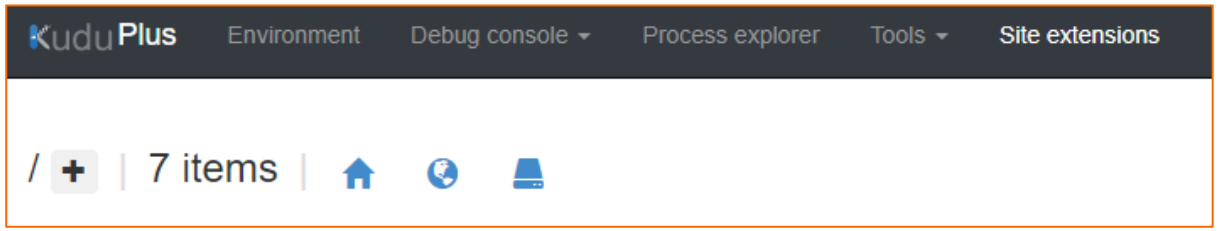
***Inclusive of Virtual Machine Scale Sets (VMSS)**



Attacks by Service – App Services*

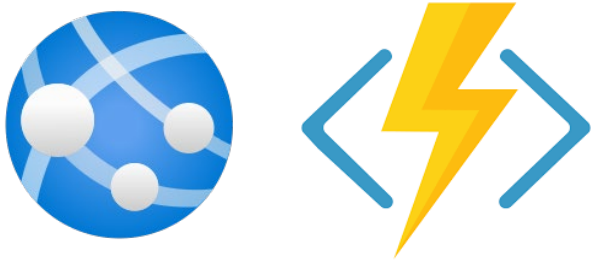
Initial Access Method

- Command Execution
  - Command Injection
  - Execution with Azure Credentials
    - Portal Console/SSH menu
    - Kudu (Advanced Tools) Console
  - Function App Overwrite
- Requires a Header – (secret or X-IDENTITY-HEADER)
- MSI Secret Option:
  - curl "%MSI_ENDPOINT%?resource=https://management.azure.com&api-version=2017-09-01" -H secret:%MSI_SECRET%
- X-IDENTITY-HEADER Option:
  - curl "%IDENTITY_ENDPOINT%?resource=https://management.azure.com&api-version=2019-08-01" -H X-IDENTITY-HEADER:%IDENTITY_HEADER%



Tooling Automation Options

- MicroBurst – Invoke-AzAppServicesKuduDebug and Invoke-AzAppServicesCMD
- FuncoPop (https://github.com/NetSPI/FuncoPop) - Invoke-AzFunctionAppTakeover



***Inclusive of Function Apps**

Attacks by Service – ACR

Initial Access Method

- Task Execution with Azure Credentials
- Create the task that authenticates with an attached identity
  - Output the token in the task output

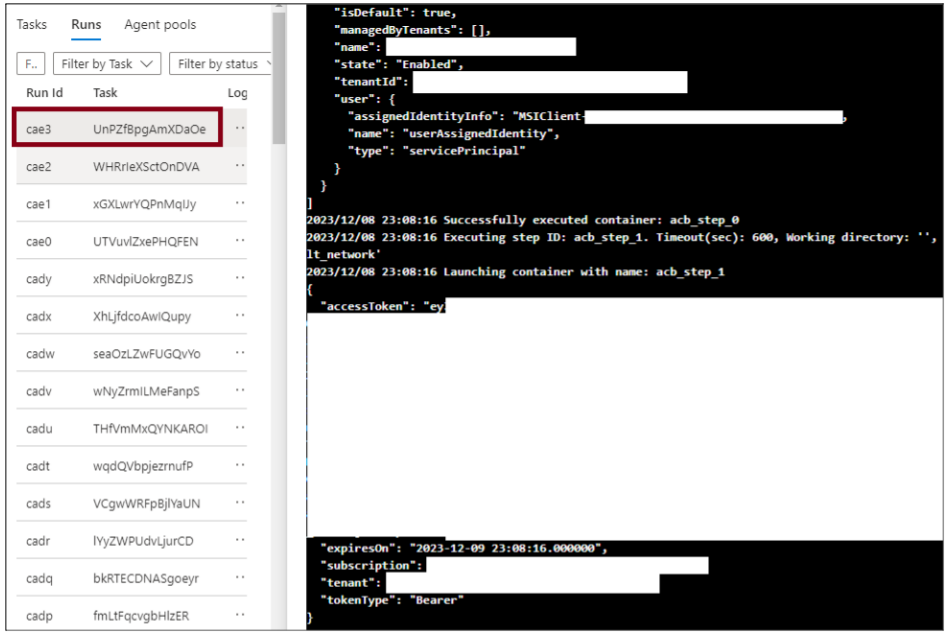
- Task YAML:

```
version: v1.1.0
steps:
  - cmd: az login --identity --allow-no-subscriptions
  - cmd: az account get-access-token --resource=$TokenScope
```

- Run the task
- Remove/Clean up the task

Tooling Automation Options

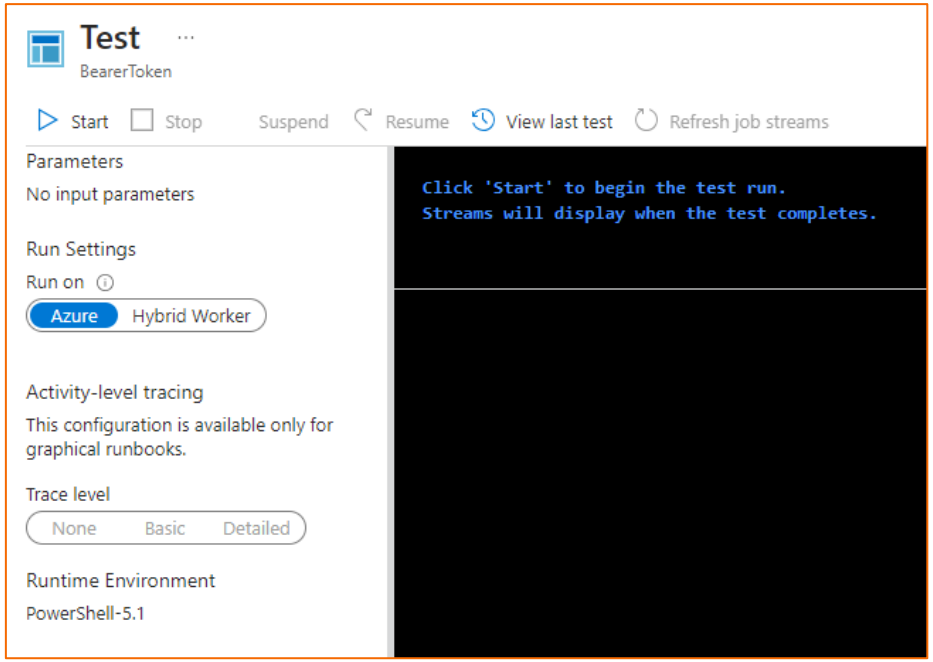
- MicroBurst - Invoke-AzACRTokenGenerator



Attacks by Service – Automation Accounts

Initial Access Method

- Runbook creation/edit and execution with Azure Credentials
- Create/edit the runbook that authenticates with an attached identity
  - Output the token in the task output
    - Run in the test pane
      - Run a second blank job to clear previous output
    - MicroBurst encrypts the output
  - Exfiltrate token via HTTP/DNS/etc.
- Can authenticate with Az PS modules
  - Better to make local HTTP requests with “X-IDENTITY-HEADER” header
- AutoWarp Vulnerability
  - High number ports listening on shared containers with no isolation
  - Could generate tokens for other tenants/subscriptions
- Automation Account Hybrid Workers
  - Query the JRDS from the HW, with a MI token of the HW



Tooling Automation Options

- MicroBurst – Get-AzPasswords
  - Use the “-AutomationAccount Y” Flag

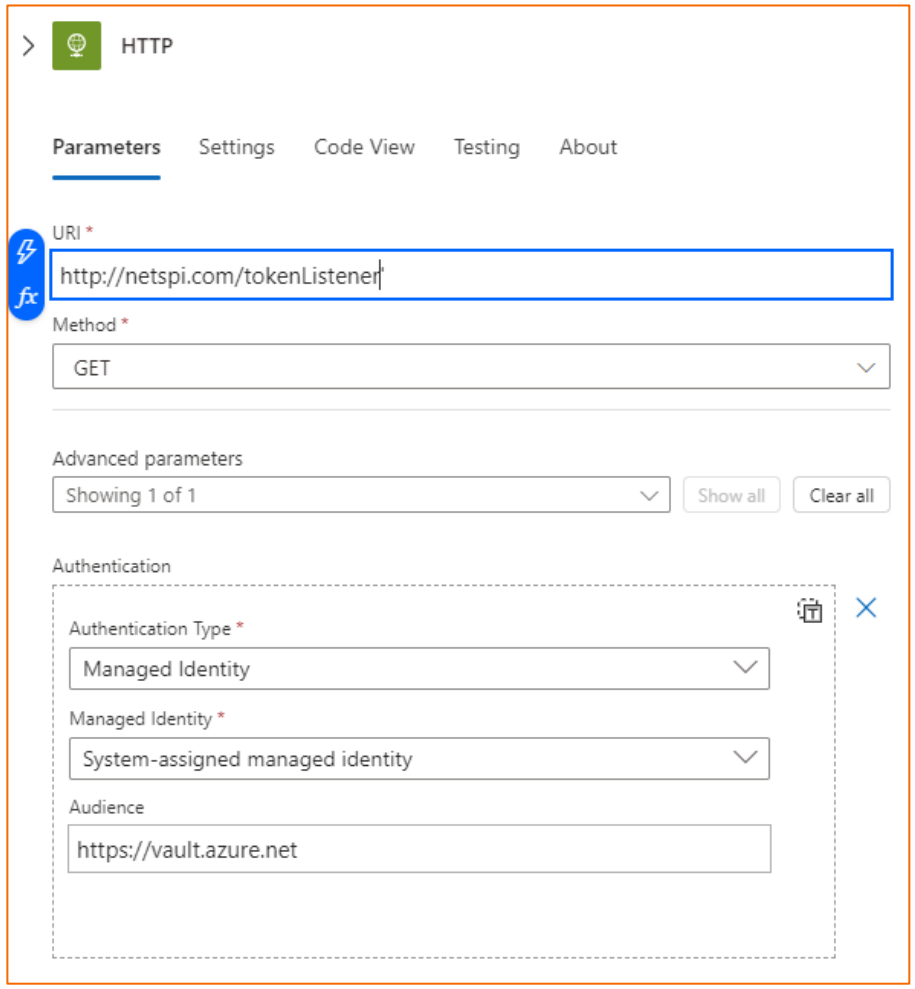
Attacks by Service – Logic Apps

Initial Access Method

- Logic App Modification with Azure Credentials
  - Modify an existing App with an attached Managed Identity
  - Create new app and attach a User-Assigned Managed Identity
- Use the HTTP Request function to send a token to an endpoint
  - Capture the token
  - Revert any changes or delete your new app
  - Set on a schedule trigger for additional persistence
- Backdoor the Function App associated with the Logic App (Standard Plan)
  - Same idea as previously noted in App Services

Tooling Automation Options

- FuncoPop (https://github.com/NetSPI/FuncoPop)
  - Invoke-AzFunctionAppTakeover



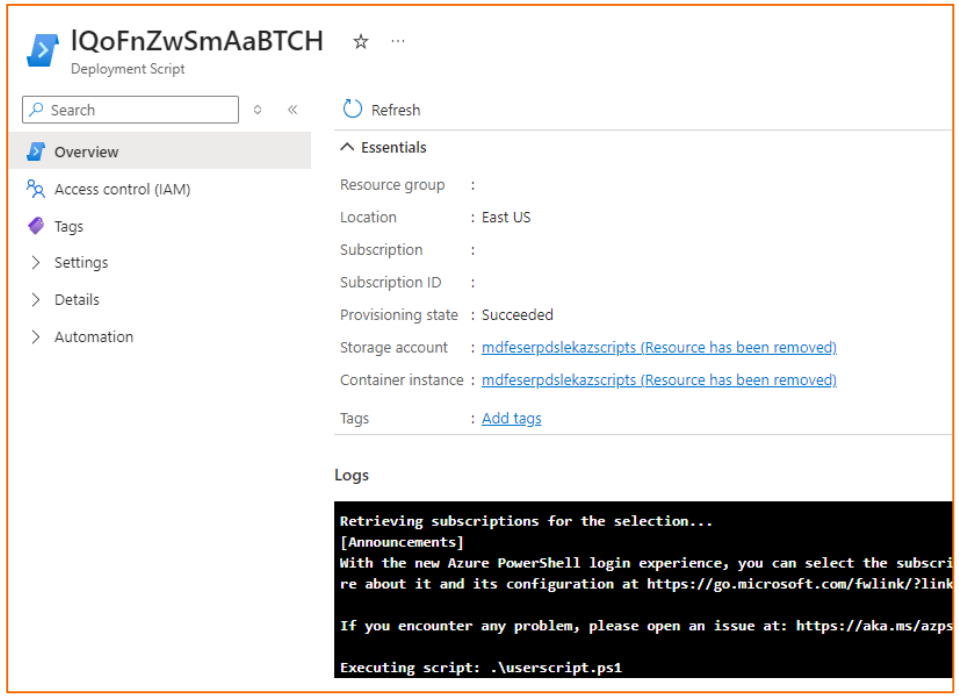
Abusing Deployment Scripts

```
"resources": [
  {
    "type": "Microsoft.Resources/deploymentScripts",
    "apiVersion": "2020-10-01",
    "name": "runPowerShellInlineWithOutput",
    "location": "[resourceGroup().location]",
    "kind": "AzurePowerShell",
    "properties": {
      "forceUpdateTag": "[parameters('utcValue')]",
      "azPowerShellVersion": "8.3",
      "scriptContent": "
        param([string] $name)
        $output = \"Hello {0}\" -f $name
        Write-Output $output
        $DeploymentScriptOutputs = @{}
        $DeploymentScriptOutputs['text'] = $output
      ",
      "arguments": "[concat('-name', ' ', parameters('name'))]",
      "timeout": "PT1H",
      "cleanupPreference": "OnSuccess",
      "retentionInterval": "P1D"
    }
  }
],
```

Azure Deployment Scripts Overview

What They Are

- A method to programmatically run scripts in the context of an Azure Deployment
  - Create a template
    - Works with ARM and Bicep
  - Deploy a resource
  - Run a script to configure the resource
- Can attach a User-Assigned Managed Identity
- Creates a Storage Account and Container Instance



The Attack

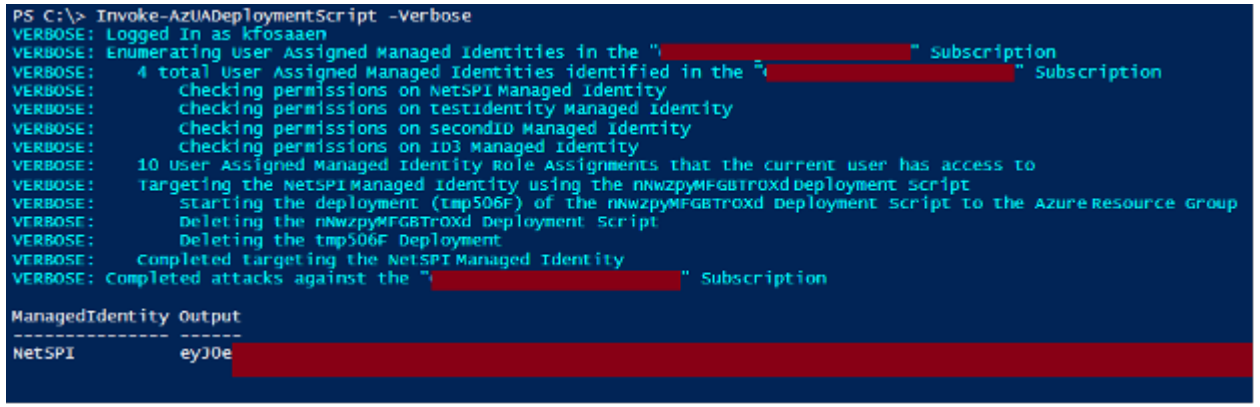
- Target a specific User-Assigned Managed Identity
- Attach it to a Deployment Script template
  - Template includes command to run “Get-AzAccessToken” as the Managed Identity
- Deploy the template and return the command output
- Clean up the resources

Original research: Rogier Dijkman - Project Miaow

Automating the Attack

Attack Overview

- Authenticate via Az PS module
- Run the script - Invoke-AzUADeploymentScript



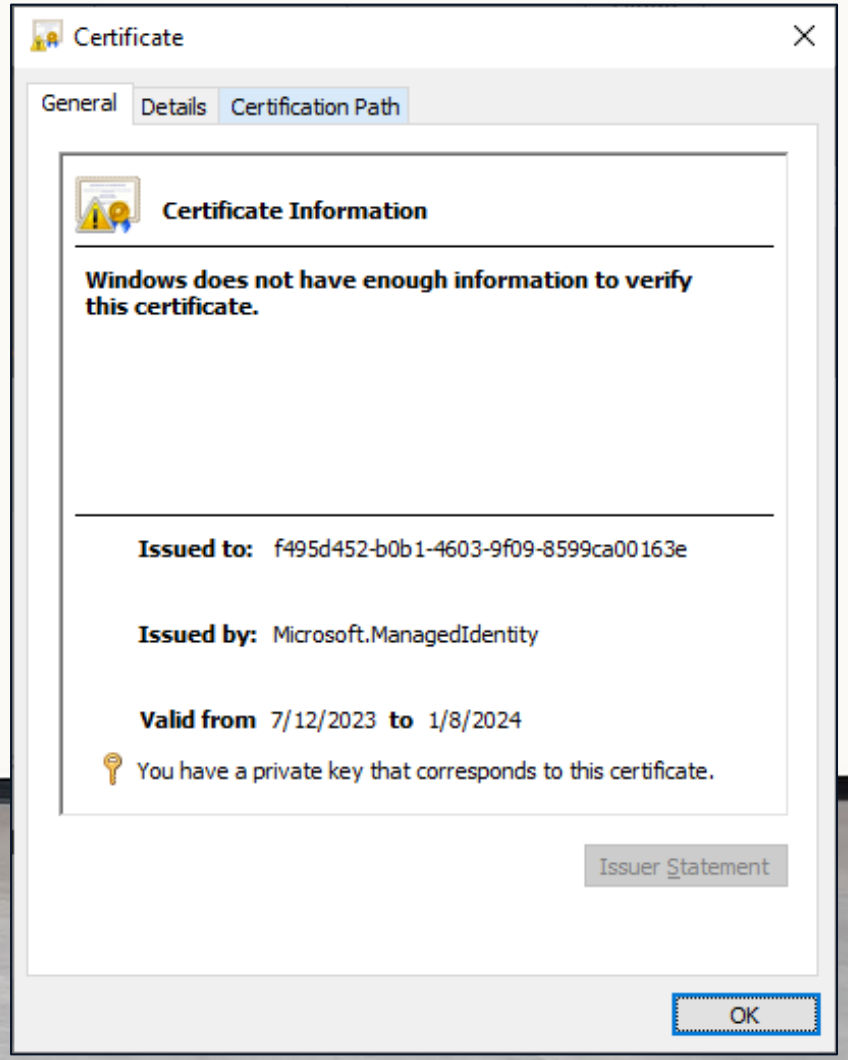
Script Overview

- Select a subscription
- Scan for available User-Assigned Managed Identities
- Select the Identities to attack
- Locally generate an ARM template to use
  - Configured with the MI and a command
  - Default command is *"(Get-AzAccessToken).Token"*
- Deploy the template
  - Creates Deployment Script, Storage Account, Container Instance
- Clean up
  - Deletes Deployment Script, Storage Account, Container Instance

Script Usage

- Token Generation
- Run Az PowerShell commands as the selected User-Assigned Managed Identity

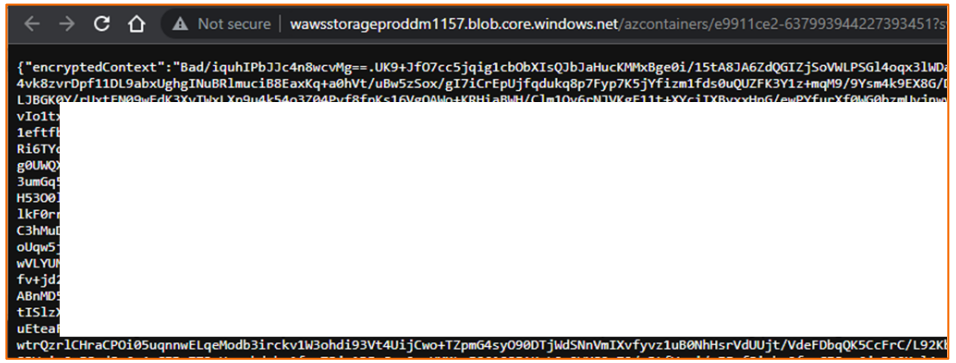
Finding Managed Identity Certificates

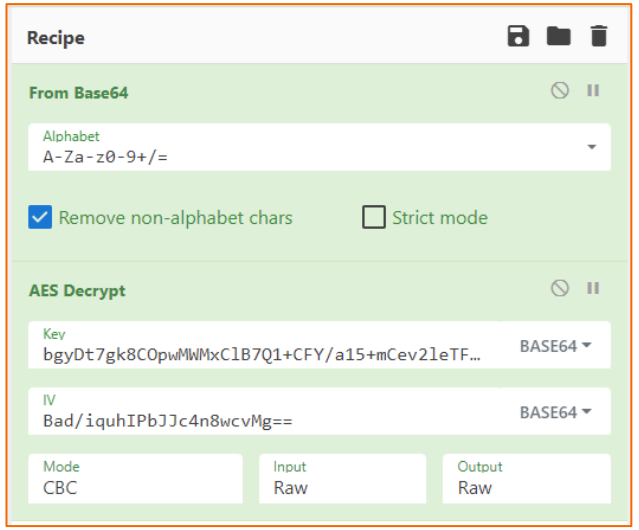


Gaining Access to Managed Identity Certificates

Function App Certificates

- What are Function Apps?
  - APIs as a Service
  - Subset of the App Services resource type
- Accessing the Configuration – Linux Containers
  - Command execution in the container is required
    - Either direct (Azure Mgmt) or indirect (Function SA Overwrite)
  - SAS Token Configuration File (CONTAINER_START_CONTEXT_SAS_URI)
    - EncryptedContext contains data and Initialization Vector (IV)
  - Decrypt the Configuration File
    - Key is in environmental variables
    - Storage Account Connection String
    - Links to Source Code Zip Files:
    - Function App Secrets
    - “MSISpecializationPayload”
      - Certificate returned as Base64 value
      - Applies to User Assigned and System Assigned





- [Publicly Disclosed 11/16/23]

Gaining Access to Managed Identity Certificates

Azure Arc Managed Identity Certificates

- What is Arc?
  - Method for enrolling non-Azure computers into Azure
    - Enables VM Extensions
    - Can enroll systems as Automation Account Hybrid Workers
    - Bridges the cloud and on-prem

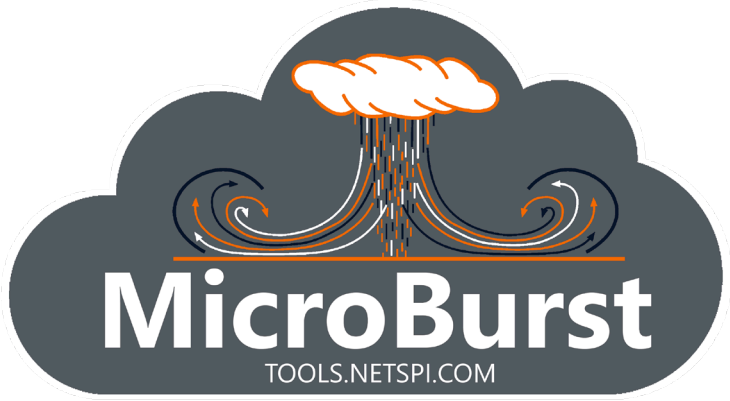
- How to Access the Managed Identity Certificates (as a local admin)
  - Windows Systems
    - C:\ProgramData\AzureConnectedMachineAgent\Certs\myCert.cer
  - Linux Systems
    - /var/opt/azcmagent/certs/myCert
  - Both files are PFX files

- Potential Impact
  - Dependent on permissions assigned to the Managed Identity
  - Allows for persistent access as the Managed Identity

Gaining Access to Managed Identity Certificates

Automating Dumping of Azure Arc Managed Identity Certificates

- The “Get-AzArcCertificates” function in MicroBurst
  - Uses existing Az PowerShell module authentication
  - Prompts for the subscription to attack
  - Enumerates the available Arc systems
  - Prompts for targets
  - Uses the “Run Command” feature to get the certificates:
    - Windows:
      - gc C:\ProgramData\AzureConnectedMachineAgent\Certs\myCert.cer
    - Linux:
      - cat /var/opt/azcmagent/certs/myCert
  - Deletes the Run Command instance
  - Writes the certificate and “AuthenticateAs” script locally



Gaining Access to Managed Identity Certificates

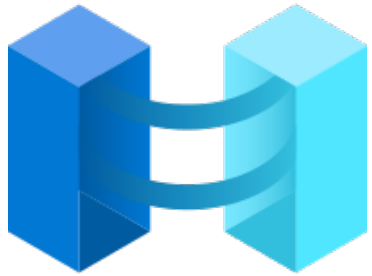
Azure Arc Managed Identity Certificates

- **MSRC Response**

  Thanks for sharing the information. Upon investigation, our teams determined that this is working as expected and is not a vulnerability for servicing.

  This is because to obtain the certificate, an attacker must be local admin on an enrolled device. Certificates used to authenticate endpoints will always be obtainable from at least memory on the endpoint.

- **Publicly Disclosed at DEF CON 32 Cloud Village – 8/10/24**



Abusing Tokens and Certificates

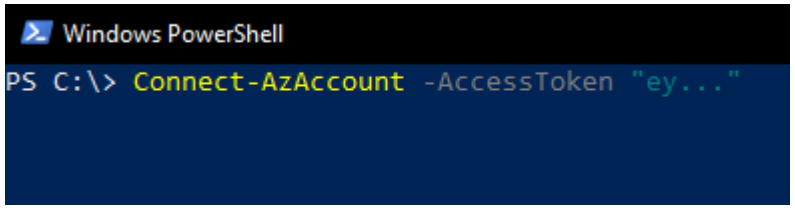
Abusing Tokens and Certificates

Authenticating as the Managed Identity

- **Token**
  - Az PowerShell
  - REST APIs
- **Certificate**
  - Az PowerShell
  - Az CLI

Dumping Data

- **MicroBurst**
- **PowerZure**
- **Etc.**




```
PS C:\temp\ARC> .\AuthenticateAs-3444a6fd-92f1-4d82-96a1-c4e2bed969c9.ps1


   PSParentPath: Microsoft.PowerShell.Security\Certificate::LocalMachine\My

Thumbprint                                Subject
----------                                -------
528175BFAEF2A6AF9ECF9765A7CAA9E9C67B614A  CN=c72455ae-3be7-487e-af60-ea165cdad717
Retrieving subscriptions for the selection...

Environments : {[AzureChinaCloud, AzureChinaCloud], [AzureCloud, AzureCloud], [AzureUSGovernment, AzureUSGovernment]}
Context      : Microsoft.Azure.Commands.Profile.Models.Core.PSAzureContext

[Announcements]
With the new Azure PowerShell login experience, you can select the subscription you want to use more easily. Learn more

If you encounter any problem, please open an issue at: https://aka.ms/azpsissue

PS C:\temp\ARC> Get-AzContext


   Tenant: [redacted]

SubscriptionName    SubscriptionId                    Account                           Environment
----------------    --------------                    -------                           -----------
Research            0[redacted]7 c72455ae-3be7-487e-af60-ea165cdad717 AzureCloud
```

Conclusions & Questions

Conclusions & Questions

Conclusions

- Managed Identities are better than cleartext credentials
- They are not without their own flaws
- Want to persist as a Managed Identity?
  - Find an ARC server and grant it a role

Questions?

- Feel free to reach out!
- Where you can find me for further questions:

twitter.com/kfosaaen

linkedin.com/in/karl-fosaaen

- **Slides will be shared on both platforms**

Additional NetSPI References

Azure Privilege Escalation Using Managed Identities

Lateral Movement in Azure App Services

Azure Deployment Scripts: Assuming User-Assigned Managed Identities

Automating Managed Identity Token Extraction in Azure Container Registries

Mistaken Identity: Extracting Managed Identity Credentials from Azure Function Apps

MicroBurst Toolkit

FuncoPop Toolkit

Let’s stay in touch!

Karl Fosaaen

VP of Research

twitter.com/kfosaaen

linkedin.com/in/karl-fosaaen

241 N 5th Ave Suite 1200

Minneapolis, MN 55401

netspi.com

HYBRID IDENTITY PROTECTION conf24
NEW ORLEANS

Thank You!