



Demystifying Graph API and Service Now

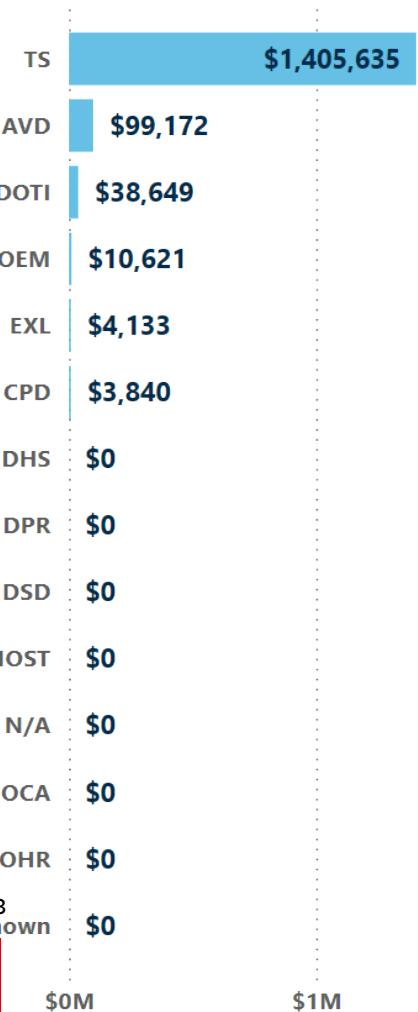
Using a Service account, CrowdStrike and Ansible to
glue it all together





Last Refresh: 5/8/2024 3:30:10 PM

Total Dollar Savings by Agency



Total Dollar Savings*

*Data from 2023 to present

\$1,562,051

\$1,025,923

TS Soft Savings

\$310,800

One-Time Savings

\$171,433

Non-TS Soft Savings

\$53,895

Hard Savings

Total Refocused Hours

24,804

20,518

TS Staff Hours Saved

4,286

Non-TS Staff Hours Saved

Averages

Rolling Average - Past 30 Days

\$4,513

Dollars Saved per Day

81.8

Hours Saved per Day

Automated Processes

133

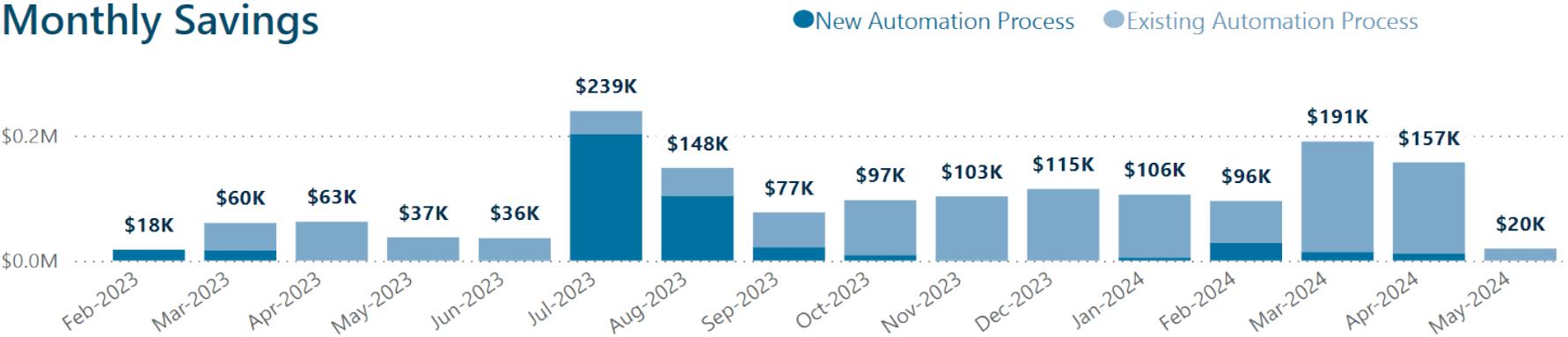
Year

All

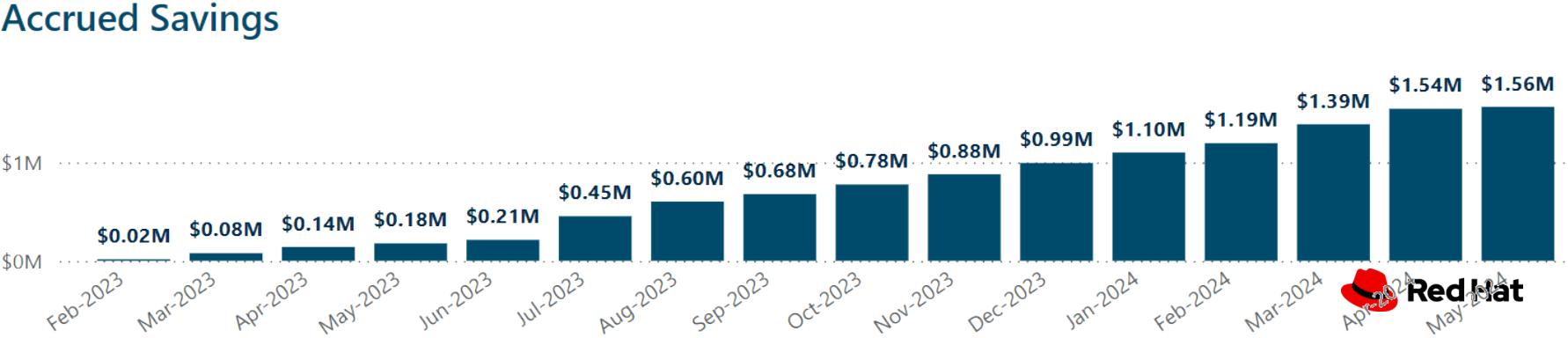
Tool

Ansible

Monthly Savings



Accrued Savings



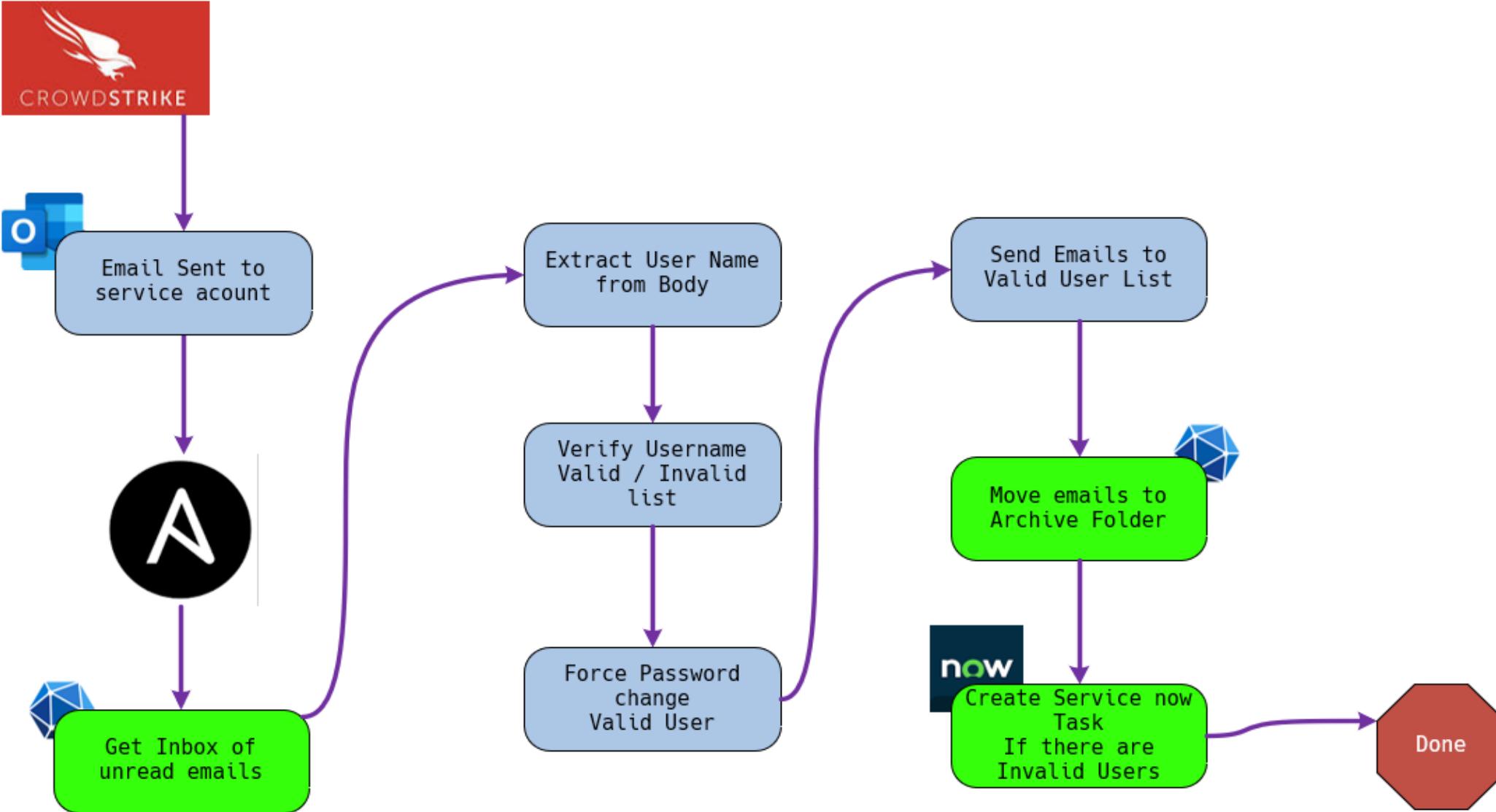
50,000 foot view

The Problem

- CrowdStrike sends an email to a service account, each email is a user that is using a password that is on a known compromise list. Like “password123”
- Ansible once an hour will do the following:
 - Grab any new mail and process them (Graph API)
 - Ansible then will match the account and send a notification and force a password change (A D)
 - Move all the messages that were processed to the Archive folder (Graph API)
 - Unmatched accounts will get a ServiceNow task assigned to Security (Service Now)

Step 1

The Problem



25,000 foot view

Pieces to get the job done

1. **Get with the Azure Admin and create an Enterprise application and get the Delegated and Application permissions you need.** (in my case I have the needed permissions)
2. **Put your app secrets in your Ansible vault or your favorite secure location like Secret Server**
3. **Get with the ServiceNow folks get access to your Dev/Test/Sandbox environment and permission to access the Rest API Explorer its your friend, like me.**
4. **Get a ServiceNow service account setup with the correct permissions to create TASK**
Bonus points for getting impersonate rights in Dev/Test/Sandbox or full Admin.
Trust me it helps

Down in the weeds

Graph API task pieces

This is where you need your Azure Admin

The screenshot shows the Microsoft Azure portal interface for managing enterprise applications. A specific application named 'ansible_app' is selected. The portal displays several key pieces of information and configuration options:

- Essentials:** Shows the application's display name ('svc ansible msteams'), application (client) ID ('22a...5fae'), object ID ('...'), and directory (tenant) ID ('39c8...03').
- Supported account types:** My organization only.
- Information Bar:** Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory updates but we will no longer provide feature updates. Applications will need to be up
- Client secrets:** A tab labeled 'Certificates & secrets' is active, showing one existing client secret. The details for this secret are:
 - Description: app_teamssecret
 - Expires: 12/31
 - Value: **** (redacted)
 - Secret ID: f63

Annotations with yellow circles and numbers highlight specific elements:

- 1: Points to the browser address bar showing the URL for the Azure portal.
- 2: Points to the application name 'ansible_app' in the breadcrumb navigation.
- 3: Points to the 'app_teamsid' and 'tenantId' values in the Essentials section.
- 4: Points to the 'Certificates & secrets' tab in the Client secrets section.

Down in the weeds

Graph API task pieces

This is where you need your Azure Admin

The screenshot shows the Microsoft Azure portal interface. At the top left, there's a yellow circle with the number '1'. In the top right, there's a yellow circle with the number '2'. On the left sidebar, under the 'Manage' section, 'API permissions' is highlighted with a yellow circle containing the number '2'. A green arrow points from this '2' to the 'Add a permission' button in the main content area. In the main content area, there's a yellow circle with the number '3' pointing to the 'Add a permission' button. Another green arrow points from this '3' to a callout box containing the text 'get with your O365 admin to get permissions'. A yellow circle with the number '4' is located at the top right of the 'Microsoft Graph' card in the 'Commonly used Microsoft APIs' section.

Microsoft Azure

Search resources, services, and docs (G+)

Home > App registrations > ansible > API permissions

Overview Quickstart Integration assistant Manage Branding & properties Authentication Certificates & secrets Token configuration API permissions (2) Expose an API App roles Owners Roles and administrators Manifest Support + Troubleshooting

1

2

3

4

get with your O365 admin to get permissions

Request API permissions

Select an API

Microsoft APIs APIs my organization uses My APIs

Commonly used Microsoft APIs

Microsoft Graph

Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Microsoft Entra ID, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

Azure Communication Services

Rich communication experiences with the same secure CPaaS platform used by Microsoft Teams

Azure Cosmos DB

Fast NoSQL database with open APIs for any scale.

Azure Data Catalog

Programmatic access to Data Catalog resources to register, annotate and search data assets

Azure Data Explorer

Perform ad-hoc queries on terabytes of data to build near real-time and complex analytics solutions

Azure Data Lake

Access to storage and compute for big data analytic scenarios

Azure DevOps

Integrate with Azure DevOps and Azure DevOps server

Azure Key Vault

Manage your key vaults as well as the keys, secrets, and certificates within your Key Vaults

Azure Maps

Create location-aware web and mobile applications using simple and secure geospatial services, APIs, and SDKs in Azure.

Azure Rights Management Services

Allow validated users to read and write protected content

API / Permissions name	Type	Description
Application.Read.All	Delegated	Read applications
Application.Read.All	Application	Read all applications
Application.ReadWrite.All	Delegated	Read and write all applications
Application.ReadWrite.All	Application	Read and write all applications
Calendars.Read	Delegated	Read user calendars
Calendars.Read	Application	Read calendars in all mailboxes
Calendars.Read.Shared	Delegated	Read user and shared calendar items
Calendars.ReadWrite	Delegated	Have full access to user calendar items

Get a Token

To `curl://` or to `ansible.builtin.uri`

```
- name: Get Application Token for microsoft graph api
  ansible.builtin.shell: |
    curl -s -k -X POST -d "grant_type=client_credentials\
    &client_id={{ app_teamsid }}&client_secret={{ app_teamssecret }}&scope=https://graph.microsoft.com/.default" \
    https://login.microsoftonline.com/{{ tenantId }}/oauth2/v2.0/token \
    | jq -r ".access_token"
  register: shell_token
  changed_when: false

- name: Get Graph API Application Token
  ansible.builtin.uri:
    url: "https://login.microsoftonline.com/{{ tenantId }}/oauth2/v2.0/token"
    method: POST
    headers:
      Content-Type: "application/x-www-form-urlencoded"
    body_format: form-urlencoded
    body:
      grant_type: "client_credentials"
      client_id: "{{ app_teamsid }}"
      client_secret: "{{ app_teamssecret }}"
      scope: "https://graph.microsoft.com/.default"
  register: tokena
```

Microsoft Graph API

A screenshot of a search results page from a web browser. The search bar at the top contains the query "graph api mailfolders". Below the search bar is a navigation bar with tabs: ALL (which is selected), VIDEOS, IMAGES, NEWS, MAPS, SHOPPING, BOOKS, and SEARCH TOOLS. The main content area shows a single search result titled "List mailFolders - Microsoft Graph v1.0". The URL of the result is "learn.microsoft.com/en-us/graph/api/user-list-mailfolders?view=graph-rest-1.0&tabs=http". Below the title, there is a snippet of text: "Oct 27, 2023 · To return all mail folders in a mailbox, each child folder must be traversed separately. This API is available in the following national cloud ...". A green arrow points from the search bar down to the URL.

Permissions

Choose the permission or permissions marked as least privileged for this API. Use a higher privileged permission or permissions [only if your app requires it](#). For details about delegated and application permissions, see [Permission types](#). To learn more about these permissions, see the [permissions reference](#).

[Expand table](#)

Permission type	Least privileged permissions	Higher privileged permissions
Delegated (work or school account)	Mail.ReadBasic	Mail.ReadWrite, Mail.Read
Delegated (personal Microsoft account)	Mail.ReadBasic	Mail.ReadWrite, Mail.Read
Application	Mail.ReadBasic.All	Mail.ReadWrite, Mail.Read

HTTP request

To get all the mail folders in the root folder in the specified user's mailbox, excluding those that are hidden:

HTTP

```
GET /me/mailFolders  
GET /users/{id | userPrincipalName}/mailFolders
```

To include *hidden* mail folders in the response:

HTTP

```
GET /me/mailFolders/?includeHiddenFolders=true  
GET /users/{id | userPrincipalName}/mailFolders/?includeHiddenFolders=true
```

Optional query parameters

To return a list of all mailFolders including those that are hidden (their `isHidden` property is true), in the request URL, specify the `includeHiddenFolders` query parameter as `true`, as shown in the [HTTP request](#) section.

This method supports [OData query parameters](#) to help customize the response.

```
- name: "Get root folder of email "
  ansible.builtin.uri:
    url: "https://graph.microsoft.com/v1.0/users/{{ mailaccount }}/mailFolders"
    method: GET
    headers:
      Authorization: 'Bearer {{ token }}'
      Content-Type: application/json
    status_code:
      - 200
      - 201
      - 404
  register: root_folders
```

```
- name: show results of root folders
ansible.builtin.debug:
  msg:
    - "{{ root_folders }}"
  verbosity: "{{ verbosity_level }}"

- name: show results
ansible.builtin.debug:
  msg:
    - "{{ root_folders | json_query(query) }}"
  verbosity: "{{ verbosity_level }}"
vars:
  query: "json.value[?displayName=='Inbox'].id"
```

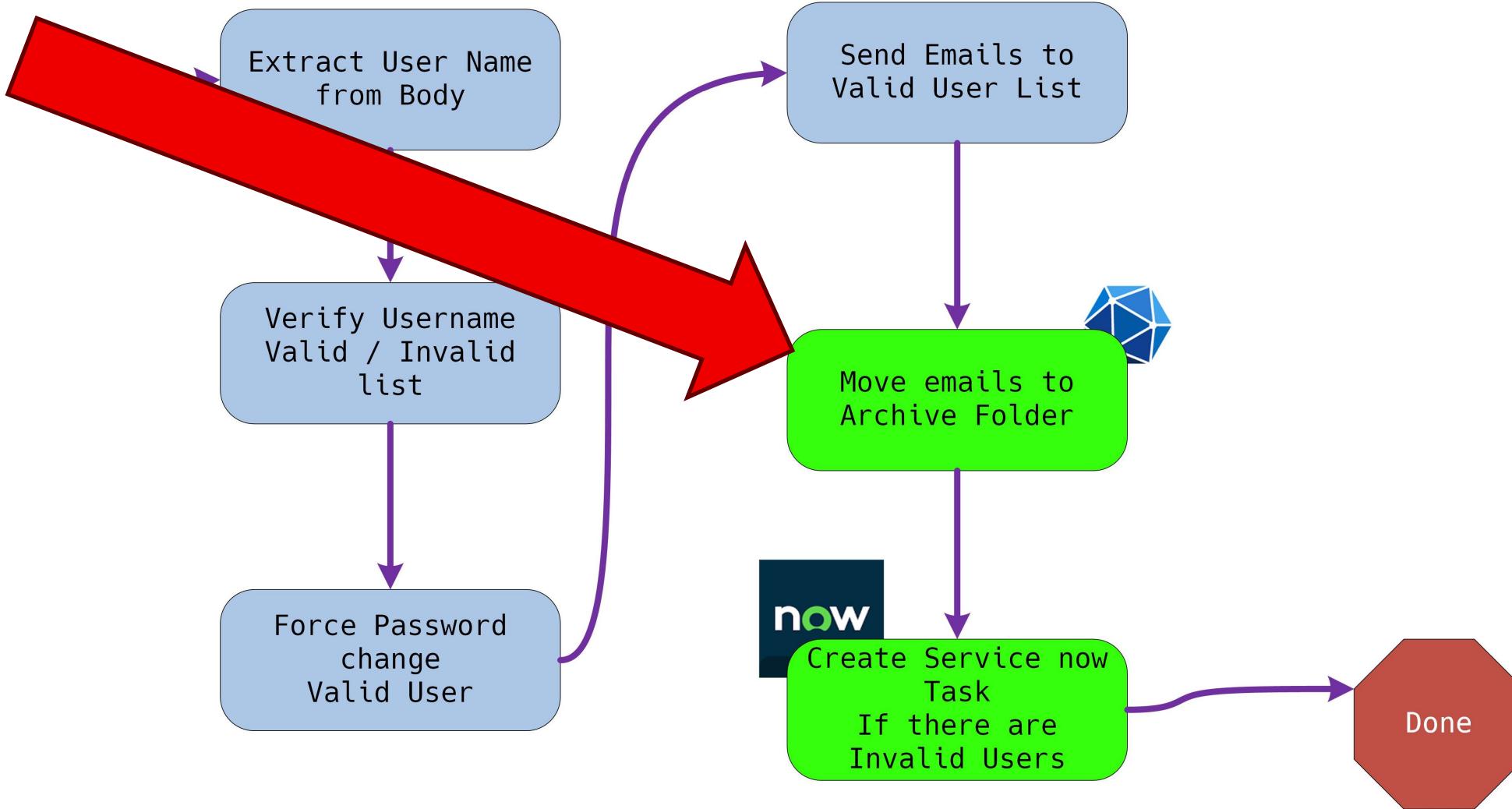
```
- name: set fact for root folder
ansible.builtin.set_fact:
  inbox: "{{ (root_folders | json_query(query))[0] }}"
vars:
  query: "json.value[?displayName=='Inbox'].id"
```

```
# subject_search: "CrowdStrike Identity Protection Policy Notification - Rule 'Compromised Password Detected'  
1  
- name: "Get Inbox email"  
  ansible.builtin.uri:  
    url: "{{ url_search }}"  
    method: GET  
    headers:  
      Authorization: 'Bearer {{ token }}'  
      Content-Type: application/json  
    status_code:  
      - 200  
      - 201  
      - 404  
      - 400  
  vars:  
    url_search: "https://graph.microsoft.com/v1.0/users/{{ mailaccount }}/mailFolders/{{ inbox }}/messages?  
    \\$filter=startswith(Subject,'{{ subject_search|urlencode }}')&$top=2000" 2  
  register: messages
```

The reason behind using \$top=2000 is because I am lazy, I didn't want to loop using the @odata.nextLink but if you do wind up having to do pagination Microsoft style this is now you do it

```
- set_fact:  
  nextpage: "graphapidataresult.json['@odata.nextLink']"
```

Mark the messages read and move them to the archive folder



Mark messages read and move to Archive folder

Set a fact to the email id's but do it as a batch of 100 messages to avoid a graph timeout

```
1 - name: set fact to emails id
  ansible.builtin.set_fact:
    | email_id: "{{ messages | json_query(queryid) }}"
  vars:
    queryid: "json.value[*].id"

2 - name: set fact to email list batch
  set_fact:
    | email_id_batch: "{{ email_id | batch(100) }}"

3 - name: include graph batch
  include_tasks:
    | tasks/graph_batch.yml
  loop: "{{ email_id_batch }}"

4 - name: Get Graph API Application Token
  ansible.builtin.uri:
    url: "https://login.microsoftonline.com/{{ tenantId }}/oauth2/v2.0/token"
    method: POST
    headers:
      Content-Type: "application/x-www-form-urlencoded"
    body_format: form-urlencoded
    body:
      grant_type: "client_credentials"
      client_id: "{{ app_teamsid }}"
      client_secret: "{{ app_teamssecret }}"
      scope: "https://graph.microsoft.com/.default"
    register: tokena
```

Mark messages read and move to Archive folder

Patch the mail record and mark it read

```
- name: "Mark messages read"
  ansible.builtin.uri:
    url: "{{ url_search }}"
    method: PATCH ①
    headers:
      Authorization: 'Bearer {{ token }}'
      Content-Type: application/json
    body_format: json
    body:
      isRead: true ⑤
    status_code:
      - 200
      - 201
      - 404
      - 400
  vars:
    url_search: "https://graph.microsoft.com/v1.0/users/{{ mailaccount }}/messages/{{ item_z }}"
    ignore_errors: true          #####
    until: message_patch is not failed # This is for unstable API's #
    retries: 20 ⑥                #
    delay: 60                     #####
    loop: "{{ item }}" ②
    loop_control:
      extended: true
      loop_var: item_z ③
    register: message_patch
```

1) Patch the mail id
2) loop our 100 messages
3) set a custom loop
4) item_z = our message id
5) mark the message read
6) if the message is failed
retry 20 times every minute

④

Mark messages read and move to Archive folder

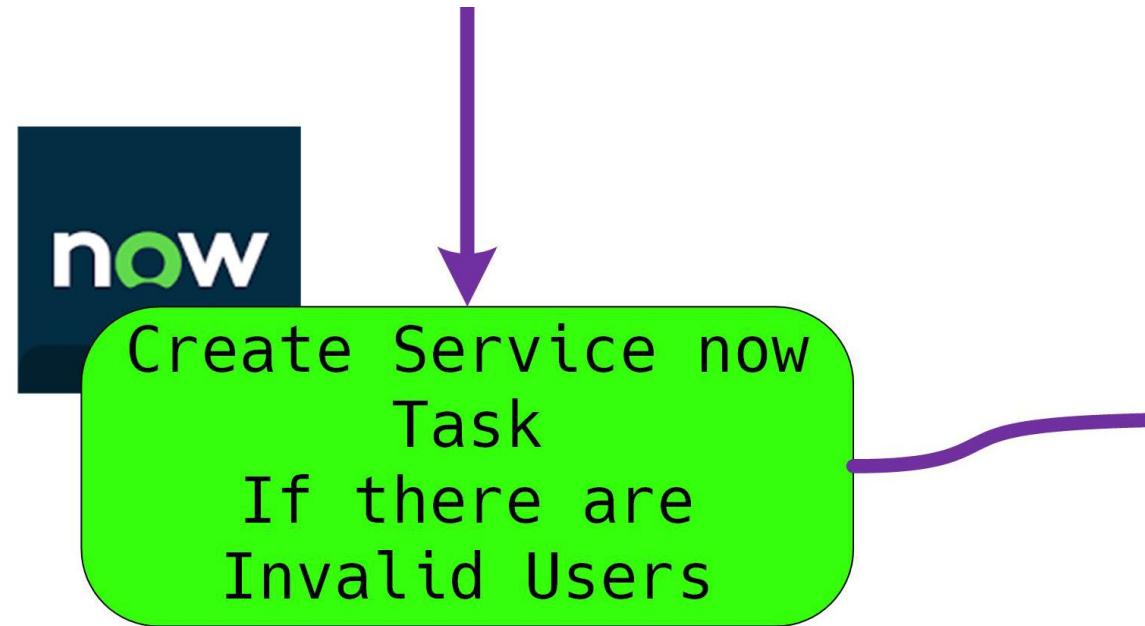
Move the message now

```
- name: "Move message to Archive "
  ansible.builtin.uri:
    url: "{{ url_move }}"
    method: POST
    headers:
      Authorization: 'Bearer {{ token }}'
      Content-Type: application/json
    body_format: json
    body:
      destinationId: "{{ (root_folders | json_query(destfolder))[0] }}"
    status_code:
      - 200
      - 201
      - 404
      - 400
  vars:
    url_move: "https://graph.microsoft.com/v1.0/users/{{ mailaccount }}/messages/{{ item_z }}/move"
    destfolder: "json.value[?displayName=='Archive'].id"
  register: message_move
  ignore_errors: true
  until: message_move is not failed
  retries: 20
  delay: 60
  loop: "{{ item }}"
  loop_control:
    loop_var: item_z
```

Diagram the issue

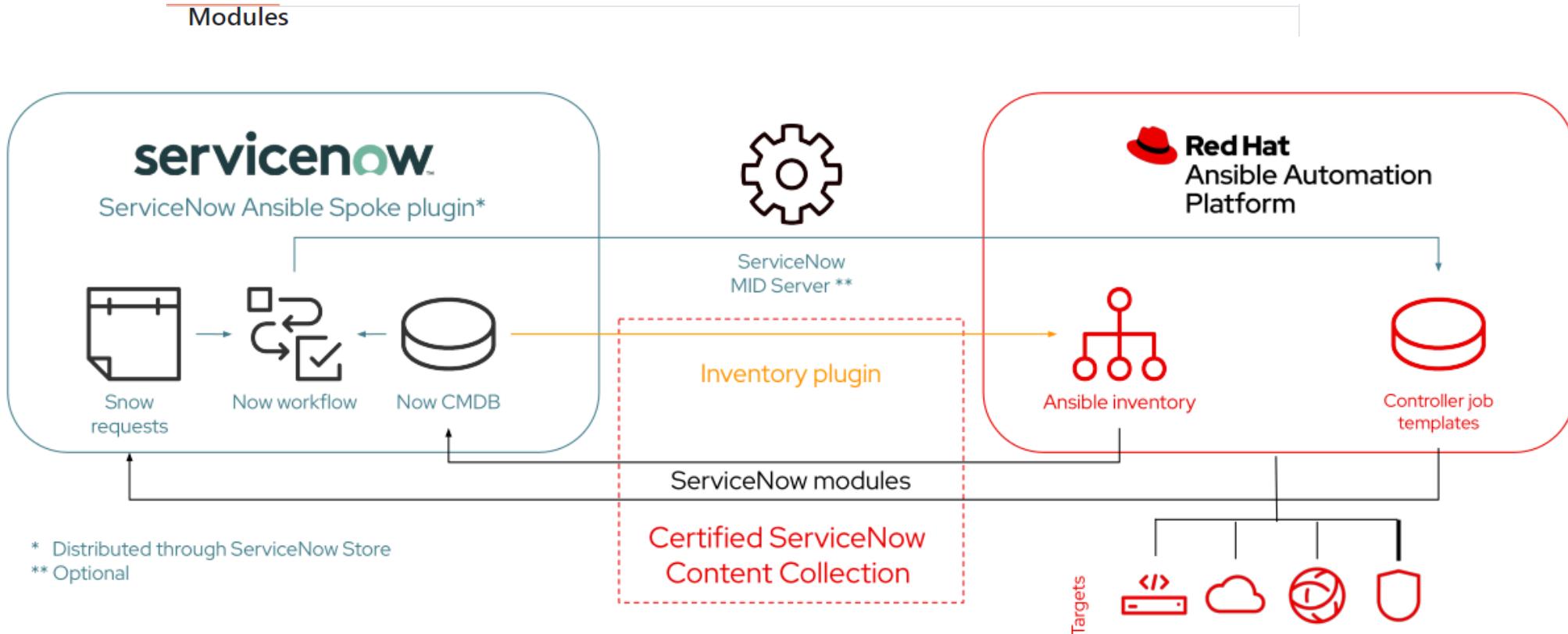
1,000 foot view

Lay it out, use your favorite diagram program I use Visio



ServiceNow, SNOW

servicenow.itsm vs. ansible.builtin.uri



servicenow.itsm.problem	Manage ServiceNow problems
servicenow.itsm.problem_info	List ServiceNow problems
servicenow.itsm.problem_task	Manage ServiceNow problem tasks
servicenow.itsm.problem_task_info	List ServiceNow problem tasks

Manage ServiceNow problems
List ServiceNow problems
Manage ServiceNow problem tasks
List ServiceNow problem tasks

ServiceNow – Its in front of you just Look

A screenshot of a ServiceNow Catalog Task record titled "Catalog Task TASK0518733". The record details are as follows:

- Number: TASK0518733
- Caller: Tyler Zimmerer (highlighted with a green box and circled 1)
- User: (highlighted with a green box and circled 2)
- Opened: 02-06-2024 17:54:22
- Opened by: (highlighted with a green box and circled 3)

Annotations with green boxes and arrows:

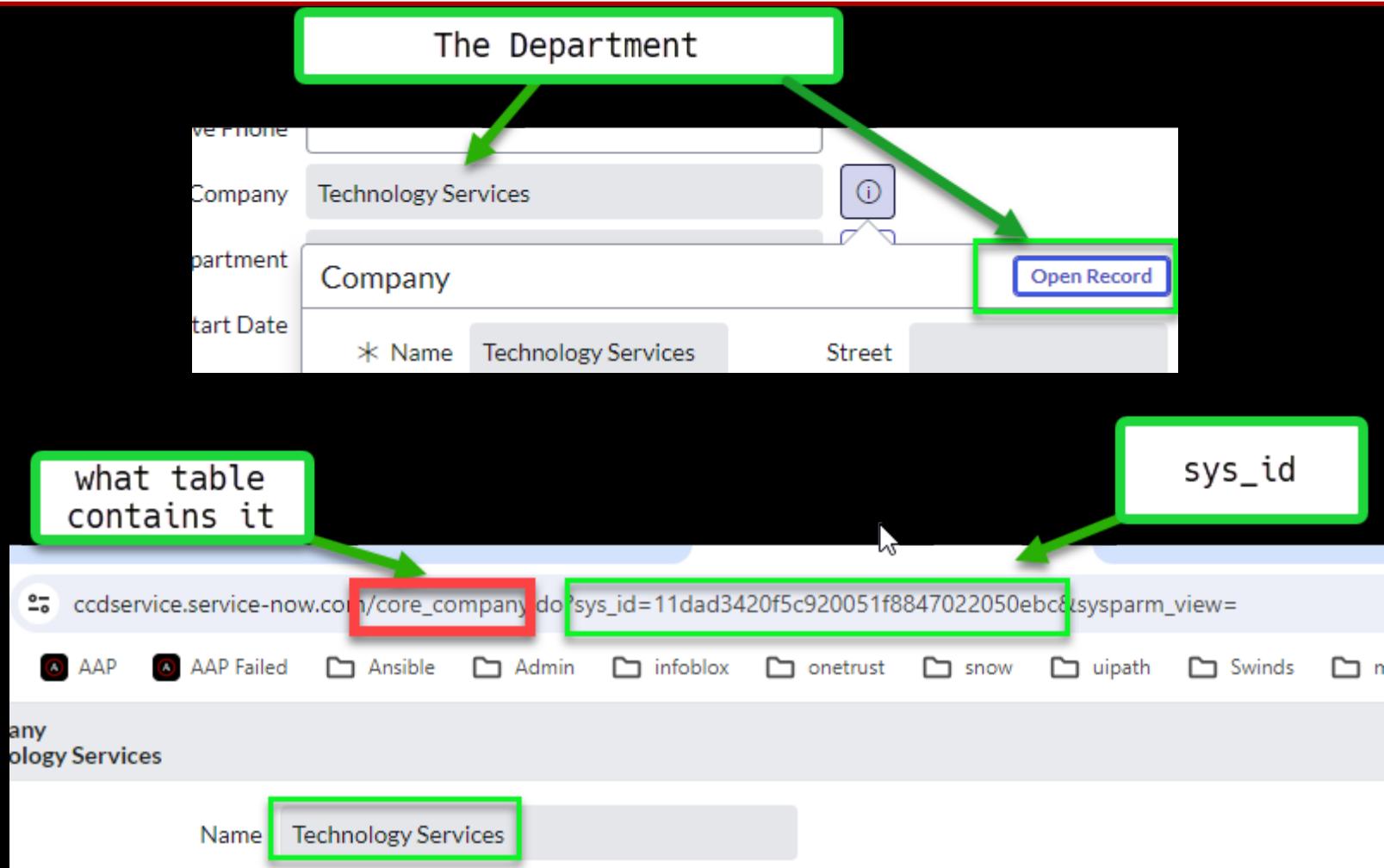
- "Thats the user" points to the "User" field.
- "open the record" points to the "Open Record" button.

A screenshot of a browser window showing the URL of a ServiceNow User record. The URL is:

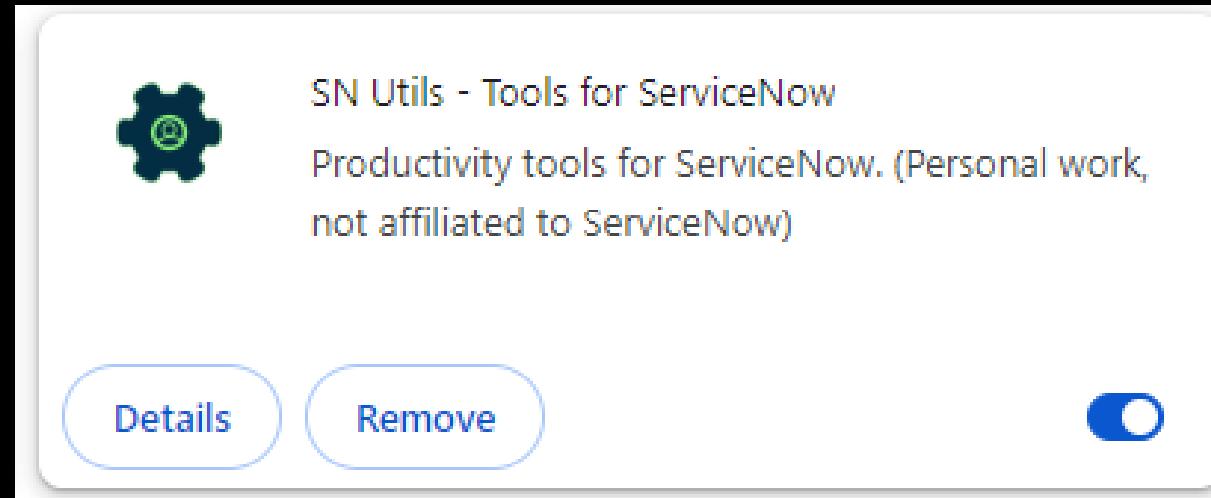
```
733 | Catalog Task | Se X c Tyler Zimmerer | User | ServiceN X +  
ccdservice.service-now.com/sys_user.do?sys_id=dd7647fedba67e00391c7c1ebf96193b&sysparm_view=
```

The browser tab shows "Tyler Zimmerer | User | ServiceN". The URL bar shows the full ServiceNow URL with the user ID and view parameter. The browser address bar also displays the URL.

ServiceNow – Its in front of you just Look



ServiceNow – There is a tool



ServiceNow – There is a tool

Catalog Task
TASK0532527 | sc_task [scratchpad][table fields][toggle label]

Discuss Update Add to Azure DevOps Close Task Start Microsoft Teams Chat

* Business Application | SR_Application ⓘ
GIS

* Sequence Number (000-999) | SR_Sequence ⓘ
023

* Operating System | SR_OperatingSystem ⓘ
Windows 2022 | Windows 2022

IP Address | SR_ip_address

* Patch Track | patch_track ⓘ
Patch Track 1 | Patch Track 1

PROD: TRACKS 1&2 Run concurrently during the 3rd week of the month (This is known as Track "A")
TRACKS 3&4 Run concurrently during the 3rd week of the month (This is known as Track "B")
TRACK OTHER Runs concurrently during the 3rd week and REQUIRES MANUAL REBOOT BY APPLICATION OWNER | prod_label

* Reboot times | reboot_times ⓘ
3:00 AM Reboot | 300

* Server Function | SR_ServerFunction ⓘ
Web server | WEB

* Server Environment | SR_ServerEnvironment ⓘ
QA | Q

Server Name | SR_ServerName
GISWEB023Q

Mic Drop

NON PROD: TRACK 1 occurs Thursday of 2nd week TRACK 2 occurs Friday of 2nd week TRACK 3 occurs Saturday of 2nd week TRACK 4 occurs Sunday of 2nd week TRACK OTHER Occurs Sunday of 2nd week and REQUIRES MANUAL REBOOT BY APPLICATION OWNER | nonprod_label

ServiceNow – Rest API Explorer – get a user

Table API

Allows you to perform create, read, update and delete (CRUD) operations on existing tables



Retrieve records from a table

```
GET https://ccdsandbox.service-now.com/api/now/table/{tableName}
```

Prepare request

Path parameters

Name	Value
* tableName	-- Select a table --

Query parameters

Name	Value	Description
sysparm_query	<input type="text"/>	An encoded query string used to filter the results
sysparm_display_value	<input type="text"/>	Return field display values (true), actual values (false), or both (all) (default: false)
sysparm_exclude_reference_link	<input type="text"/>	True to exclude Table API links for reference fields (default: false)
sysparm_suppress_pagination_header	<input type="text"/>	True to suppress pagination header (default: false)

ServiceNow – Rest API Explorer

REST API Explorer

Namespace: now

API Name: Table API

API Version: latest

Table API
Allows you to perform create, read, update and delete (CRUD) operations on existing tables

Retrieve records from a table
GET https://ccdsandbox.service-now.com/api/now/table/{tableName}

Prepare request

Create a record (POST) **(highlighted)**

Retrieve a record (GET)

Modify a record (PUT)

Delete a record (DELETE)

Update a record (PATCH)

Export OpenAPI Specification (YAML)

Export OpenAPI Specification (JSON)

Path parameters

Name: * tableName Value: Catalog Task (sc_task)

Query parameters

Name Value Description

sysparm_query

Return field display string used to filter the results

sysparm_display_value

Return field display values (true), actual values (false), or both (all) (default: false)

Request Body

Builder Raw

Active: true

Assignment Group: reference , Eg .. 9605e0c3c611227c017f67

you can see fields that are needed ***mostly**

```
{"active":"true","assignment_group":""}
```

ServiceNow – Rest API Explorer

The screenshot shows the ServiceNow Rest API Explorer interface. At the top left, there's a 'CURL Code Sample' section with a yellow circle containing the number '2'. Below it is a code block containing a cURL command to create a task in the 'sc_task' table. To the right of the code block are three buttons: 'Select Snippet', 'X', and a large 'Add header' button. At the bottom left, there's a 'Send' button and a 'Code Samples' section with a yellow circle containing the number '1'. The 'Code Samples' section contains a note about using provided snippets and links to other languages like ServiceNow Script, cURL, Python, Ruby, JavaScript, Perl, and Powershell.

CURL Code Sample

2

```
curl "https://ccdsandbox.service-now.com/api/now/table/sc_task" \
--request POST \
--header "Accept:application/json" \
--header "Content-Type:application/json" \
--data "{\"active\":\"true\",\"assignment_group\":\"\"}" \
--user 'admin':'admin'
```

Select Snippet X

Add header

Send

Code Samples

1

Use the provided code samples to send this request from commonly used languages.

[ServiceNow Script] [cURL] [Python] [Ruby] [JavaScript] [Perl] [Powershell]

ServiceNow – uri get user

```
- name: Search for user
  ansible.builtin.uri:
    url: "https://{{ snow_instance }}.service-now.com/api/now/table/sys_user?name={{ task_assignment_user|urlencode }}"
    method: GET
    user: "{{ Uname }}"
    password: "{{ Upass }}"
    headers:
      Accept: application/json
      Content-Type: application/json
    body_format: json
    status_code:
      - 200
      - 201
    register: snow_user_results
```

```
- name: set key facts based searches
  ansible.builtin.set_fact:
    assignedto: "{{ snow_user_results.json.result[0].sys_id }}"
    cmdbci: "{{ snow_cmdb_ci_results.json.result[0].sys_id }}"
    assignedgroup: "{{ snow_group_results.json.result[0].sys_id }}"
```

ServiceNow – uri create task

```
- name: Create Task
  ansible.builtin.uri:
    url: "https://{{ snow_instance }}.service-now.com/api/now/table/sc_task"
    method: POST
    body_format: json
    headers:
      Accept: "application/json"
      Content-Type: "application/json"
    body:
      1 active: "true"
      assigned_to: "{{ assignedto }}"
      u_ci_class: "{{ task_ci_class }}"
      assignment_group: "{{ assignedgroup }}"
      cmdb_ci: "{{ cmdbci }}"
      description: "{{ task_description }}"
      short_description: "{{ task_short_name }}"
      priority: "{{ task_priority }}"
      work_notes: "{{ task_work_notes }}"
      Approval: "approved"
      url_password: "{{ Upass }}"
      url_username: "{{ Uname }}"
      status_code: 201
      validate_certs: false
  register: create_task
```

ServiceNow – Make is a role in your collection

```
- name: Create Task for Security for accounts not found in AD
when: '(email_addresses_notfound |length | int ) > 0'
include_role:
  name: ccd.master_roles.snow_create_task          # if using automation hub
  public: true
vars:
  snow_instance: "{{ servicenow }}" ①             # snow instance to use
  task_short_name: "CS compromised accounts not found" ② # short description
  ③task_description: "Crowdstrike compromised accounts users not found: [code]{{ email_addresses_notfound |regex_replace(sqq,'<br>')|regex_replace(sq,'')|regex_replace('\\\\[','<br>') |regex_replace('\\\\]','<br>') }}[/code]" # long description
  ④task_assignment_group: "TS Information Security"      # assignment group if empty then gotten from user below
  task_cmdb_ci: "Crowdstrike" ⑤                      # ci aka SERVER
  task_ci_class: "Business Application" ⑥            # can be Server just take a look at snow drop down
  ⑦task_work_notes: "[code]{{ email_addresses_notfound |regex_replace(sqq,'<br>')|regex_replace(sq,'')|regex_replace('\\\\[','<br>') |regex_replace('\\\\]','<br>') }}[/code]"
  ⑧task_priority: "4" #1 = high 3 moderate 4 low
```



Thank you



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat

