



0365 ...
Pitting the Theory Against the
Practice

64th TF-CSIRT



Once upon a time ...

- ... a company gets a report from one of its customer
- Customer received an email
 - “Hey our bank account info changed”
 - “New info in attachment”
- Email analysis:
 - Mail from: a foreign domain
 - From: spoofed, someone the victim used to exchange with
 - Body: usual phrasing, company signature
 - Attachment: company template
- How could this happen ?
 - BEC, despite MFA
 - Malicious application registered, with too much permissions
 - Guest access abused
 - Anti-spoofing policy bypassed
 - New inbox rule created, forwarding to attacker
- Attacker accessed other things ?
- How to lock out the attacker ?

This talk

- The surface: attacker getting in
 - Usual one: web GUI (admin consoles, Teams, SharePoint/OneDrive, Office online, Webmail,)
 - But several new entry points
 - You said MFA ?
- The complexity: sealing the holes (at least trying `_(\`)/`)
 - “Click here, click there and you are done” Really ?
 - Configuration everywhere, redundancy, overwriting ?
 - Logs: what is what
 - Documentation
- Incident Response: what happened
 - Collecting/Parsing/Understanding logs
 - Focus on sign-ins logs and email transactions (Message Trace Report)
 - O365 activity
 - Azure activity
 - The compromised user is disabled ... Really ?

The Exposed Surface ... Web Services



Name	Description
Azure ATP	Identify, detect, and investigate advanced threats, compromised identities, and malicious insider actions directed at your organization.
Azure Active Directory	Go deep with identity management. Enable multi-factor authentication, self-service password reset, and edit company branding.
Compliance	Manage your compliance needs using integrated solutions for data governance, encryption, access control, eDiscovery, and more.
Endpoint Manager	A single management experience for the End User Computing team in IT to ensure employees' Microsoft 365 devices and apps are secured, managed, and current.
Exchange	Manage advanced email settings, such as quarantine, encryption, and mail flow rules.
Power Automate	Manage the automation of repetitive and time-consuming tasks in the Power Platform admin center, where you can set up connections to web services, files, or cloud-based data and put them to work.
Office configuration	Manage, configure, and monitor deployment of Microsoft 365 Apps for your organization.
Search & intelligence	Manage Microsoft Search settings including services and content that are available for people in your organization. Make finding internal tools, documents, and people just as easy as searching the web in Bing.
Stream	Choose how Microsoft Stream works for your organization.
OneDrive	Control access and sharing settings, default storage, and allowed file types.
Power Apps	Use the Power Platform admin center to manage activity, licenses, and policies for user-generated Power Apps, which can connect to your data and work across web and mobile.
Power BI	This admin center enables Power BI service admins to manage a Power BI tenant for your organization. The portal includes items such as usage metrics and settings.
Security	Get visibility into your security state, investigate and protect against threats, get recommendations on how to increase your security, and more.
SharePoint	Manage site collections, list and library permissions, file storage and sharing.
Dynamics 365 Apps	Use the Dynamics 365 admin center to manage your environment, manage capacity, monitor usage and perform other admin operations.
Teams	Configure messaging, conferencing, and external communication options for your users.
Yammer	Manage your Yammer network, set a usage policy, control external network settings, and enable features like translation.

The Exposed Surface ... and also

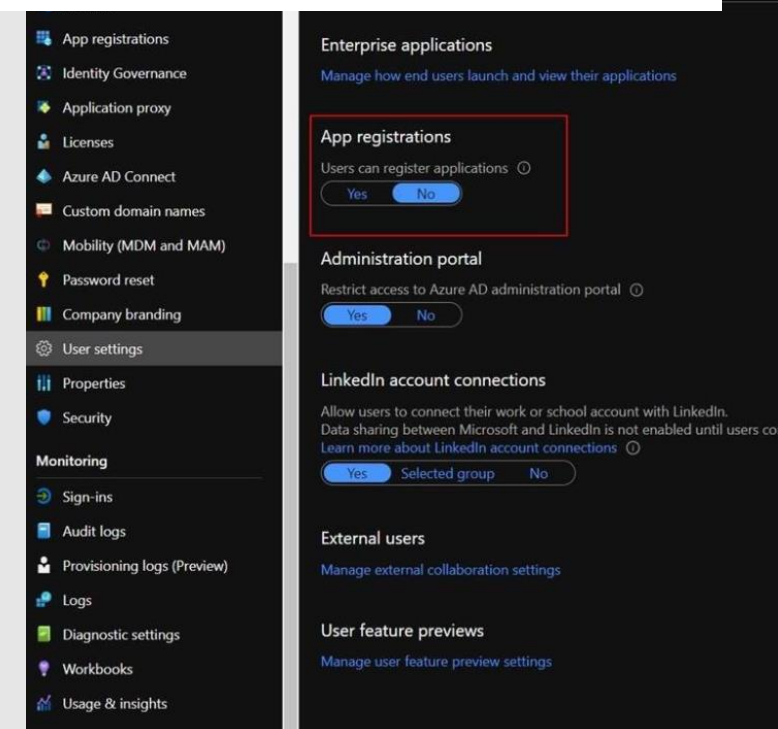
- Guest/Partner: can **access** your SharePoint and Teams
- Application registered (OAuth2) based on **user consent**
- Azure: cloud computing (VM, storage, DB, ...) **exposed** by default
- Authentication protocols
 - Modern (OAuth2): support MFA
 - Legacy:
 - does not support MFA (**MFA policy bypassed**)
 - IMAP, POP, SMTP, ActiveSync, MAPI, EWS
 - What and how to:
<https://docs.microsoft.com/en-us/exchange/clients-and-mobile-in-exchange-online/disable-basic-authentication-in-exchange-online>

Sharing

When this setting is selected, all users can add people outside the organization as guests, so they appear on the Guest users page. When this setting isn't selected, only admins can add guests. [Learn more about guests in your organization.](#)

You can also [change the external sharing settings for SharePoint.](#)

☒ Let users add new guests to the organization



Data Management

- Sharing
 - user can share **links** with external user
 - to company SharePoint or to their OneDrive
- Add-In
 - Can be installed by users in Office online, Teams
 - **Additional applications** (Evernote, Zoom, ...)
 - Nothing in logs !
- Recycle bin
 - **nothing deleted**
 - until applied on the 2nd recycle bin

The image shows two screenshots from the Microsoft Office 365 ecosystem.

The top screenshot is the 'Apps' page in the Office 365 portal. It features a search bar and a sidebar with categories like 'Home', 'Featured', 'Categories', 'App features', and 'Utilities'. The main area displays a grid of various third-party and Microsoft apps, including Sailpoint, Additio, Commander, School Day Wellbeing, Zeplin, Piazza, Group Address Book - Next..., Wallboard, Flowdoh, TagMyFav, Mapwize, Alvao Service Desk, AtBot, Nikabot, Contacts by InfraCom, and Smartflo.

The bottom screenshot shows the 'Corbeille' (Recycle Bin) interface in SharePoint. The left sidebar has a menu with 'Conversations', 'Documents', 'Bloc-notes', 'Pages', 'Contenu du site', 'Corbeille' (highlighted with a red box), and 'Modifier'. The main area is titled 'Corbeille' and contains a table with columns: 'Nom', 'Date de suppression', 'Supprimé par', 'Créé par', and 'Emplacement d'origine'. At the bottom, there is a red box around the text 'Revenir à l'affichage standard de SharePoint' and another red box around the text 'Vous ne trouvez pas ce que vous recherchez ? Regardez dans la Corbeille secondaire'.

The Complexity

- Admin interfaces
 - More than **15 consoles**
 - Admin center, AAD portal, compliance center, security center,
 - OneDrive admin, Teams admin, exchange admin, ...
- Licensing
 - Impact **log retention, features** (Powershell cmdlets, Identity Protection, policies, ...)
 - E1/E3/E5/P1/P2/Business Premium/M365/O365/D365/....
- Configuration
 - **Conditional Access** in multiple locations: overwritten ?
 - User management: **redundancy** between Azure AD and admin center
 - Best practices in official documentation
- Logs
 - Sign-ins, audit logs, activity logs, risky users, risky sign-ins, ...
 - Multiple GUIs + PowerShell to extract: **different results** (limitations, fields, latency)
 - Consoles: ATP, Log analytics, ADX, Cloud App Security, ...
- Documentation
 - A lot on docs.microsoft.com
 - But lots of **embedded links** in pages ... quickly 10 pages opened for a simple question
 - difficult to find **clear information** (fields meaning, options, how to configure, ...)



Forensic: Azure AD sign-ins logs (1/4)

- What: all logins (O365 apps, admin consoles, Azure AD)
- 4 types
 - Interactive: performed explicitly by the user
 - Non-interactive: performed by an application on behalf the user
 - Service Principals: performed by non-user account
 - Managed: performed by resource that have their secrets managed by AAD
 - More details: <https://docs.microsoft.com/en-us/azure/active-directory/reports-monitoring/concept-all-sign-ins>
- Caveats
 - For **small tier**: 1 week retention, no PowerShell cmdlet
 - Anyway, max **1 month** retention
 - From GUI: 1 day latency
- With powershell (no latency):

```
Import-Module DFIR-O365RC

$enddate = get-date
$startdate = $enddate.adddays(-30)
Get-AADLogs -StartDate $startdate -Enddate $enddate
```


- Log export
 - GUI: **4 csv** with logins + 4 csv “auth details”
 - PowerShell (if available)
 - Authentication **with details** in each object
 - **All in one** with DFIR-O365RC module
- Meaningful fields (Json naming)
 - CreatedDateTime
 - UserPrincipalName (email)
 - IPAddress
 - Location
 - Status.ErrorCode (0=success)
 - **AppId/AppDisplayName** = “from where the attacker logged in?”
 - **ResourceId/ResourceDisplayName** “on what the attacker logged in?”
 - ClientAppUsed
 - DeviceDetails (OS, browser version, ...)
 - AppliedConditionalAccessPolicies
 - AuthenticationProcessingDetails (factors, token, ...)

```
[
  {
    "Id": "e18f67d4-c6b8-49e3-bbff-9b31133dbe00",
    "CreatedDateTime": "2021-08-02T13:40:48Z",
    "UserDisplayName": "MOD Administrator",
    "UserPrincipalName": "admin@m365x497090.onmicrosoft.com",
    "UserId": "21472bf1-a44c-4ef0-90b6-d0c5ec2e39b8",
    "AppId": "89bee1f7-5e6e-4d8a-9f3d-ecd601259da7",
    "AppDisplayName": "Office365 Shell WCSS-Client",
    "IpAddress": "217.31.74.130",
    "ClientAppUsed": "Browser",
    "CorrelationId": "3b8b3515-20ac-4ed5-ac88-796ac0366051",
    "ConditionalAccessStatus": "notApplied",
    "OriginalRequestId": "",
    "IsInteractive": true,
    "TokenIssuerName": "",
    "TokenIssuerType": "AzureAD",
    "ProcessingTimeInMilliseconds": 58,
    "RiskDetail": "none",
    "RiskLevelAggregated": "none",
    "RiskLevelDuringSignIn": "none",
    "RiskState": "none",
    "RiskEventTypes": [
      ],
    "ResourceDisplayName": "Microsoft Graph",
    "ResourceId": "00000003-0000-0000-c000-000000000000",
    "AuthenticationMethodsUsed": [
      ],
    "Status": {
      "ErrorCode": 0,
      "FailureReason": "Other.",
      "AdditionalDetails": null
    }
  },

```

Forensic: Azure AD sign-ins logs (3/4)

- “Fun” Fact 1
 - I logged in to 1 of the web interface
 - Can you guess which one ? 😊
- Caveats with web logins
 - 1 login = **multiple lines** in interactive + non-interactive
 - Span over few seconds
 - Browser version: not always the one of the user
 - OS version: not always the one the user
- Other logins
 - Outlook client: ok, 1 line
 - Webmail: ok, 1 line (Exchange)
 - PowerShell: ok, application identifiable

```
interactive:
"2021-08-02T13:40:48Z", "Office365 Shell WCSS-Client", "Microsoft Graph"
"2021-08-02T13:40:48Z", "Office365 Shell WCSS-Client", "Office365 Shell WCSS-Server"
"2021-08-02T13:40:47Z", "Office365 Shell WCSS-Client", ""
"2021-08-02T13:40:44Z", "Microsoft Office 365 Portal", "Windows Azure Active Directory"
"2021-08-02T13:40:39Z", "Microsoft Office 365 Portal", "Windows Azure Active Directory"
"2021-08-02T13:38:45Z", "Office365 Shell WCSS-Client", "Office365 Shell WCSS-Server"
"2021-08-02T13:38:44Z", "Office365 Shell WCSS-Client", "Office 365 Exchange Online"

non interactive:
"2021-08-02T13:40:49Z", "M365 Admin Services", "Office 365 Exchange Online"
"2021-08-02T13:40:49Z", "M365 Admin Services", "Windows Azure Active Directory"
"2021-08-02T13:40:49Z", "M365 Admin Services", "Microsoft Graph"
"2021-08-02T13:40:49Z", "Microsoft Office 365 Portal", "M365 Admin Services"
"2021-08-02T13:40:48Z", "My Apps", "AAD App Management"
"2021-08-02T13:40:48Z", "My Apps", "Windows Azure Active Directory"
"2021-08-02T13:40:48Z", "Office365 Shell WCSS-Server", "PowerApps Service"
"2021-08-02T13:40:48Z", "Office365 Shell WCSS-Server", "My Apps"
"2021-08-02T13:40:48Z", "Microsoft Office 365 Portal", "Microsoft password reset service"
"2021-08-02T13:40:47Z", "Microsoft Office 365 Portal", "Skype and Teams Tenant Admin API"
"2021-08-02T13:40:47Z", "Microsoft Office 365 Portal", "Office 365 Reports"
"2021-08-02T13:40:47Z", "Microsoft Office 365 Portal", "Office 365 Reports"
"2021-08-02T13:40:47Z", "Microsoft Office 365 Portal", "Office 365 Exchange Online"
"2021-08-02T13:40:47Z", "Microsoft Office 365 Portal", "Microsoft Graph"
"2021-08-02T13:40:46Z", "Microsoft Office 365 Portal", "Microsoft Office 365 Portal"
"2021-08-02T13:40:46Z", "Microsoft Office 365 Portal", "Microsoft Office 365 Portal"
"2021-08-02T13:40:46Z", "Microsoft Office 365 Portal", "Microsoft Office 365 Portal"
"2021-08-02T13:40:46Z", "Microsoft Office 365 Portal", "Windows Store for Business"
"2021-08-02T13:40:44Z", "Microsoft Office 365 Portal", "Windows Azure Active Directory"
"2021-08-02T13:40:44Z", "Microsoft Office 365 Portal", "Microsoft Office 365 Portal"
"2021-08-02T13:38:46Z", "Microsoft Office 365 Portal", "Windows Azure Active Directory"
"2021-08-02T13:38:46Z", "Microsoft Office 365 Portal", "Microsoft Office 365 Portal"
"2021-08-02T13:38:46Z", "Microsoft Office 365 Portal", "Microsoft Office 365 Portal"
```

Forensic: Azure AD sign-ins logs (4/4)

- “Fun” Fact 2
 - **Applications IDs**
 - Rogue applications might have nice name
- What is a registered application
 - Internal or external application
 - For which an admin or user gave **consent**
 - **Permissions**: access data or act on behalf the user (defined in the consent request popup, OAuth scope)
 - Can be listed using PowerShell (see later)
- Caveats
 - At Excellium, 635 applications (!! mainly Microsoft stuff)
 - 1 week of sign-ins: 147 distinct AppId
 - ... only 91 application IDs recognized !
 - Opened a ticket for the 56 remaining: **not documented, and will not be**



Forensic: Azure AD audit logs

- What: tenant **management**, user/group CRUD, admin operations
- Caveats
 - For **small tier**: 1 week retention, no PowerShell cmdlet
 - Anyway, max **1 month** retention
 - From GUI: 1 day latency
- With powershell (no latency):
- Log content: straightforward

```
Import-Module DFIR-O365RC

$enddate = get-date
$startdate = $enddate.adddays(-30)
Get-AADLogs -StartDate $startdate -EndDate $enddate
```

Forensic: Collect applications registered (1/2)

- Reason 1

- To **interpret** sign-ins logs (well ... now you know ... not exhaustive !)
- How:
 - 3/4 Screenshots from GUI 😊
 - For the tenant:

```
Import-Module AzureADPreview
Connect-AzureAD
```

```
Get-AzureADServicePrincipal -All:$true | ConvertTo-Json | Out-File -Encoding utf8 -FilePath AllApplications.json
```

- “First party”: <https://docs.microsoft.com/en-us/troubleshoot/azure/active-directory/verify-first-party-apps-sign-in>

- Reason 2

- Identify **consent given** to rogue application, and associated **permissions**
- Subset of Azure audit logs (hence, same limitations)
- How:

- Collect consents given:
- List permissions:
<https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/detect-and-remediate-illicit-consent-grants?view=o365-worldwide>

```
Import-Module DFIR-O365RC
```

```
$enddate = get-date
```

```
$startdate = $enddate.adddays(-30)
```

```
Get-AADApps -StartDate $startdate -Enddate $enddate
```

Forensic: Applications consent given (2/2)

- Caveats:
 - **Parsing** ... well ...
 - **IpAddress** always empty in my tests
 - **Permissions** granted not listed

```
{
  "Id": "Directory_564a0c64-1a12-4366-9aa5-b4fc9b715037_QW092_110411038",
  "Category": "ApplicationManagement",
  "CorrelationId": "564a0c64-1a12-4366-9aa5-b4fc9b715037",
  "Result": "success",
  "ResultReason": "",
  "ActivityDisplayName": "Consent to application",
  "ActivityDateTime": "/Date(1626858159447)/",
  "LoggedByService": "Core Directory",
  "OperationType": "Assign",
  "InitiatedBy": {
    "User": "class InitiatedByUser {\n  Id: 21472bf1-a44c-4ef0-90b6-d0c5ec2e39b8\n  DisplayName: \n  IpAddress: \n  UserPrincipalName: admin@M365x497090.onmicrosoft.com\n  App: null\n}",
    "App": null
  },
  "TargetResources": [
    "class TargetResource {\n  Id: 814a3e87-085f-40d9-a157-48aa4bb3937f\n  DisplayName: Pickit App\n  Type: ServicePrincipal\n  UserPrincipalName: \n  GroupType: \n}"
  ],
  "AdditionalDetails": [
    "class AdditionalDetail {\n  Key: User-Agent\n  Value: EvoSTS\n}"
  ]
}
```




Forensic: O365 audit logs (1/2)

- What: activity on applications (Office, Webmail, Teams, OneDrive, ...)
- **Operations** of interest:
 - UserLoggedIn/UserLoginFailed: **sign-ins are better**, but can overcome sign-ins limitations
 - MailItemsAccessed (only license E5 😊, only contain the message ID)
 - Create, Sent, New-InboxRule, Set-InboxRule
 - FileAccessed
 - <https://docs.microsoft.com/en-us/microsoft-365/compliance/search-the-audit-log-in-security-and-compliance?view=o365-worldwide#audited-activities>
- Collect from **GUI**:
 - 1 month max
 - Csv: json in csv, sometimes **corrupted**
- Collect from **PowerShell**:
 - 3 months max
 - With Search-UnifiedAuditLog cmdlet: json escaped in json ... ?!?!
 - With DFIR-O365RC module: json line

```
Import-Module DFIR-O365RC

$enddate = get-date
$startdate = $enddate.adddays(-90)
Search-O365 -StartDate $startdate -Enddate $enddate -UserIds "email1@domain.tld", "email2@domain.tld"
```

```
Import-Module DFIR-O365RC

$enddate = get-date
$startdate = $enddate.adddays(-90)
Get-O365Full -StartDate $startdate -Enddate $enddate -RecordSet "All"
```

Forensic: O365 audit logs (2/2)

- Caveats with **Search-UnifiedAuditLog**
 - “Search-UnifiedAuditLog -StartDate 07/01/2021 -EndDate 08/03/2021 | ConvertTo-Json”
 - “AuditData” is namely where all info are ...

```
{
  "PSComputerName": "outlook.office365.com",
  "RunspaceId": "b93626fd-7d16-4983-b90d-8cf4185b56fe",
  "PSShowComputerName": false,
  "RecordType": "AzureActiveDirectoryStsLogon",
  "CreationDate": "\/Date(1627946033000)\/",
  "UserIds": "admin@M365x497090.onmicrosoft.com",
  "Operations": "UserLoggedIn",
  "AuditData": "{\\"CreationTime\\":\\"2021-08-02T23:13:53\\",\\"Id\\":\\"dcb61483-e579-4b0e-8e62-5f1ec8f4de00\\",\\"Operation\\":\\"UserLoggedIn\\",",
  "ResultIndex": 1,
  "ResultCount": 2710,
  "Identity": "dcb61483-e579-4b0e-8e62-5f1ec8f4de00",
  "IsValid": true,
  "ObjectState": "Unchanged"
},
```

- DFIR-O365RC module: all good, normal json
- Event **content**: straightforward



Forensic: Azure activity logs

- What: all related to Azure **resource** CRUD
 - Storage
 - DB
 - VM
 - Firewall
 - Network interface
 - ...
- Content
 - Json escaped in json is back 😊
 - Can identify rogue resource **creation/modification**
- **How**: DFIR-O365 module

```
Import-Module DFIR-O365RC

$enddate = get-date
$startdate = $enddate.adddays(-90)
Get-AzRMActivityLogs -StartDate $startdate -EndDate $enddate
```

```
{
  "category": {
    "value": "Administrative",
    "localizedValue": "Administrative"
  },
  "httpRequest": {
    "clientRequestId": "9996f36c-8ef4-4ca4-9739-720038734827",
    "clientIpAddress": "10.10.10.10",
    "method": "PUT"
  },
  "id": "/subscriptions/...",
  "level": "Informational",
  "resourceGroupName": "...",
  "resourceProviderName": {
    "value": "Microsoft.Compute",
    "localizedValue": "Microsoft.Compute"
  },
  "resourceId": "/subscriptions/...",
  "resourceType": {
    "value": "Microsoft.Compute/virtualMachines",
    "localizedValue": "Microsoft.Compute/virtualMachines"
  },
  "operationId": "109c484f-166a-4ef5-8513-53131326f71d",
  "operationName": {
    "value": "Microsoft.Compute/virtualMachines/write",
    "localizedValue": "Create or Update Virtual Machine"
  },
  "properties": {
    "statusCode": "Created",
    "serviceRequestId": "37a74ea3-a900-43a0-a5ac-66f64c2a6400",
    "responseBody": "{\n  \"name\": \"test-gwen\", \"id\": \"/subscriptions/...\", \"location\": \"West Europe\", \"hardwareProfile\": {\n    \"vmSize\": \"Standard_B1ms\"\n  }, \"osProfile\": {\n    \"linuxConfiguration\": {\n      \"disablePasswordAuthentication\": true\n    }\n  }, \"storageProfile\": {\n    \"imageReference\": {\n      \"publisher\": \"Canonical\", \"offer\": \"UbuntuServer\", \"sku\": \"16.04-LTS\", \"version\": \"latest\"\n    }, \"osDisk\": {\n      \"createOption\": \"FromImage\", \"managedBy\": true\n    }, \"dataDisks\": [\n      {\n        \"lun\": 0,\n        \"createOption\": \"FromImage\", \"managedBy\": true\n      }\n    ]\n  }, \"networkProfile\": {\n    \"networkInterfaces\": [\n      {\n        \"id\": \"/subscriptions/.../resourceGroups/.../networkInterfaces/...\", \"name\": \"nic1\", \"location\": \"West Europe\", \"properties\": {\n          \"enableAcceleratedNetworking\": true\n        }\n      }\n    ]\n  }, \"securityProfile\": {\n    \"securityType\": \"Standard\", \"securityProfile\": {\n      \"baseline\": \"AzureSecurityBaselineForUbuntu1604 LTS\"\n    }\n  }, \"tags\": {\n    \"Name\": \"test-gwen\"\n  }\n}"
    "entity": "/subscriptions/...",
    "message": "Microsoft.Compute/virtualMachines/write",
    "hierarchy": "..."
  },
  "status": {
    "value": "Accepted",
    "localizedValue": "Accepted"
  },
  "subStatus": {
    "value": "Created",
    "localizedValue": "Created (HTTP Status Code: 201)"
  },
  "eventTimestamp": "2021-08-05T15:45:24.3010792Z",
  "submissionTimestamp": "2021-08-05T15:46:46.1734172Z",
}
```

Forensic: Message Trace Reports

- What: email gateway logs, MTA in/out
- Collect from the **GUI**:
 - Asynchronous, takes time
 - 3 months available
 - Choose “Extended Report”
- Caveats:
 - **Latency**: 24 hours to see last emails
 - Sender: body, not envelop (**body spoofing** not distinguishable)
 - Some of the **headers**: not documented, and won't be
 - <https://github.com/MicrosoftDocs/OfficeDocs-o365seccomp/issues/442>
 - “We do not publicly advertise the purposes of all headers as the bad guys would then be able to use them to game the system”
 - Might be **not exhaustive**
 - Filter on recipient only: 3 emails missing
 - Filter on recipient + original client IP: the 3 emails appears !?
 - Ticket to Microsoft: “blocked emails not included until explicitly requested in the filter”
 - BUT: the 3 emails were “blocked” at one step ... and were finally **delivered to user inbox**
 - So: How do we **identify all impacted** users by a fraudulent email ???



Forensic: Disk timeline

- What: MACB timestamps of files and directories
- OneDrive “on-demand”
 - Files only **temporarily** downloaded when user opens it
 - Any files under “Documents” and “Desktop” are **synced**
 - Impact: forget about the **MFT** for user documents 😊
- “Solution”
 - LNK files are still there
 - Rely on **O365 audit logs**
 - FileAccessed
 - FileCopied
 - FileDeleted
 - FileModified
 - FileMoved
 - FileRenamed
 - FileDeletedFirstStageRecycleBin
 - FileDeletedSecondStageRecycleBin
 - FileDownloaded
 - FileUploaded

Containment: Token revocation, password reset

- Caveats on hybrid environment with Azure AD and AD on premise:
 - Scenario 1
 - Admin change an on-premise password with “user must **change password at next logon**”
 - Result: old password still active in Azure AD until user change it 😊
 - Scenario 2
 - Admin **disable** a user account on-premise
 - Result: account still active until next synchronization
 - Solution: force synchronization or disable the user everywhere
 - Scenario 3
 - User gave consent to a rogue application.
 - The application gets 2 tokens to impersonate the user’s account: access token and refresh token (to generate new access token)
 - Result: access token is **non-revocable**
- Documentation:
 - <https://docs.microsoft.com/en-us/azure/active-directory/hybrid/tshoot-connect-password-hash-synchronization#one-object-is-not-synchronizing-passwords-manual-troubleshooting-steps>
 - <https://docs.microsoft.com/en-us/microsoft-365/admin/add-users/remove-former-employee-step-1>
 - <https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-configurable-token-lifetimes>
 - <https://docs.microsoft.com/en-us/azure/active-directory/develop/refresh-tokens>
 - <https://docs.microsoft.com/en-us/powershell/module/azuread/revoke-azureaduserallrefreshtoken>



Remediation: Email policies

- Case:
 - Simple body spoofing
 - Ended in inbox ... despite appropriate **phishing policy**
- Caveat:
 - Welcome to **Artificial “Intelligence”** 😊
 - “Spoof Intelligence” = Mail From (not body)
 - “degree of confidence”
 - If no spam policy, might ends in inbox finally
- “Solution”:
 - Click everywhere to configure policies
 - And **test** with the client (spoofing, body spoofing)

Edit protection settings

Set your phishing thresholds and desired impersonation and spoof protections for this policy. [Learn more](#)

Phishing email threshold ⓘ

☐ 1 - Standard

This is the default value. The severity of the action that's taken on the message depends on the degree of confidence that the message is phishing (low, medium, high, or very high confidence).

Impersonation

☒ **Enable users to protect (5/350)** ⓘ

Enable impersonation protection for up to 350 internal and external users. [Learn more about adding users to impersonation protection](#)

[Manage 5 sender\(s\)](#)

☒ **Enable domains to protect (1)**

Enable impersonation protection for these internal and external sender domains.

☒ Include domains I own ⓘ

[View my domains](#)

☐ Include custom domains ⓘ

Add trusted senders and domains (12)

Add trusted senders and domains so they are not flagged as an impersonation-based attack

[Manage 12 trusted sender\(s\) and domain\(s\)](#)

☒ **Enable mailbox intelligence (Recommended)**

Enables artificial intelligence (AI) that determines user email patterns with their frequent contacts to identify potential impersonation attempts [Learn more](#)

☒ **Enable Intelligence for impersonation protection (Recommended)**

Enables enhanced impersonation results based on each user's individual sender map and allows you to define specific actions on impersonated messages

Spoof

☒ **Enable spoof intelligence (Recommended)**

Choose how you want to filter email from senders who are spoofing domains. To control which senders are allowed to spoof your domains or external domains, use the [Tenant Allow/Block List Spoofing page](#). [Learn more about Spoof Intelligence](#)



In a “nutshell”

- Initial Access
 - **MFA bypass** due to legacy protocols on mailboxes, bruteforce possible
 - **Consent** to application registration by default
 - **Azure resource** without basic practices (VM, blob storage, DB)
 - **Guest/partners** access rights by default
 - **Policies** bypassed (phishing/spam)
 - ... and all we don't know yet 😊
- Data acquisition
 - PowerShell to collect **logs**, <https://github.com/ANSSI-FR/DFIR-O365RC>
 - It works
 - It handles token refresh, API throttling, limited number of results per query
 - PowerShell to collect **configuration** (application IDs, application permissions)
 - Collect Message Trace Reports from **GUI** (beware of latency)
 - Collect other configurations with ... **screenshots** (user consent, policies)
- Logs analysis
 - Logins: “sometimes” hard to **identify** the source and targeted application
 - application ID **puzzle**
 - Some “Operations” available but not **filled**
 - Caution: Message Trace Report latency and still, not **exhaustive**

Thank you

