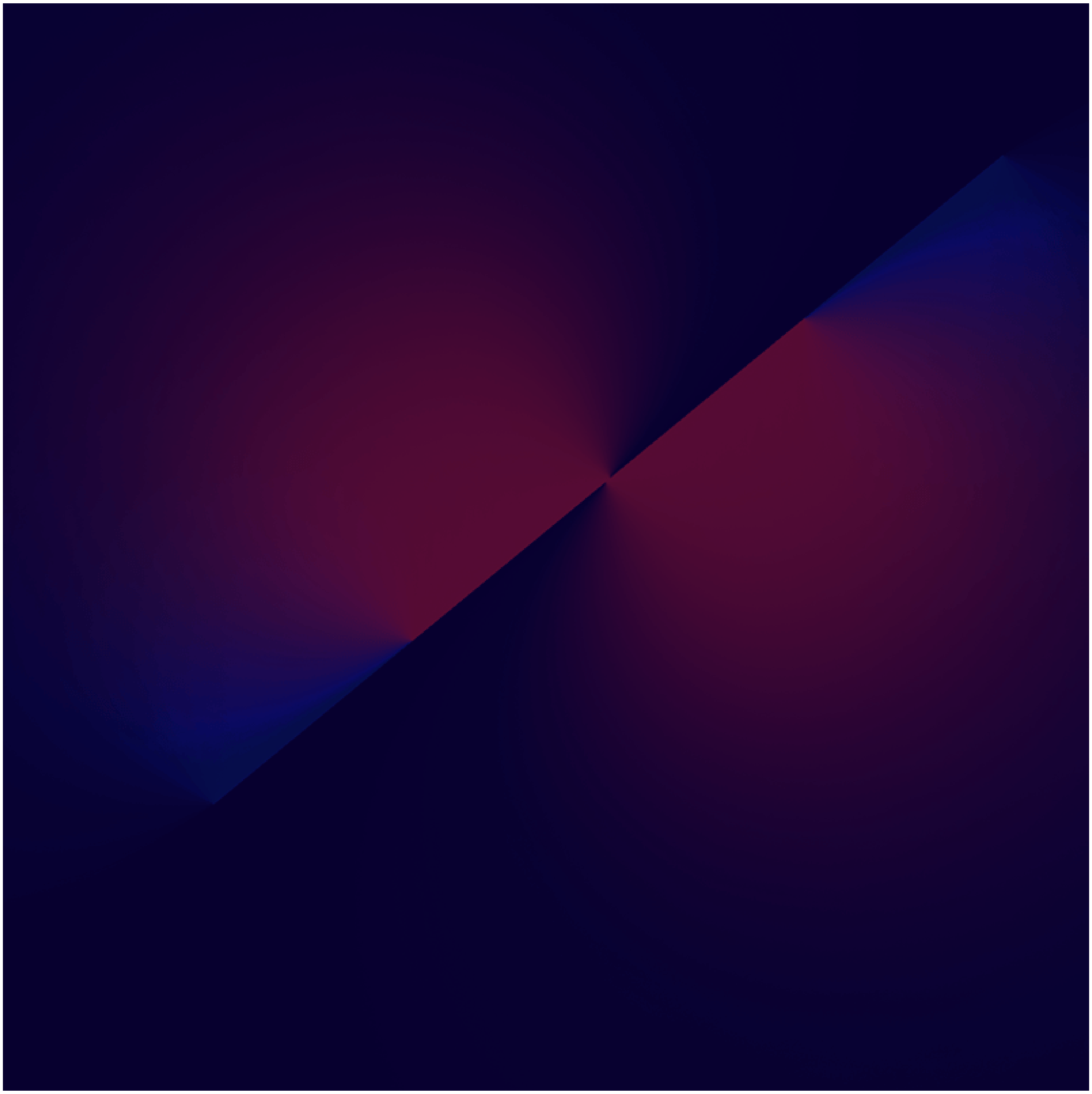
Azure Active Directory Exposes Internal Information

secureworks.com/research/azure-active-directory-exposes-internal-information

Counter Threat Unit Research Team



Updated: April 12, 2022

Summary

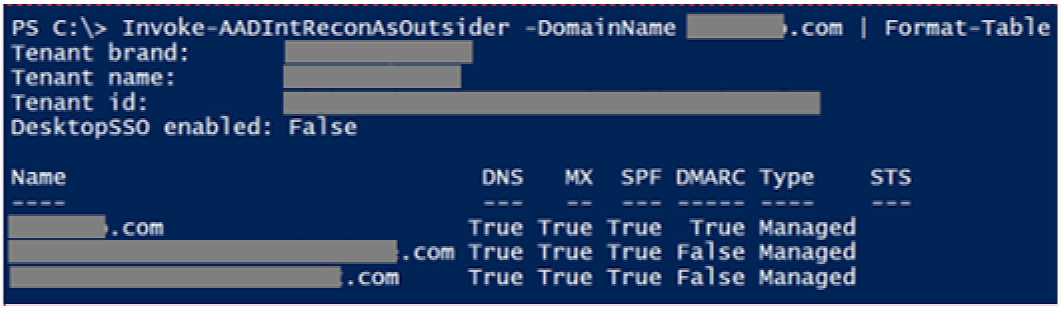
Microsoft Azure Active Directory (Azure AD) is an identity and access management solution used by over 88 percent of Fortune 500 companies as of this publication. This market penetration makes Azure AD a lucrative target for threat actors. In the second half of 2021, Secureworks® Counter Threat Unit™ (CTU) researchers analyzed Azure AD tenants and were able to extract open-source intelligence (OSINT) about organizations. Threat actors frequently use OSINT to perform reconnaissance. CTU™ researchers identified several application programming interfaces (APIs) that access internal information of any organization that uses Azure AD. Collected details included licensing information, mailbox information, and directory synchronization status.

CTU researchers shared their findings with Microsoft, and all but two of the issues have been mitigated as of this publication. Microsoft applied the updates automatically to all Azure AD tenants, so there are no actions required for Azure AD administrators. Microsoft classified the unmitigated issues as “by-design.” The first issue allows anyone to query the directory synchronization status. In some scenarios, Azure AD reveals the name of the high-privileged account used for synchronization. The second issue could reveal internal information about the target Azure AD tenant, including the technical contact’s full name and phone number. The technical contact usually holds Azure AD Global Administrator privileges.

Update: Microsoft addressed the remaining issues in April 2022.

OSINT details in Azure AD

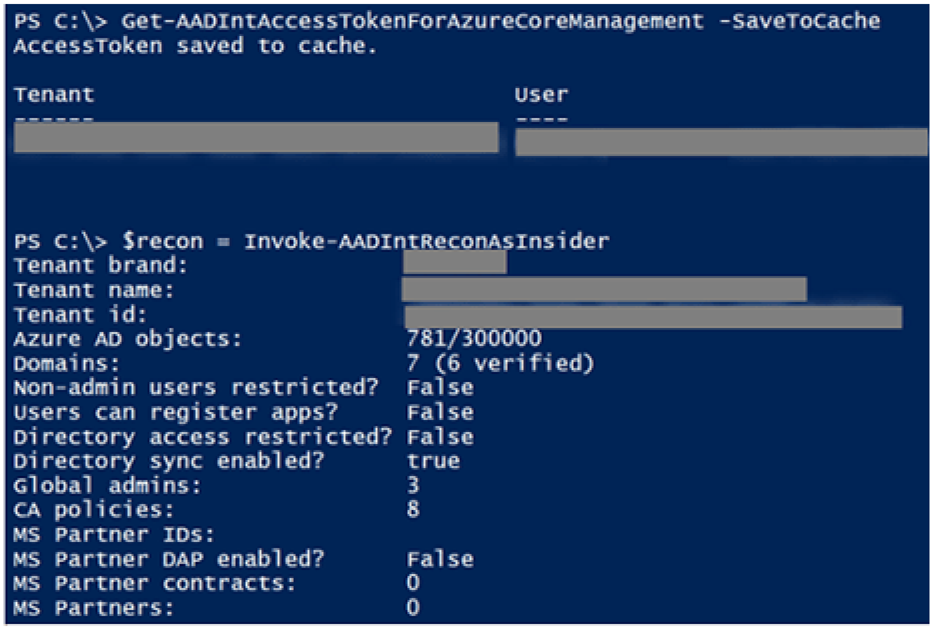
Tools such as AADInternals gather OSINT from Azure AD using unauthenticated APIs. This OSINT includes the target tenant’s registered domains and types, tenant name and ID, and seamless single sign-on status (also known as DesktopSSO). Figure 1 lists Invoke-AADIntReconAsOutsider command output that contains OSINT information about the organization.



```
PS C:\> Invoke-AADIntReconAsOutsider -DomainName [redacted].com | Format-Table
Tenant brand:       [redacted]
Tenant name:        [redacted]
Tenant id:          [redacted]
DesktopSSO enabled: False

Name                               DNS   MX   SPF  DMARC Type    STS
----                               ---   --   ---  ----- ----    ---
[redacted].com                    True True True   True Managed
[redacted].com                    True True True  False Managed
[redacted].com                    True True True  False Managed
```

Figure 1. Invoke-AADIntReconAsOutsider output listing OSINT from unauthenticated APIs. (Source: Secureworks)

In addition to the unauthenticated APIs, there are authenticated APIs that can only be used after logging into an Azure AD tenant. Figure 2 lists the information that any user can access from their own tenant. Administrator privileges are not required. CTU researchers discovered authenticated APIs that could access information about any tenant, not just the authenticated user's tenant.



```
PS C:\> Get-AADIntAccessTokenForAzureCoreManagement -SaveToCache
AccessToken saved to cache.

Tenant                                                User
------                                                ----
[redacted]                                            [redacted]



PS C:\> $recon = Invoke-AADIntReconAsInsider
Tenant brand:                 [redacted]
Tenant name:                  [redacted]
Tenant id:                    [redacted]
Azure AD objects:             781/300000
Domains:                      7 (6 verified)
Non-admin users restricted?   False
Users can register apps?      False
Directory access restricted?  False
Directory sync enabled?       true
Global admins:                3
CA policies:                  8
MS Partner IDs:
MS Partner DAP enabled?       False
MS Partner contracts:         0
MS Partners:                  0
```

Figure 2. Invoke-AADIntReconAsInsider output listing data from authenticated APIs. (Source: Secureworks)

Diagnostics API

Microsoft uses the undocumented Diagnostics API with the Support and Recovery Assistant (SaRA) tool to help the logged-in user diagnose and solve problems when accessing Microsoft cloud services. In 2019, CTU researchers observed SaRA using an analysis API endpoint. The traffic between the SaRA client and the analysis endpoint used the process in Figure 3.

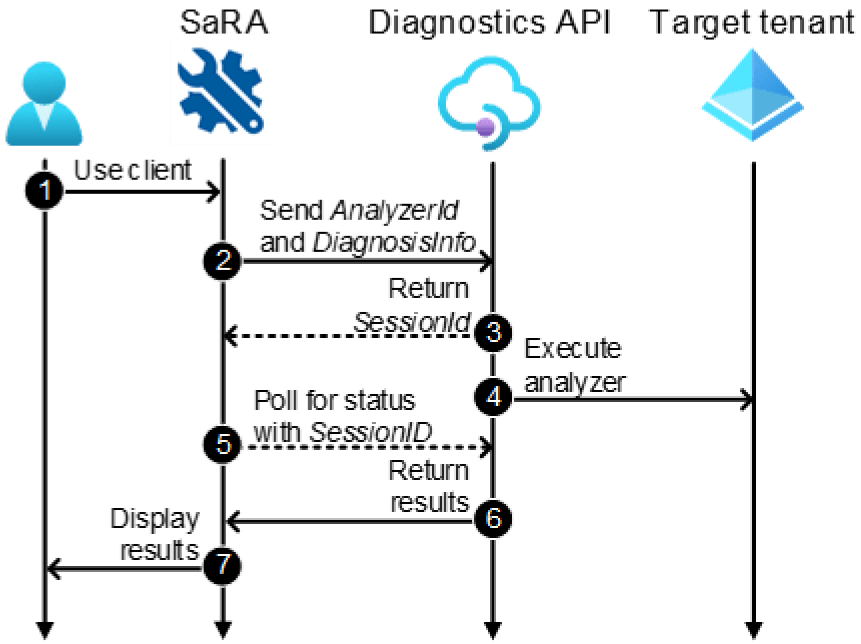
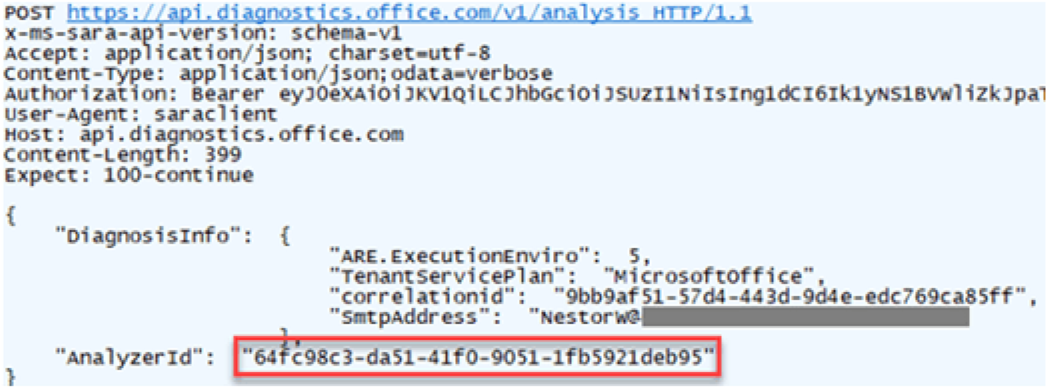


Figure 3. Diagnostics API analysis endpoint process. (Source: Secureworks)

1. A user opens SaRA, enters symptoms, and starts the diagnostic.

2. SaRA makes an initial HTTP POST request to the analysis endpoint (see Figure 4). The request contains an AnalyzerId and DiagnosisInfo.



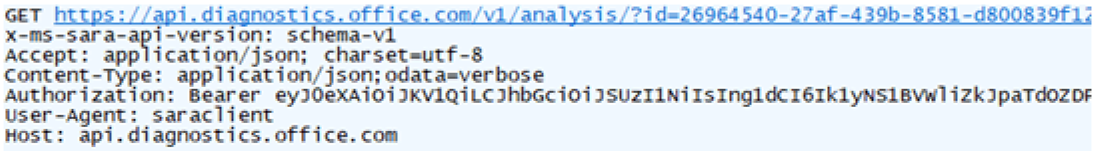
```
POST https://api.diagnostics.office.com/v1/analysis HTTP/1.1
x-ms-sara-api-version: schema-v1
Accept: application/json; charset=utf-8
Content-Type: application/json;odata=verbose
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ik1yNS1BVWlIZkJpa
User-Agent: saraclient
Host: api.diagnostics.office.com
Content-Length: 399
Expect: 100-continue

{
    "DiagnosisInfo":  {
                          "ARE.ExecutionEnviro":  5,
                          "TenantServicePlan":  "MicrosoftOffice",
                          "correlationid":  "9bb9af51-57d4-443d-9d4e-edc769ca85ff",
                          "SmtpAddress":  "NestorW@[redacted]
                      },
    "AnalyzerId":  "64fc98c3-da51-41f0-9051-1fb5921deb95"
}
```

Figure 4. Diagnostics API analysis endpoint initial request. (Source: Secureworks)

3. The response returns the SessionId to SaRA.

4. The Diagnostics API backend starts the analyzer to explore the defined user's tenant and mailbox.

5. SaRA uses an HTTP GET request and the SessionId to poll the analysis status (see Figure 5).



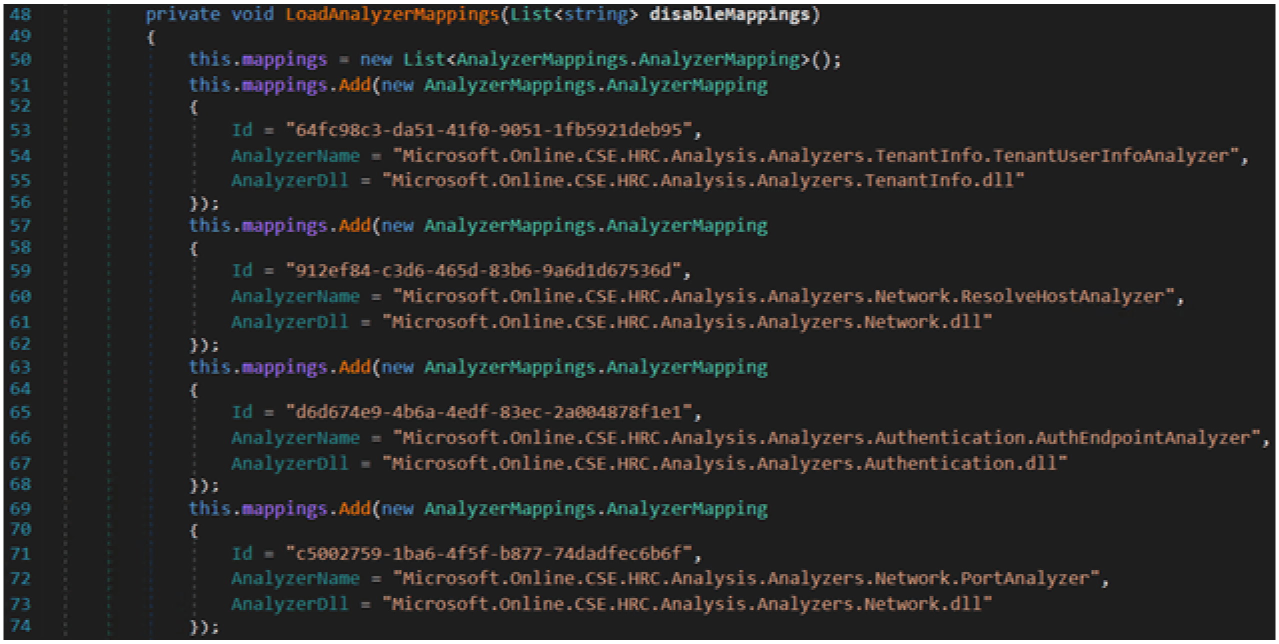
```
GET https://api.diagnostics.office.com/v1/analysis/?id=26964540-27af-439b-8581-d800839f1
x-ms-sara-api-version: schema-v1
Accept: application/json; charset=utf-8
Content-Type: application/json;odata=verbose
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ik1yNS1BVWlIZkJpaTdOZD
User-Agent: saraclient
Host: api.diagnostics.office.com
```

Figure 5. Diagnostics API analysis endpoint poll request. (Source: Secureworks)

6. The Diagnostics API returns analysis results to SaRA.

7. SaRA displays the results to the user.

The AnalyzerId represents an analyzer containing the diagnostic instructions that SaRA tasks the Diagnostics API to perform on the user's behalf. The SaRA client source code contains a list of analyzers (see Figure 6).



```
private void LoadAnalyzerMappings(List<string> disableMappings)
{
    this.mappings = new List<AnalyzerMappings.AnalyzerMapping>();
    this.mappings.Add(new AnalyzerMappings.AnalyzerMapping
    {
        Id = "64fc98c3-da51-41f0-9051-1fb5921deb95",
        AnalyzerName = "Microsoft.Online.CSE.HRC.Analysis.Analyzers.TenantInfo.TenantUserInfoAnalyzer",
        AnalyzerDll = "Microsoft.Online.CSE.HRC.Analysis.Analyzers.TenantInfo.dll"
    });
    this.mappings.Add(new AnalyzerMappings.AnalyzerMapping
    {
        Id = "912ef84-c3d6-465d-83b6-9a6d1d67536d",
        AnalyzerName = "Microsoft.Online.CSE.HRC.Analysis.Analyzers.Network.ResolveHostAnalyzer",
        AnalyzerDll = "Microsoft.Online.CSE.HRC.Analysis.Analyzers.Network.dll"
    });
    this.mappings.Add(new AnalyzerMappings.AnalyzerMapping
    {
        Id = "d6d674e9-4b6a-4edf-83ec-2a004878f1e1",
        AnalyzerName = "Microsoft.Online.CSE.HRC.Analysis.Analyzers.Authentication.AuthEndpointAnalyzer",
        AnalyzerDll = "Microsoft.Online.CSE.HRC.Analysis.Analyzers.Authentication.dll"
    });
    this.mappings.Add(new AnalyzerMappings.AnalyzerMapping
    {
        Id = "c5002759-1ba6-4f5f-b877-74dadfec6b6f",
        AnalyzerName = "Microsoft.Online.CSE.HRC.Analysis.Analyzers.Network.PortAnalyzer",
        AnalyzerDll = "Microsoft.Online.CSE.HRC.Analysis.Analyzers.Network.dll"
    });
```

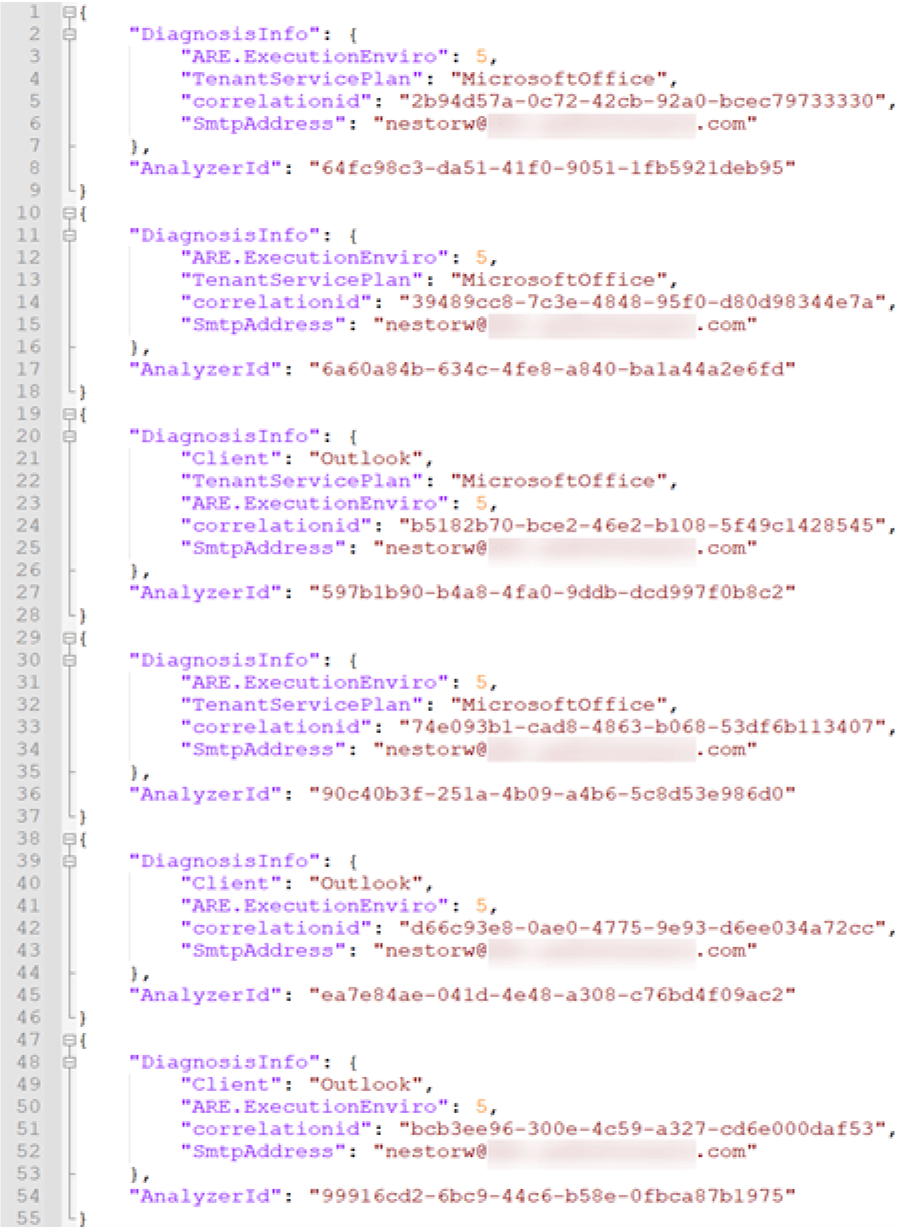
Figure 6. Sample of SaRA analyzer IDs and names from the analyzer list in the source code. (Source: Secureworks)

CTU researchers identified the cloud-related analyzers from this list (see Table 1).

Identifier	Name
64fc98c3-da51-41f0-9051-1fb5921deb95	TenantInfo.TenantUserInfoAnalyzer
6a60a84b-634c-4fe8-a840-ba1a44a2e6fd	TenantInfo.TenantSoftwareSettingsAnalyzer
99916cd2-6bc9-44c6-b58e-0fbca87b1975	ExchangeCmdlets.ExchangeHybridTenantAnalyzer
90c40b3f-251a-4b09-a4b6-5c8d53e986d0	ExchangeCmdlets.GetMailboxAnalyzer
597b1b90-b4a8-4fa0-9ddb-dcd997f0b8c2	ExchangeCmdlets.GetUserAnalyzer
ea7e84ae-041d-4e48-a308-c76bd4f09ac2	ExchangeCmdlets.CasMailboxAnalyzer

Table 1. Cloud-related Diagnosis API analysis endpoint analyzers.

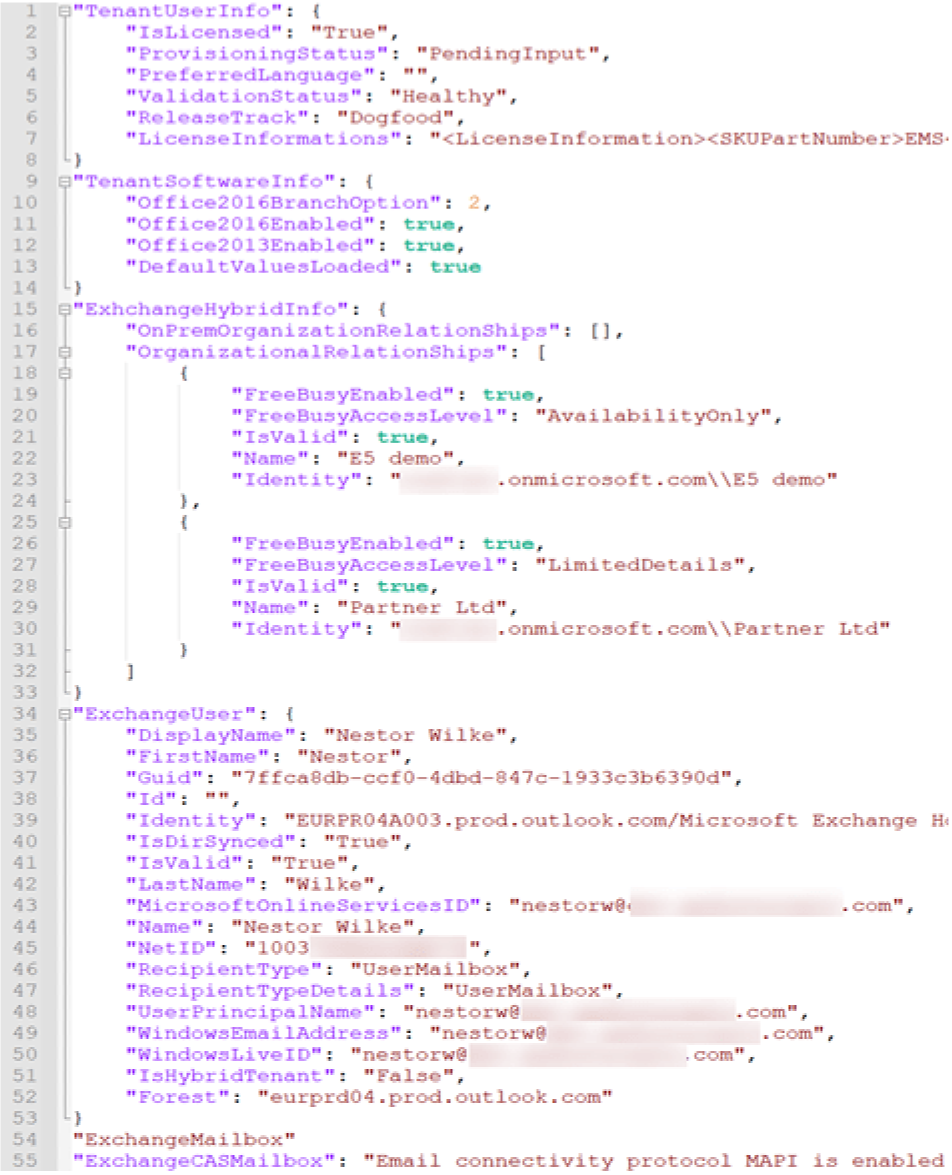
The SaRA client uses the DiagnosisInfo structure to pass parameters to analyzers. Figure 7 lists the parameters used by each of the cloud-related analyzers.



```
{
    "DiagnosisInfo": {
        "ARE.ExecutionEnviro": 5,
        "TenantServicePlan": "MicrosoftOffice",
        "correlationid": "2b94d57a-0c72-42cb-92a0-bcec79733330",
        "SmtpAddress": "nestorw@[redacted].com"
    },
    "AnalyzerId": "64fc98c3-da51-41f0-9051-1fb5921deb95"
}
{
    "DiagnosisInfo": {
        "ARE.ExecutionEnviro": 5,
        "TenantServicePlan": "MicrosoftOffice",
        "correlationid": "39489cc8-7c3e-4848-95f0-d80d98344e7a",
        "SmtpAddress": "nestorw@[redacted].com"
    },
    "AnalyzerId": "6a60a84b-634c-4fe8-a840-ba1a44a2e6fd"
}
{
    "DiagnosisInfo": {
        "Client": "Outlook",
        "TenantServicePlan": "MicrosoftOffice",
        "ARE.ExecutionEnviro": 5,
        "correlationid": "b5182b70-bce2-46e2-b108-5f49c1428545",
        "SmtpAddress": "nestorw@[redacted].com"
    },
    "AnalyzerId": "597b1b90-b4a8-4fa0-9ddb-dcd997f0b8c2"
}
{
    "DiagnosisInfo": {
        "ARE.ExecutionEnviro": 5,
        "TenantServicePlan": "MicrosoftOffice",
        "correlationid": "74e093b1-cad8-4863-b068-53df6b113407",
        "SmtpAddress": "nestorw@[redacted].com"
    },
    "AnalyzerId": "90c40b3f-251a-4b09-a4b6-5c8d53e986d0"
}
{
    "DiagnosisInfo": {
        "Client": "Outlook",
        "ARE.ExecutionEnviro": 5,
        "correlationid": "d66c93e8-0ae0-4775-9e93-d6ee034a72cc",
        "SmtpAddress": "nestorw@[redacted].com"
    },
    "AnalyzerId": "ea7e84ae-041d-4e48-a308-c76bd4f09ac2"
}
{
    "DiagnosisInfo": {
        "Client": "Outlook",
        "ARE.ExecutionEnviro": 5,
        "correlationid": "bcb3ee96-300e-4c59-a327-cd6e000daf53",
        "SmtpAddress": "nestorw@[redacted].com"
    },
    "AnalyzerId": "99916cd2-6bc9-44c6-b58e-0fbca87b1975"
}
```

Figure 7. DiagnosisInfo content for each cloud-related analyzer. (Source: Secureworks)

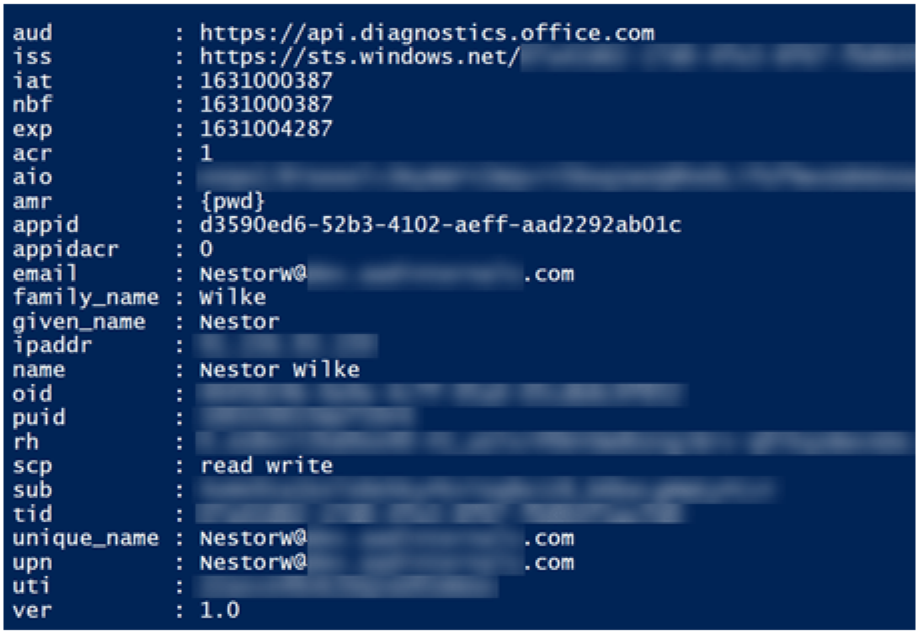
The results contain user information, including full licensing information, Office versions enabled in the tenant, the organization’s Exchange hybrid configuration and external relationships, user mailbox information, and Messaging Application Programming Interface (MAPI) status (see Figure 8).



```
"TenantUserInfo": {
    "IsLicensed": "True",
    "ProvisioningStatus": "PendingInput",
    "PreferredLanguage": "",
    "ValidationStatus": "Healthy",
    "ReleaseTrack": "Dogfood",
    "LicenseInformations": "<LicenseInformation><SKUPartNumber>EMS
}
"TenantSoftwareInfo": {
    "Office2016BranchOption": 2,
    "Office2016Enabled": true,
    "Office2013Enabled": true,
    "DefaultValuesLoaded": true
}
"ExhchangeHybridInfo": {
    "OnPremOrganizationRelationShips": [],
    "OrganizationalRelationShips": [
        {
            "FreeBusyEnabled": true,
            "FreeBusyAccessLevel": "AvailabilityOnly",
            "IsValid": true,
            "Name": "E5 demo",
            "Identity": "[redacted].onmicrosoft.com\\E5 demo"
        },
        {
            "FreeBusyEnabled": true,
            "FreeBusyAccessLevel": "LimitedDetails",
            "IsValid": true,
            "Name": "Partner Ltd",
            "Identity": "[redacted].onmicrosoft.com\\Partner Ltd"
        }
    ]
}
"ExchangeUser": {
    "DisplayName": "Nestor Wilke",
    "FirstName": "Nestor",
    "Guid": "7ffca8db-ccf0-4dbd-847c-1933c3b6390d",
    "Id": "",
    "Identity": "EURPR04A003.prod.outlook.com/Microsoft Exchange H
    "IsDirSynced": "True",
    "IsValid": "True",
    "LastName": "Wilke",
    "MicrosoftOnlineServicesID": "nestorw@[redacted].com",
    "Name": "Nestor Wilke",
    "NetID": "1003[redacted]",
    "RecipientType": "UserMailbox",
    "RecipientTypeDetails": "UserMailbox",
    "UserPrincipalName": "nestorw@[redacted].com",
    "WindowsEmailAddress": "nestorw@[redacted].com",
    "WindowsLiveID": "nestorw@[redacted].com",
    "IsHybridTenant": "False",
    "Forest": "eurprd04.prod.outlook.com"
}
"ExchangeMailbox"
"ExchangeCASMailbox": "Email connectivity protocol MAPI is enabled
```

Figure 8. Information returned by Diagnostics API analysis endpoint. (Source: Secureworks)

The SaRA client extracts the logged-in user’s email address from their OAuth token (see Figure 9) and uses that as the target SmtpAddress in the DiagnosisInfo parameter.



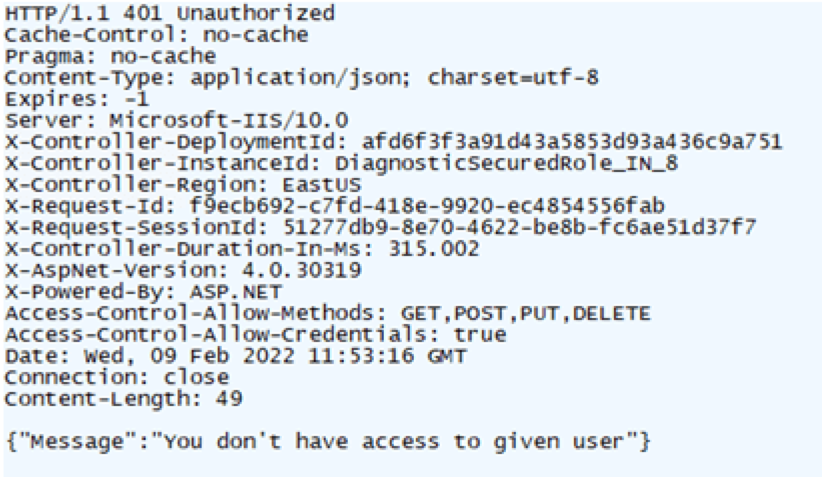
```
aud         : https://api.diagnostics.office.com
iss         : https://sts.windows.net/[redacted]
iat         : 1631000387
nbf         : 1631000387
exp         : 1631004287
acr         : 1
aio         : [redacted]
amr         : {pwd}
appid       : d3590ed6-52b3-4102-aeff-aad2292ab01c
appidacr    : 0
email       : Nestorw@[redacted].com
family_name : Wilke
given_name  : Nestor
ipaddr      : [redacted]
name        : Nestor Wilke
oid         : [redacted]
puid        : [redacted]
rh          : [redacted]
scp         : read write
sub         : [redacted]
tid         : [redacted]
unique_name : Nestorw@[redacted].com
upn         : Nestorw@[redacted].com
uti         : [redacted]
ver         : 1.0
```

Figure 9. OAuth token for Diagnostics API. (Source: Secureworks)

The Diagnostics API does not validate whether the SmtpAddress matches the logged-in user. It is possible to retrieve information for any user from any tenant by replacing the SmtpAddress with the email address of the target user. If the target user does not exist but the domain is correct, the API returns all tenant-related information. This information is valuable to threat actors. For instance, the licensing information shows which protective components the target tenant could be using. Moreover, the organizational relationships identify additional individuals that could be targeted in phishing attacks to gain access to a tenant.

CTU researchers reported this vulnerability to Microsoft on September 7, 2021. On September 22, Microsoft responded that the issue was resolved. CTU researchers confirmed that the resolution included two modifications:

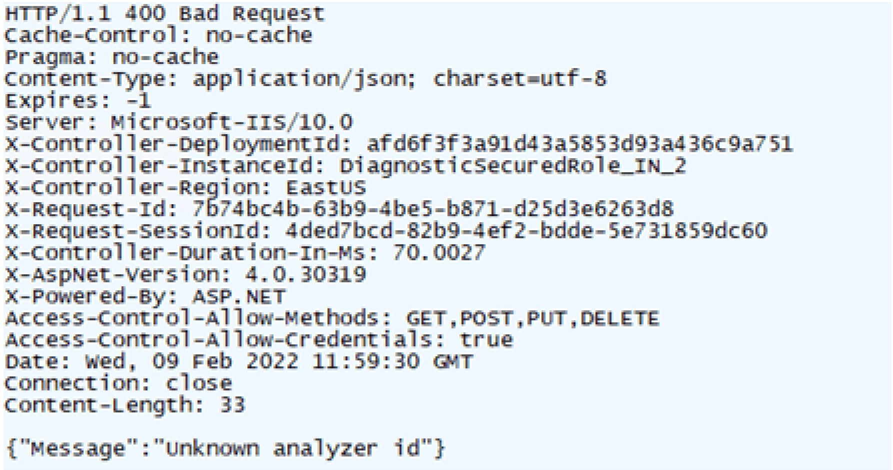
- Denies access to other users' information (see Figure 10).



```
HTTP/1.1 401 Unauthorized
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/10.0
X-Controller-DeploymentId: afd6f3f3a91d43a5853d93a436c9a751
X-Controller-InstanceId: DiagnosticSecuredRole_IN_8
X-Controller-Region: EastUS
X-Request-Id: f9ecb692-c7fd-418e-9920-ec4854556fab
X-Request-SessionId: 51277db9-8e70-4622-be8b-fc6ae51d37f7
X-Controller-Duration-In-Ms: 315.002
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Methods: GET,POST,PUT,DELETE
Access-Control-Allow-Credentials: true
Date: Wed, 09 Feb 2022 11:53:16 GMT
Connection: close
Content-Length: 49

{"Message":"You don't have access to given user"}
```

Figure 10. 'You don't have access to given user' response. (Source: Secureworks

- Invalidates all AnalyzerIDs, making the analysis endpoint obsolete (see Figure 11).



```
HTTP/1.1 400 Bad Request
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/10.0
X-Controller-DeploymentId: afd6f3f3a91d43a5853d93a436c9a751
X-Controller-InstanceId: DiagnosticSecuredRole_IN_2
X-Controller-Region: EastUS
X-Request-Id: 7b74bc4b-63b9-4be5-b871-d25d3e6263d8
X-Request-SessionId: 4ded7bcd-82b9-4ef2-bdde-5e731859dc60
X-Controller-Duration-In-Ms: 70.0027
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Methods: GET,POST,PUT,DELETE
Access-Control-Allow-Credentials: true
Date: Wed, 09 Feb 2022 11:59:30 GMT
Connection: close
Content-Length: 33

{"Message":"Unknown analyzer id"}
```

Figure 11. 'Unknown analyzer id' response. (Source: Secureworks)

In 2021, CTU analysis of SaRA version 17.0.7.7119.4 revealed the client using the cloudcheck endpoint instead of the analysis endpoint. Figure 12 depicts the cloudcheck endpoint process.

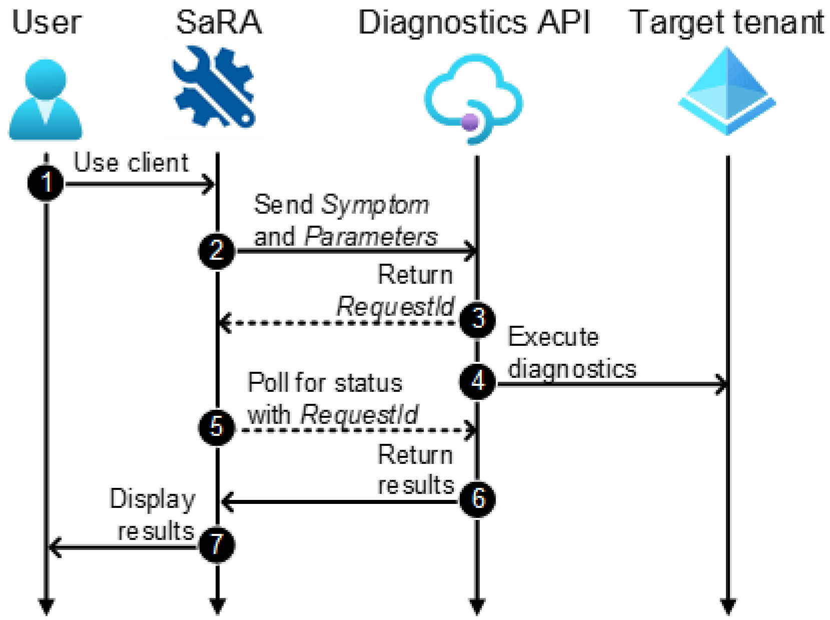
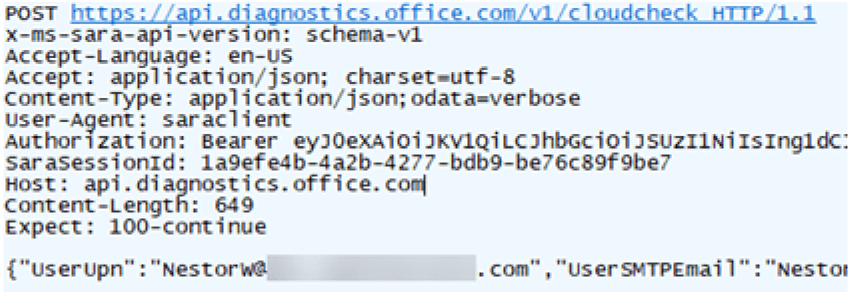


Figure 12. Diagnostics API cloudcheck endpoint process. (Source: Secureworks)

1. A user opens SaRA, enters symptoms, and starts the diagnostic.

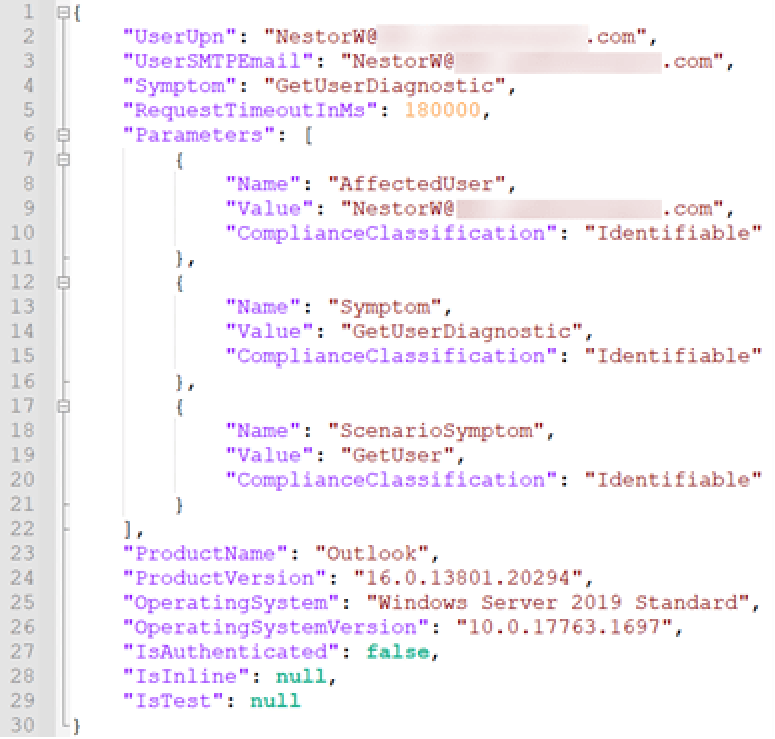
2. SaRA makes an initial HTTP POST request to the cloudcheck endpoint (see Figure 13).



```
POST https://api.diagnostics.office.com/v1/cloudcheck HTTP/1.1
x-ms-sara-api-version: schema-v1
Accept-Language: en-US
Accept: application/json; charset=utf-8
Content-Type: application/json;odata=verbose
User-Agent: saraclient
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dC
SaraSessionId: 1a9efe4b-4a2b-4277-bdb9-be76c89f9be7
Host: api.diagnostics.office.com
Content-Length: 649
Expect: 100-continue

{"UserUpn":"NestorW@[redacted].com","UserSMTPEmail":"Nestor
```

Figure 13. Diagnostics API cloudcheck endpoint initial request. (Source: Secureworks)

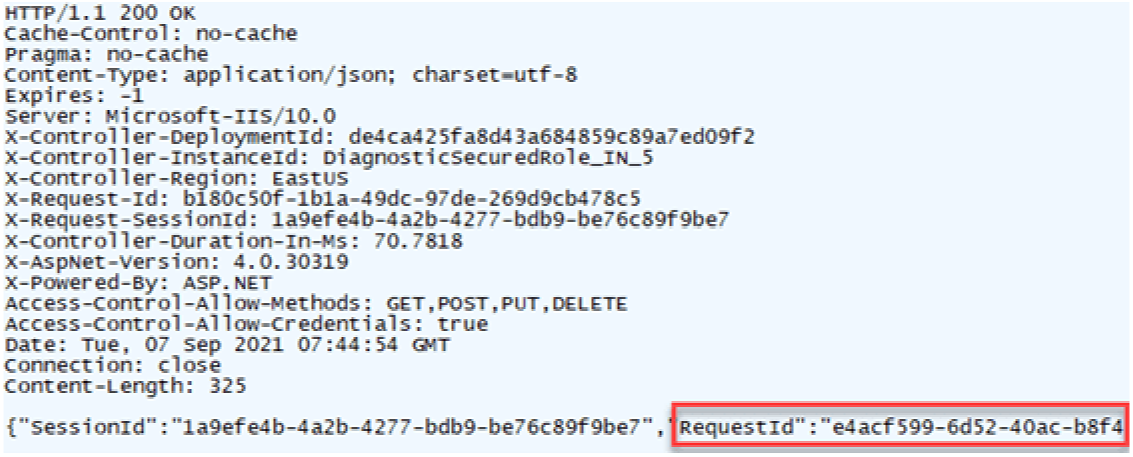
The request contains the Symptom and Parameters details (see Figure 14) the user entered in Step 1.



```
{
    "UserUpn": "NestorW@[redacted].com",
    "UserSMTPEmail": "NestorW@[redacted].com",
    "Symptom": "GetUserDiagnostic",
    "RequestTimeoutInMs": 180000,
    "Parameters": [
        {
            "Name": "AffectedUser",
            "Value": "NestorW@[redacted].com",
            "ComplianceClassification": "Identifiable"
        },
        {
            "Name": "Symptom",
            "Value": "GetUserDiagnostic",
            "ComplianceClassification": "Identifiable"
        },
        {
            "Name": "ScenarioSymptom",
            "Value": "GetUser",
            "ComplianceClassification": "Identifiable"
        }
    ],
    "ProductName": "Outlook",
    "ProductVersion": "16.0.13801.20294",
    "OperatingSystem": "Windows Server 2019 Standard",
    "OperatingSystemVersion": "10.0.17763.1697",
    "IsAuthenticated": false,
    "IsInline": null,
    "IsTest": null
}
```

Figure 14. Information sent to cloudcheck endpoint. (Source: Secureworks)

3. The response returns the RequestId to SaRA (see Figure 15).

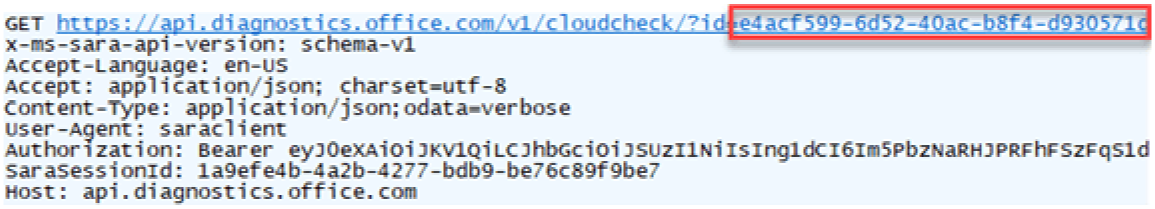


```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/10.0
X-Controller-DeploymentId: de4ca425fa8d43a684859c89a7ed09f2
X-Controller-InstanceId: DiagnosticSecuredRole_IN_5
X-Controller-Region: EastUS
X-Request-Id: b180c50f-1b1a-49dc-97de-269d9cb478c5
X-Request-SessionId: 1a9efe4b-4a2b-4277-bdb9-be76c89f9be7
X-Controller-Duration-In-Ms: 70.7818
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Access-Control-Allow-Methods: GET,POST,PUT,DELETE
Access-Control-Allow-Credentials: true
Date: Tue, 07 Sep 2021 07:44:54 GMT
Connection: close
Content-Length: 325

{"SessionId":"1a9efe4b-4a2b-4277-bdb9-be76c89f9be7","RequestId":"e4acf599-6d52-40ac-b8f4
```

Figure 15. Diagnostics API initial response. (Source: Secureworks)

4. The diagnosis API backend starts the diagnostics to explore the defined user’s tenant and mailbox.

5. SaRA uses an HTTP GET request and the RequestId to poll the analysis status (see Figure 16).



```
GET https://api.diagnostics.office.com/v1/cloudcheck/?id=e4acf599-6d52-40ac-b8f4-d930571c
x-ms-sara-api-version: schema-v1
Accept-Language: en-US
Accept: application/json; charset=utf-8
Content-Type: application/json;odata=verbose
User-Agent: saraclient
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5PbzNaRHJPRFhFSzFqS1d
SaraSessionId: 1a9efe4b-4a2b-4277-bdb9-be76c89f9be7
Host: api.diagnostics.office.com
```

Figure 16. Diagnostics API v1 poll request. (Source: Secureworks)

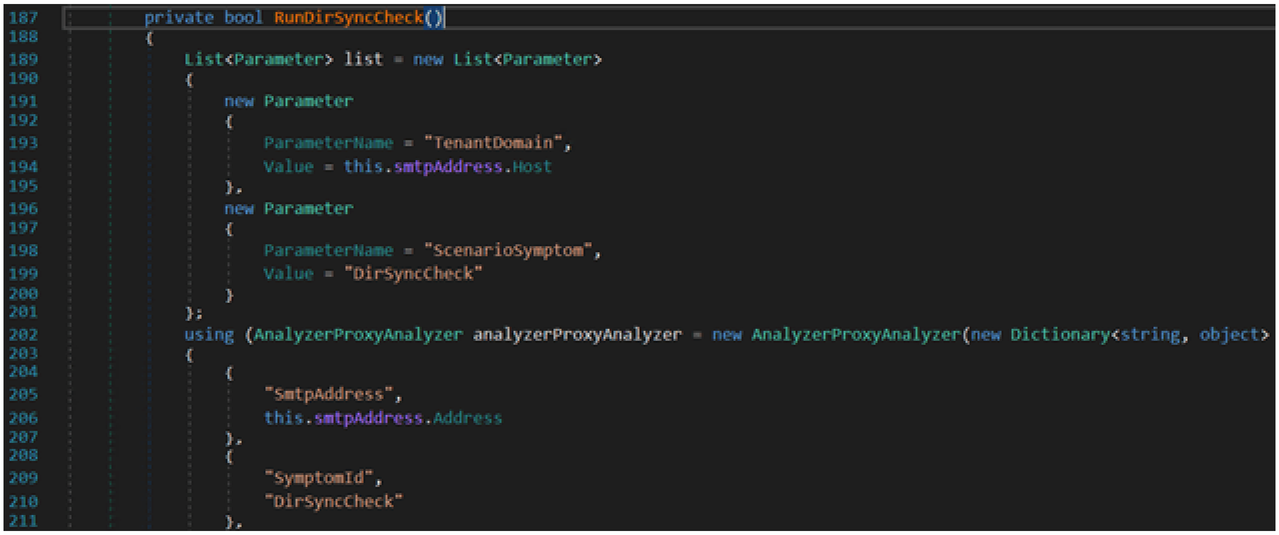
6. The cloudcheck endpoint returns diagnostic results to SaRA.

7. SaRA displays the results to the user.

The SaRA client revealed the following symptoms that could retrieve similar diagnostic information as the analysis endpoint:

- CasMailbox
- DirSyncCheck
- ExchangeHybridTenant
- GetUserDiagnostic
- TenantUserInfo

Figure 17 lists the parameters used by the DirSyncCheck symptom.



```
private bool RunDirSyncCheck()
{
    List<Parameter> list = new List<Parameter>
    {
        new Parameter
        {
            ParameterName = "TenantDomain",
            Value = this.smtpAddress.Host
        },
        new Parameter
        {
            ParameterName = "ScenarioSymptom",
            Value = "DirSyncCheck"
        }
    };
    using (AnalyzerProxyAnalyzer analyzerProxyAnalyzer = new AnalyzerProxyAnalyzer(new Dictionary<string, object>
    {
        {
            "SmtpAddress",
            this.smtpAddress.Address
        },
        {
            "SymptomId",
            "DirSyncCheck"
        },
```

Figure 17. Parameter values for DirSyncCheck request. (Source: Secureworks)

Like the analysis endpoint, the UserUpn and UserSMTPEmail attributes in the initial request were the same as the user principal name of the bearer token used to access the API. As with the analysis endpoint, it was possible to retrieve information for other users and tenants by replacing the values with the email address of the target user. After Microsoft addressed the analysis endpoint issue, the logged-in user could only retrieve CasMailBox information for users of the same tenant. However, all other information could still be requested from any tenant.

CTU researchers reported this vulnerability to Microsoft on September 23, 2021. On December 2, 2021, Microsoft applied an update. CTU researchers confirmed that everything except the directory synchronization status issue was addressed. On January 28, 2022, Microsoft closed the issue as fixed, leaving the synchronization status intact.

Table 2 lists the directory synchronization status values. While all status information is important for threat actors, the password expiration message is the most valuable as it reveals the account name used for synchronization. This account has high privileges in the target tenant. It can be used to create, edit, and delete users in all tenants, and to reset users' passwords in some tenants. By default, the synchronization account's password is generated during the configuration and is not set to expire. For security purposes, some organizations configure the password to expire in their tenants, which could expose the account name. The password expiration reminder can be configured to be sent 1 to 30 days prior to the expiration date.

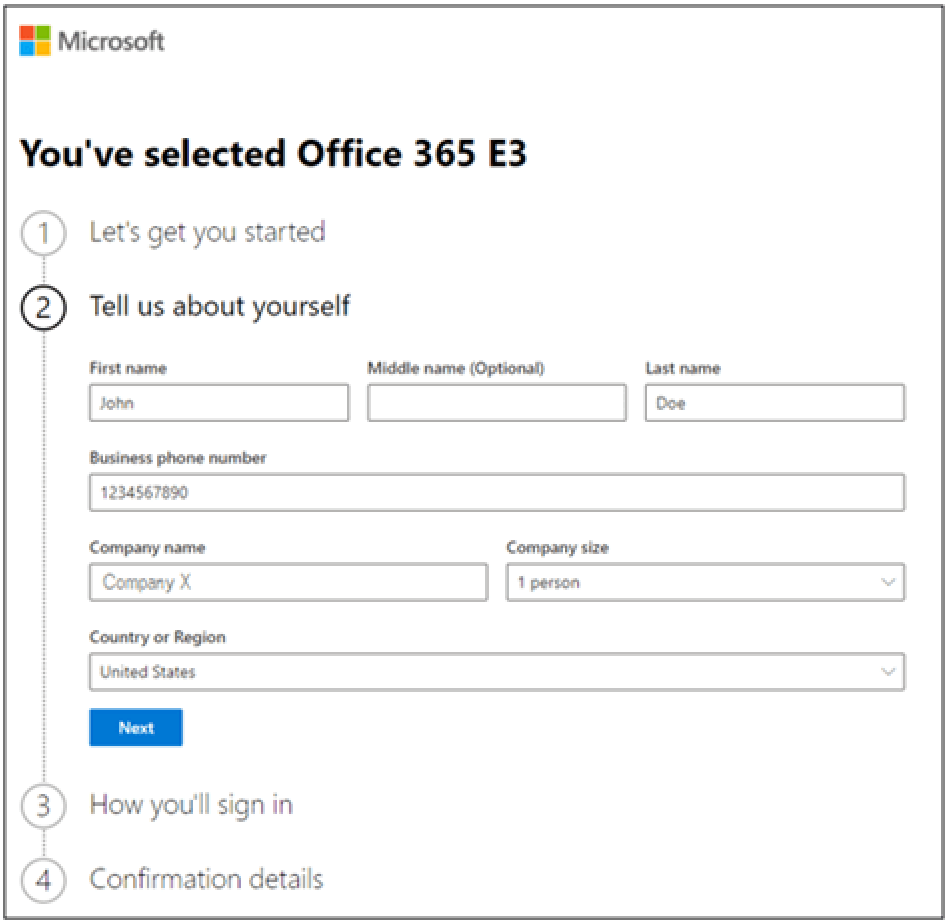
Synchronization status message	Description
Directory Synchronization (or) password Synchronization is enabled for your tenant: *<redacted>*	Directory synchronization is enabled and working normally

Synchronization status message	Description
Active Directory Synchronization or Password Synchronization needs to be enabled for your tenant: *<redacted>*. This is something your Office 365 administrator can fix.	Directory synchronization is not enabled
Your tenant *<redacted>* password Synchronization server hasn't successfully synchronized with Office 365 in the last three hours. The last time it synced was 9/23/2020.	Directory synchronization is enabled but has not been successfully synchronized after the listed date
Your tenant *<redacted>* directory Synchronization server hasn't successfully synchronized with Office 365 in the last three hours. The last time it synced was 1/1/0001.	Directory synchronization is enabled but has never been successfully synchronized
Your tenant *<redacted>* directory synchronization service account *<redacted>*@*<redated>*.onmicrosoft.com password is expiring in 11 days. This is something your Office 365 administrator can fix.	Directory synchronization is enabled and working normally, but the password of the account used for synchronization is expiring soon

Table 2. Directory synchronization status messages.

Organization information

Azure AD collects information when a representative from an organization signs up for a new Microsoft 365 or Azure AD environment or tenant. The form collects the full name and phone number of this representative (see Figure 18), and that person becomes the technical contact of the tenant.



You've selected Office 365 E3

1 Let's get you started

2 Tell us about yourself

First name	Middle name (Optional)	Last name
John		Doe

Business phone number: 1234567890

Company name: Company X

Company size: 1 person

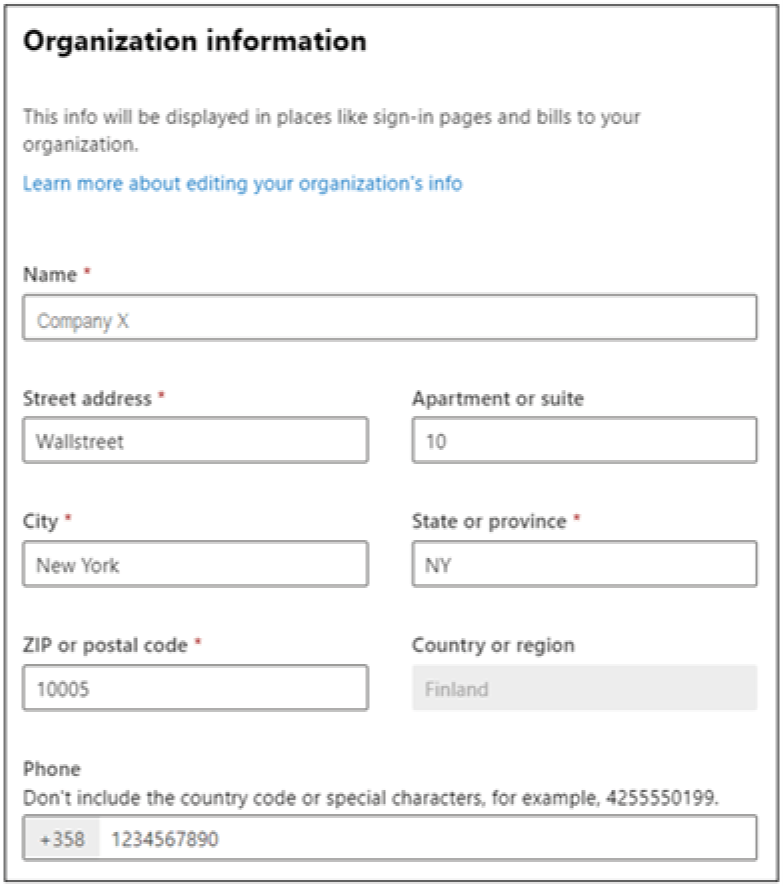
Country or Region: United States

3 How you'll sign in

4 Confirmation details

Figure 18. Office 365 signup form. (Source: Secureworks)

After signing up, this technical contact can edit their contact details in the Microsoft 365 admin center (see Figure 19). The company name and phone number are pre-populated from the original signup form.



Organization information

This info will be displayed in places like sign-in pages and bills to your organization.

Learn more about editing your organization's info

Name *

Company X

Street address *

Wallstreet

Apartment or suite

10

City *

New York

State or province *

NY

ZIP or postal code *

10005

Country or region

Finland

Phone

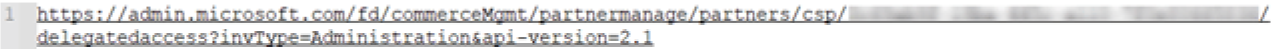
Don't include the country code or special characters, for example, 4255550199.

+358 1234567890

Figure 19. Organization information in the admin center. (Source: Secureworks)

Microsoft business partners offer services to customer organizations that use Microsoft cloud services such as Microsoft 365 and Azure AD. Azure AD administrators in customer organizations can authorize these partners to access their tenants, which creates a partner relationship in the customer's tenant. These partner relationships can only be accessed via the Microsoft 365 admin center. Only administrators have access to the admin center.

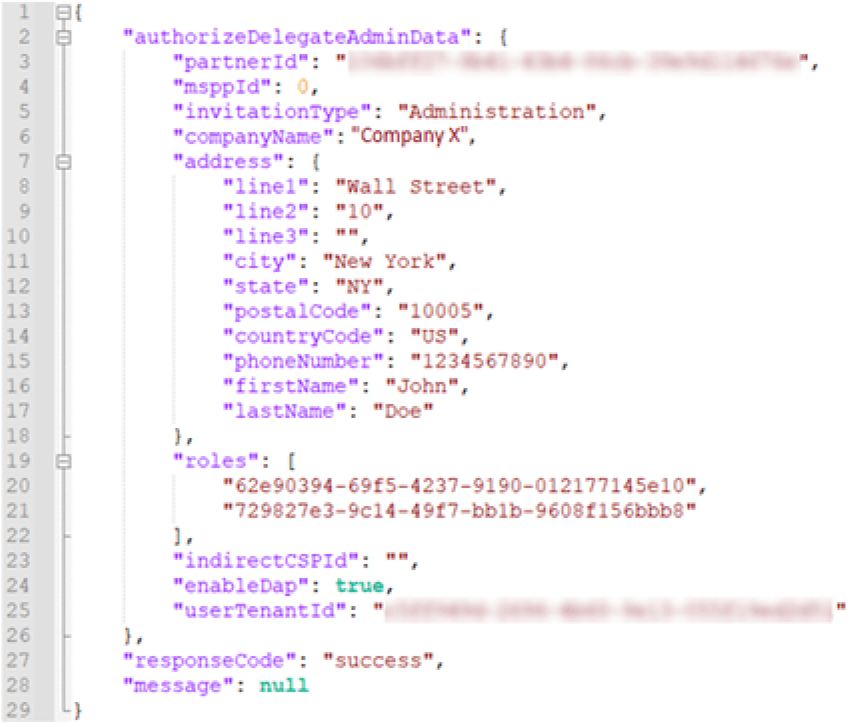
CTU researchers discovered an API (see Figure 20) used by the admin center to retrieve details regarding the partner's organization. Although the API is exclusively used by the admin center, it does not require administrative permissions to be accessed. The API requires the partner's tenant ID as an input.



```
https://admin.microsoft.com/fd/commerceMgmt/partnermanage/partners/csp/[redacted]/
delegatedaccess?invType=Administration&api-version=2.1
```

Figure 20. Admin API request for partner details. (Source: Secureworks)

The response (see Figure 21) contains contact data from the organization information and signup form. After the initial signup, the first and last name can only be changed by Microsoft. Those fields cannot be viewed or modified in the admin center.



```
{
    "authorizeDelegateAdminData": {
        "partnerId": "[redacted]",
        "msppId": 0,
        "invitationType": "Administration",
        "companyName":"Company X",
        "address": {
            "line1": "Wall Street",
            "line2": "10",
            "line3": "",
            "city": "New York",
            "state": "NY",
            "postalCode": "10005",
            "countryCode": "US",
            "phoneNumber": "1234567890",
            "firstName": "John",
            "lastName": "Doe"
        },
        "roles": [
            "62e90394-69f5-4237-9190-012177145e10",
            "729827e3-9c14-49f7-bb1b-9608f156bbb8"
        ],
        "indirectCSPId": "",
        "enableDap": true,
        "userTenantId": "[redacted]"
    },
    "responseCode": "success",
    "message": null
}
```

Figure 21. Partner information returned by admin API. (Source: Secureworks)

CTU researchers verified that this API could retrieve this information for any tenant, regardless of their partner status. CTU researchers reported this vulnerability to Microsoft on December 14, 2021. On January 12, 2022, Microsoft stated that “this information is expected to be shown” and did not mitigate the issue.

Conclusion

A threat actor can gather a significant amount of OSINT from an Azure AD tenant. Microsoft addressed all but two of the issues CTU researchers identified:

- The tenant’s synchronization status can reveal if the synchronization is configured, if is it operational, the time of the last synchronization, and the synchronization account’s name. Attackers can use this information for social engineering (leveraging the synchronization error data) and targeted brute-force attacks (using the account name).

- The organization information could expose the name and phone number of the tenant’s Global Administrator. This information can be abused for social engineering, spearphishing, and targeted brute-force attacks.

CTU researchers recommend the following actions to protect tenants from OSINT abuse:

- Organizations should ensure that their directory synchronization can perform the synchronization within the defined timeframes to avoid exposing details in error messages. Administrators receive an email if synchronization has not been successful in more than 24 hours, but the error message is displayed after three hours of inactivity.

- Organizations that implement an expiration for a directory synchronization account password should reset the password before Azure AD displays the expiration reminder to prevent exposure of the directory synchronization account name.

- Organizations should change the details associated with their tenant to general labels (e.g., “IT Department”) rather than personally identifiable data. Using a generic term prevents exposing the name of the potential Global Administrator account. An organization can modify some fields (e.g., phone number), but must create a support request in the Azure portal to change the first and last name of the technical contact.

April 12 update

After this analysis was published on April 5, 2022, Microsoft reassessed the two remaining issues. CTU researchers verified that these issues have been addressed as of April 12:

- The synchronization status is only visible for user's tenant.
- Only administrators can access the admin API that exposes organizational information. Additionally, the API does not return the technical contact's name.