CWL
CyberWarFare Labs

External Attack Surface for Initial Access in Microsoft Azure Cloud

❖ CyberWarFare Labs

CW Labs is a renowned UK based Ed-tech company specializing in cybersecurity cyber range labs. They provide on-demand educational services and recognize the need for continuous adaptation to evolving threats client requirements.

The company has two primary divisions :

1. **Cyber Range Labs**
2. **Up-Skilling Platform**

About Speaker:

Parth Agrawal
(Security Intern @CWL)

Is a cloud security enthusiast with a keen interest in the intricacies of cloud services offered by AWS, Azure, and GCP. Possessing a comprehensive understanding of these platforms, they are particularly drawn to exploring Red Team methodologies. Interested in Red Team methodologies, focusing on vulnerability testing and detection across external attack surfaces.

Table of Contents

Azure Services

Active Directory, Blob Storage,
Cosmos DB,

Azure Active Directory (AAD)

- Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management service.
- It's designed to help organizations manage user identities and provide secure access to resources, both on-premises and in the cloud.
- Key Features:
  - Single Sign-On (SSO)
  - Multi-Factor Authentication (MFA)
  - Identity Protection
  - Identity Governance
  - Application & Device Management

Blob Storage

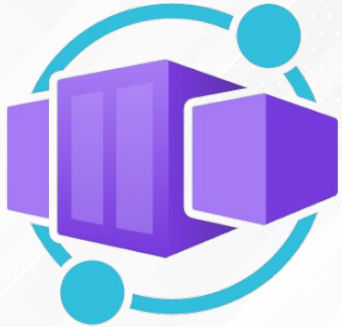
- Azure Blob Storage is a cloud-based object storage service provided by Microsoft Azure.
- Azure Blob Storage offers a highly scalable and durable platform for storing data.
- Key Features:
  - Data Backup and Archive
  - Media and Content Storage
  - Application Data Storage

Cosmos DB Storage

- Azure Cosmos DB is a globally distributed, multi-model database service provided by Microsoft Azure.
- It's designed to enable developers to build highly responsive and scalable applications by offering low-latency access to data across the world.
- Key Features:
  - Multi-Model Support
  - Global Distribution
  - Horizontal Scalability
  - Automatic Indexing
  - Consistency Levels

Container Storage

- Azure Container Storage is a volume-management service built natively for containers.
- Key Features:
  - Fully managed persistent volume deployment
  - Simple and consistent volume orchestration
  - Efficient allocation of persistent volumes into backend storage
  - Rapid scale out of storage containers

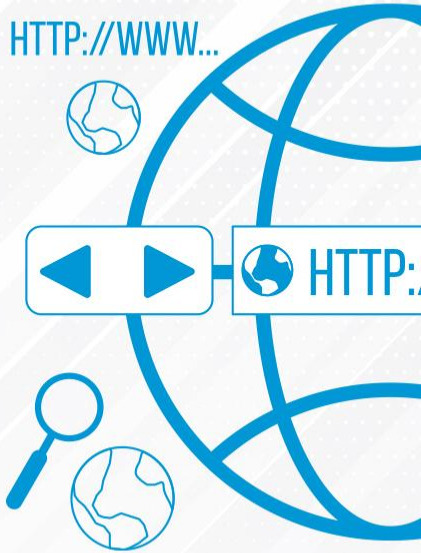


Azure Functions

- Azure Functions is a serverless compute service provided by Microsoft Azure.
- It enables developers to build and deploy event-driven, scalable, and cost-effective applications without worrying about managing underlying infrastructure.
- Key Features:
  - Event-Driven Execution
  - Serverless Execution
  - Support for Multiple Programming Languages
  - Stateful and Stateless Execution
  - Development and Deployment Options

Public URLs

For Available Services



Azure Services	Sample Public URL
Blob Storage	https://<storage_account_name>.blob.core.windows.net OR https://<stg_acc>.blob.core.windows.net/<container_name>?restype=container&comp=list
Azure Data Lake Storage Gen2	https://<storage_account>.dfs.core.windows.net
Queue storage	https://<storage_account>.queue.core.windows.net
Table storage	https://<storage_account>.table.core.windows.net
Azure Files	https://<storage_account>.file.core.windows.net
SQL Database	https://<database_name>.database.windows.net

Azure Services	Sample Public URL
Cosmos DB	`https://<account_name>.documents.azure.com`
App Service	`https://<app_name>.azurewebsites.net`
Cognitive Services	`https://<service_region>.api.cognitive.microsoft.com`
Functions	`https://<function_app_name>.azurewebsites.net/api/<function_name>`
Active Directory	`https://login.microsoftonline.com/<tenant_id>`
Virtual Machines	`https://<vm_name>.<region>.cloudapp.azure.com`
Key Vault	`https://<vault_name>.vault.azure.net`

Scenario 1: OSINT

RECON

Blob Storage Recon

Recon via Shodan:

```
azure-container-name <container_name>
```

```
org:Microsoft ssl.cert.subject.cn:blob
```

```
http.title:"Blob storage"
```

Recon via fofa:

```
header="x-ms-blob-type: BlockBlob"
```

```
cert="*.blob.core.windows.net"
```

Blob Storage Recon

CLI-based Recon:

- Cloud Enum:

```
./cloud_enum.py -k <KEYWORD> --disable-aws --disable-gcp
```

Web-based Recon:

- Bucket search:
  - https://osint.sh/buckets
  - https://buckets.grayhatwarfare.com
  - https://builtwith.com/

Blob Storage Recon

Web-based Recon:

- Dorks:
  - Google Dorks:

```
site:*.blob.core.windows.net inurl:/container_name
```

```
site:*.blob.core.windows.net intext:"confidential" OR
intext:"password"
```

  - GitHub Dorks:

```
filename:*.txt site:gist.github.com
"blob.core.windows.net"
```

```
language:JavaScript "blob.core.windows.net"
```

Cosmos DB Recon

Recon via Shodan:

```
"Azure-CosmosDB"
```

Recon via Censys:

```
Azure Cosmos DB
```

Recon via fofa:

```
title="Azure Cosmos DB"
```

Cosmos DB Recon

Web-based Recon:

- Dorks:
  - Google Dorks:

```
site:*.documents.azure.com
```

```
site:cosmos.azure.com
```

```
intitle:"Azure Cosmos DB" filetype:pdf
```

```
"Azure Cosmos DB" intitle:"Microsoft Azure"
```

```
"Azure Cosmos DB" inurl:forum
```

Cosmos DB Recon

CLI-based Recon:

- Cloud Enum:

```
./cloud_enum.py -k <KEYWORD> --disable-aws --disable-gcp
```

Azure Functions Recon

Recon via Shodan:

```
azure-function-name microsoft
```

```
org:Microsoft ssl.cert.subject.cn:functions
```

Recon via fofa:

```
body="azurewebsites.net" && body="function" && title="test"
```

Azure Functions Recon

Web-based Recon:

- Dorks:
  - Google Dorks:

```
site:*.azurewebsites.net inurl:/function_name
```

```
site:*.azurewebsites.net intext:"Sensitive Information"
                    OR intext:"API key"
```

  - GitHub Dorks:

```
language:JavaScript "azurewebsites.net"
```

```
filename:*.json site:github.com azurewebsites.net
```

Scenario 2: Unauthenticated Enumeration

`Enum`

How to Install MicroBurst Tool

1. `git clone https://github.com/NetSPI/MicroBurst.git`
2. `Move to MicroBurst Folder`
3. `Import-Module .\MicroBurst.psm1`
4. `dir -Recurse .\MicroBurst-master | Unblock-File`

Azure Active Directory Recon

CLI-based Recon:

➢ To retrieve comprehensive information about an Azure tenant using a single command from the AADInternals library

```
Invoke-AADIntReconAsOutsider -DomainName
<Corp_name>.onmicrosoft.com | Format-Table
```

★ **OUTPUT**

```
Tenant brand:       Company Ltd
Tenant name:        company
Tenant id:          1937e3ab-38de-a735-a830-3075ea7e5b39
DesktopSSO enabled: True

Name                          DNS   MX    SPF   Type      STS
----                          ---   --    ---   ----      ---
company.com                  True  True  True  Federated sts.company.com
company.mail.onmicrosoft.com True  True  True  Managed
company.onmicrosoft.com      True  True  True  Managed
int.company.com             False False False  Managed
```

Azure Active Directory Recon

CLI-based Recon:

- To check if a username exists inside a tenant. This includes also guest users, whose username is in the format:

```
<email>#EXT#@<tenant name>.onmicrosoft.com
```

```
Invoke-AADIntUserEnumerationAsOutsider
    -UserName "user@company.com"
```

```
UserName             Exists
--------             ------
user@company.com     True
```

★ **OUTPUT**

Azure Active Directory Recon

CLI-based Recon:

- A text file containing one email address per row can also be used.

```
Get-Content .\users.txt |
Invoke-AADIntUserEnumerationAsOutsider -Method Normal
```

- After discovering the valid usernames you can get info about a user.

```
Get-AADIntLoginInformation -UserName
root@corp.onmicrosoft.com
```

Azure Active Directory Recon

CLI-based Recon:

- After discovering the valid usernames you can get info about a user.

```
Get-AADIntTenanatID -UserName root@corp.onmicrosoft.com
```

Azure Active Directory Recon

Subdomain Enumeration

- Now that we've identified the domains used by the Azure tenant, it's time to search for any exposed Azure services.
- To search for the base domain name (and several variations) across multiple Azure service domains.

```
Invoke-EnumerateAzureSubDomains -Base <keyword> -Verbose
```

★ **OUTPUT**

```
PS C:\Users\haris\OneDrive\Desktop\MicroBurst-master> Invoke-EnumerateAzureSubDomains -Base atomic -Verbose
VERBOSE: Found atomic.scm.azurewebsites.net
VERBOSE: Found api-atomic.scm.azurewebsites.net
VERBOSE: Found atomictest.scm.azurewebsites.net
VERBOSE: Found atomic.onmicrosoft.com
VERBOSE: Found atomicdata.onmicrosoft.com
VERBOSE: Found atomicfinance.onmicrosoft.com
```

Blob Storage Recon

CLI-based Recon:

- Cloud Enum:

```
./cloud_enum.py -k <KEYWORD> --disable-aws --disable-gcp
```

- To discover open storage accounts.

```
Invoke-EnumerateAzureBlobs -Base <corp_name>
https://<corp_common>.blob.core.windows.net/secrets?restype=
container&comp=list
```

Blob Storage Recon

Subdomain Enumeration

```
Invoke-EnumerateAzureBlobs -Base <keyword> -Verbose
```

★ **OUTPUT**

```
PS D:\MicroBurst> Invoke-EnumerateAzureBlobs -Base secure
Found Storage Account - secure.blob.core.windows.net
Found Storage Account -  secureapi.blob.core.windows.net
Found Storage Account -  azuresecure.blob.core.windows.net
Found Storage Account -  clientsecure.blob.core.windows.net
Found Storage Account -  securedata.blob.core.windows.net
Found Storage Account -  securedev.blob.core.windows.net
Found Storage Account -  securefiles.blob.core.windows.net
Found Storage Account -  hrsecure.blob.core.windows.net
Found Storage Account -  secureimages.blob.core.windows.net
```

Cosmos DB Recon

Subdomain Enumeration

```
Invoke-EnumerateAzureSubDomains
-Base <keyword> -Verbose
```

★ **OUTPUT**

```
PS C:\Users\haris\OneDrive\Desktop\MicroBurst-master> Invoke-EnumerateAzureSubDomains -Base atomic -Verbose
VERBOSE: Found atomic.scm.azurewebsites.net
VERBOSE: Found api-atomic.scm.azurewebsites.net
VERBOSE: Found atomictest.scm.azurewebsites.net
VERBOSE: Found atomic.onmicrosoft.com
VERBOSE: Found atomicdata.onmicrosoft.com
VERBOSE: Found atomicfinance.onmicrosoft.com
VERBOSE: Found webatomic.onmicrosoft.com
VERBOSE: Found atomic.database.windows.net
VERBOSE: Found atomicdev.database.windows.net
VERBOSE: Found atomictest.database.windows.net
VERBOSE: Found atomic.mail.protection.outlook.com
VERBOSE: Found atomicdata.mail.protection.outlook.com
VERBOSE: Found atomicfinance.mail.protection.outlook.com
VERBOSE: Found webatomic.mail.protection.outlook.com
VERBOSE: Found atomic.queue.core.windows.net
VERBOSE: Found atomicfiles.queue.core.windows.net
VERBOSE: Found atomicstorage.queue.core.windows.net
VERBOSE: Found atomic.blob.core.windows.net
VERBOSE: Found atomicfiles.blob.core.windows.net
VERBOSE: Found atomicstorage.blob.core.windows.net
VERBOSE: Found atomic.file.core.windows.net
VERBOSE: Found atomicfiles.file.core.windows.net
VERBOSE: Found atomicstorage.file.core.windows.net
VERBOSE: Found atomic.vault.azure.net
VERBOSE: Found atomic.table.core.windows.net
VERBOSE: Found atomicfiles.table.core.windows.net
VERBOSE: Found atomicstorage.table.core.windows.net
VERBOSE: Found atomic.azurewebsites.net
VERBOSE: Found api-atomic.azurewebsites.net
VERBOSE: Found atomictest.azurewebsites.net
VERBOSE: Found atomic.documents.azure.com
```

Container Storage Recon

CLI-based Recon:

➢ To describe the identifier for each storage account available in the current Azure subscription.

```
az storage account list --query '[*].name'
```

★ **OUTPUT**

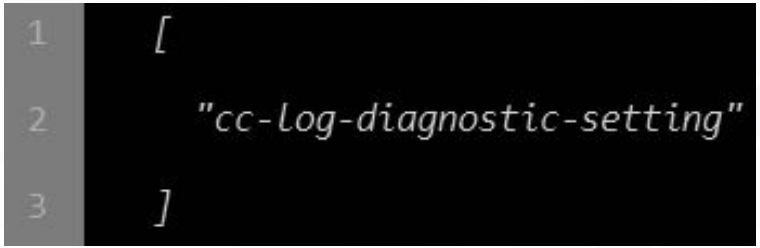
Container Storage Recon

CLI-based Recon:

➢ To describe the name of each diagnostic setting created for the selected Azure subscription.

```
az monitor diagnostic-settings subscription list
 --subscription abcdabcd-1234-abcd-1234-abcd1234abcd
                --query 'value[*].name'
```

★ **OUTPUT**



Container Storage Recon

CLI-based Recon:

➢ To get the ID of the Azure storage account configured to store activity logs within the selected subscription.

```
az monitor diagnostic-settings subscription show
--name "cc-log-diagnostic-setting" --query 'storageAccountId'
```

★ **OUTPUT**

```
1  "/subscriptions/abcdabcd-1234-abcd-1234-abcd1234abcd/....
```

➔ The command output returns the full ID of the associated storage account (the ID contains the storage account name).

Container Storage Recon

CLI-based Recon:

➢ To describe the public access level set for the selected container.

```
az storage container show --account-name abcd1234abcdabcd1234abcd
--name insights-operational-logs --query 'properties.publicAccess'
```

★ **OUTPUT**

➔ **“Container”** means the storage container that holds your activity log files is publicly accessible.

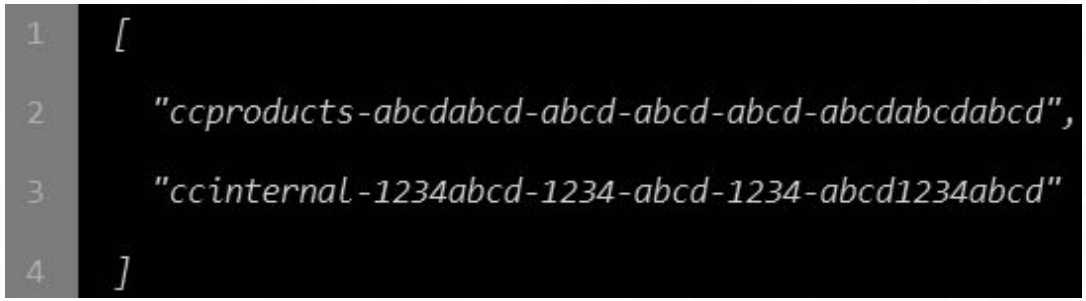
Container Storage Recon

CLI-based Recon:

➢ To list the containers available in the selected storage account

```
az storage container list --account-name
abcdabcdabcd123412341234 --query '[*].name'
```

★ **OUTPUT**



```
[
  "ccproducts-abcdabcd-abcd-abcd-abcd-abcdabcdabcd",
  "ccinternal-1234abcd-1234-abcd-1234-abcd1234abcd"
]
```

Container Storage Recon

CLI-based Recon:

➢ To describe the stored access policies for the selected container.

```
az storage container policy list
--account-name abcdabcdabcd123412341234
--container-name
ccproducts-abcdabcd-abcd-abcd-abcd-abcda
bcdabcd
```

➔ If the storage container policy list command output returns not **empty object(i.e {})**, the Stored Access Policies is enabled and not expired, but the verified stored access policy has **“racwdl” - full access.**

★ **OUTPUT**

Azure Functions Recon

CLI-based Recon:

➢ To identify any publicly exposed Azure function.

```
az functionapp list --output table
--query '[*].{name:name, resourceGroup:resourceGroup}'
```

★ **OUTPUT**

```
Name                    ResourceGroup
----------------------  ----------------------------
cc-main-function-app    cloud-shell-storage-westeurope
cc-project5-function-app  cloud-shell-storage-westeurope
```

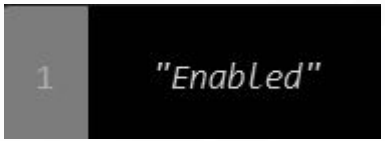
Azure Functions Recon

CLI-based Recon:

➢ To identify any publicly exposed Azure function.

```
az functionapp show  --name cc-main-function-app
 --resource-group cloud-shell-storage-westeurope
         --query 'publicNetworkAccess'
```

★ **OUTPUT**



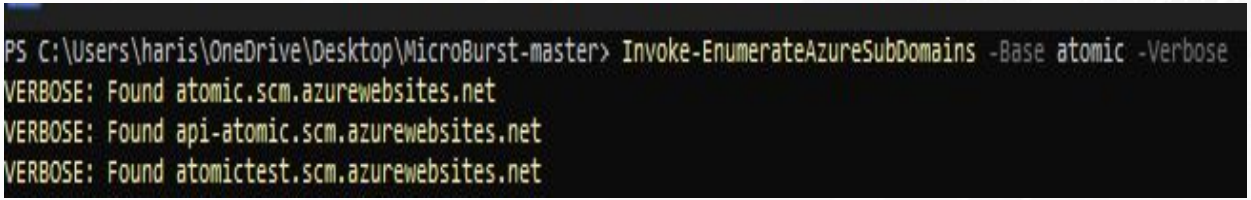
➔ **“Enabled”** means the functions managed with the selected Microsoft Azure Function App are configured to allow public network access.

Azure Function Recon

Subdomain Enumeration

```
Invoke-EnumerateAzureSubDomains -Base <keyword> -Verbose
```

★ **OUTPUT**



MCRTA Certification

AWS | Azure | GCP

➢ **Who can opt for it**

- ➔ Cyber Security Beginners / Professionals
- ➔ Security Analysts / Security Consultants / Security Engineers
- ➔ Anyone Interested in Cloud Security / Cloud Pentesting / Cloud Red Teaming Domains

Thank You

For Professional Red Team / Blue Team / Purple Team / Cloud Cyber Range labs / Trainings please contact

support@cyberwarfare.live

To know more about our offerings, please visit: **https://cyberwarfare.live**