MAYA (TWITTER ASSISTANT)

This n8n agent, named "Mega X Agent," acts as a **Twitter Assistant** called Maya. Its primary function is to manage and grow a professional Twitter account, specifically focusing on **cybersecurity content**. Maya delegates tasks to several specialized tools to achieve this:

How the Mega X Agent Works

The agent is triggered by **Slack mentions**, meaning it starts working when someone mentions it on a designated Slack channel. Here's a breakdown of its workflow:

- **Brain (AI Agent & Google Gemini Chat Model):** The "AI Agent" node, powered by the "Google Gemini Chat Model," is the core intelligence. It processes incoming requests from Slack and decides which tool to use based on the user's intent. It also leverages "Postgres Chat Memory" to remember past conversations and maintain context.

- **Tool Delegation:** Maya (the AI Agent) has access to four key tools:
    1. **Topic Finder:** This tool is used exclusively to **identify trending and relevant tweet topics related to cybersecurity**, drawing from trending tweets and news.
    2. **Tweeting:** Once a topic is approved by the user, this tool is responsible for **drafting and posting insightful and engaging tweets**. The agent will always seek user approval before posting.
    3. **Reply:** This tool is used to **fetch trending cybersecurity tweets, show them to the user, and then craft thoughtful replies** to those tweets. It also fetches the tweet ID, which is crucial for replying. User approval is required before posting any replies.
    4. **Engagement Metrics:** This tool allows Maya to **evaluate the performance of previous tweets**, providing data to optimize future content strategy.

- **Communication (Slack Integration):** After processing a request and interacting with its tools, the "AI Agent" sends its output back to the user via the "Send a message" node, which posts the response back to the specified Slack channel.

Key Operational Guidelines:

- **Simple Language:** The agent communicates in simple, conversational English, avoiding jargon unless necessary for a technical audience.
- **Originality and Consistency:** It prioritizes original, authentic content and maintains a consistent brand tone.

- **User Approval:** For crafting and posting tweets or replies, the agent will always seek approval from the user before taking action.

In essence, the Mega X Agent streamlines Twitter account management for cybersecurity content by intelligently delegating tasks to specialized AI tools, all while keeping the user in the loop for approvals.

Topic Finder Tool

How it Works:

1. **Trigger:** The workflow is activated when it receives a query from another workflow (likely the main "Mega X Agent" when it needs to find new topics).
2. **AI Agent (Brain):** The core intelligence is an "AI Agent" powered by the "Google Gemini Chat Model." This AI is specifically configured to:
    - **Focus on Cybersecurity:** It's explicitly told that it can *only* research cybersecurity topics, preventing it from asking the user for other preferences.
    - **Utilize Research Tools:** It has access to two powerful tools to gather information:
        - **Twitter API:** This tool connects to a Twitter API to fetch **trending tweets related to "Cybersecurity."** The query `"Cybersecurity" min_retweets:50 min_faves:100 min_replies:20` is designed to pull highly engaging and recent tweets, indicating trending discussions. The AI then processes these tweets to identify recurring trending topics.
        - **RSS Read:** This tool reads news from a specific RSS feed: `https://feeds.feedburner.com/TheHackersNews`. This provides the AI with **latest updates and news content directly from a reputable cybersecurity news source.**
    - **Summarize and Present:** After gathering data from both sources, the AI's instruction is to present both the trending topics from Twitter and the latest news from the RSS feed to the user in a **readable, summarized format, free of any technical metadata.**
3. **Memory:** A "Simple Memory" node helps the AI Agent maintain context throughout its research process.

Tweet

How it Works:

1. Trigger: The workflow is initiated when it's executed by another workflow (presumably the main "Mega X Agent" once a topic is approved and ready for tweeting).
2. AI Agent (Brain): The central "AI Agent," powered by the "Google Gemini Chat Model," is a helpful assistant tasked with:
    - Topic Reception: It receives a topic from the initiating workflow (implied by "check and retrieve topic for the user to tweet").
    - Research & Content Generation: It's responsible for researching the given topic and crafting the tweet content.
    - Approval & Posting: It must obtain user approval for the drafted tweet before posting it.
    - Record Keeping: It is mandated to update the Google Sheet with tweet details immediately after posting.
3. Internal Tools & Their Functions: The AI Agent has access to several specialized tools:
    - Tavily Research: This is an external web research tool (Tavily API) that the AI *must* use to gather information on the tweet topic before drafting content. This ensures the tweet is well-informed and accurate.
    - Learnings: This tool reads data from Google Sheet. This sheet contains past trending tweets, which the AI is instructed to analyze for inspiration on what kind of content performs well and to guide its tweet crafting.
    - Create Tweet in X: This tool directly interacts with the Twitter (X) API to post the crafted tweet to the professional Twitter account. It receives the tweet text from the AI.
    - Update Tweet Details: This is a crucial Google Sheets tool that updates the main "Sheet1" (the same sheet used by the "Topic Finder"). After a tweet is successfully posted, this tool *mandatorily* updates the row corresponding to the tweeted topic, marking its "Status" as "Done" and recording the "Content" (the tweet text) and the "Tweet ID" generated by Twitter. This update happens automatically without needing user confirmation.
4. Memory: A "Simple Memory" node helps the AI Agent maintain a short-term context of the conversation and the current tweeting task.

Key Operational Rules:

- Pre-Tweet Research: Tavily Research *must* be performed before drafting any tweet.
- Character Limit: Tweets must be strictly less than 260 characters.
- Learning from Past Success: The "Learnings" tool provides insights from past trending tweets to inform the current tweet's style and content.
- User Approval: The AI must present the drafted tweet to the user for approval before it is posted to X.

- Mandatory Update: Immediately after a tweet is posted, the Google Sheet must be updated with the topic, tweet content, and tweet ID. This is a non-negotiable, automatic step.

Reply

How it Works:

1. **Trigger:** The workflow is activated when it's called by another workflow (the main "Mega X Agent") with a query, likely when the user wants to engage with trending tweets.
2. **AI Agent (Brain):** The core intelligence is an "AI Agent" powered by the "Google Gemini Chat Model." Its primary responsibilities are:
    - **Iterative Tweet Presentation:** It fetches trending tweets one by one and presents them to the user. If the user rejects a tweet, it presents the next one, ensuring no repetition.
    - **Reply Crafting:** Once the user selects a tweet, the AI crafts a thoughtful reply.
    - **User Persona:** The AI is instructed to craft replies as a "seasoned yet young cybersecurity professional with 10+ years of experience in global organisations in middle east and India."
    - **Approval & Execution:** It seeks user approval for the crafted reply before posting.
    - **Comprehensive Logging:** It diligently logs details of the replied tweet and the reply content itself.
3. **Internal Tools & Their Functions:** The AI Agent utilizes several tools to achieve its goals:
    - **Trending Tweets:** This tool uses a third-party Twitter API to fetch current trending tweets relevant to "Cybersecurity." It applies filters (`min_retweets:50 min_faves:100 min_replies:20`) to ensure only highly engaging tweets are retrieved. Crucially, it provides the "Tweet ID" along with the tweet content, which is necessary for replying.
    - **Learned Content (Pinecone Vector Store):** This tool is a Pinecone vector store named "tweetlearning" that stores **past approved replies**. The AI is explicitly instructed to "understand the pattern from the data in the tool while crafting a reply to a tweet." This ensures replies are consistent with previous successful engagements.
    - **Learned Content (Google Sheets):** This is another Google Sheet named "Tweet Replies" used for storing "Learned Content" or analysis purposes
    - **Reply to Tweet in X:** This tool directly interacts with the Twitter (X) API to **post the crafted reply** to a specific tweet, using the "Tweet ID" provided by the "Trending Tweets" tool.
    - **Store Reply Details:** This Google Sheets tool appends new rows to "Sheet1" of the "Twitter Agent" Google Sheet. It records the "Tweet ID" of the original tweet, the "Content" (the crafted reply), sets the "Status" to "Done," and specifies the "Topic" as "Reply Tweet."

    - **Store Reply content for analysis:** This Google Sheets tool appends the crafted "Replies" content to a dedicated sheet named "Tweet Replies" in the "Twitter Agent" Google Sheet. This is likely for further analysis of reply effectiveness.
4. **Memory:** A "Simple Memory" node helps the AI Agent maintain a detailed conversation history and context, especially important for iterating through multiple trending tweets.

Key Operational Rules & Flow:

- **Iterative Tweet Selection:** The AI presents trending tweets one by one, along with their Tweet IDs, and avoids repetition.
- **Targeted Persona:** Replies are tailored to a specific professional persona.
- **Character Limits & No Hashtags:** Replies must be under 260 characters and should not include hashtags.
- **Learning from History:** Past successful replies from "Learned Content" guide the AI's reply generation style.
- **User Approval is Mandatory:** The crafted reply text *must* be approved by the user before it is posted.
- **Post-Reply Logging:** Immediately after a reply is posted, the tool automatically updates two Google Sheets: one for overall tweet details (`Store Reply Details`) and another specifically for reply content for analysis (`Store Reply content for analysis`).
- **Confirmation:** After all actions are complete, the user is informed that the reply has been posted and details stored

Engagement Metrics

How it Works:

1. **Trigger:** The workflow is initiated when it's called by another workflow (likely the main "Mega X Agent") with a query from the user. The user's query will specify what engagement metrics they are interested in.
2. **AI Agent (Brain):** The "AI Agent," powered by the "Google Gemini Chat Model," serves as the central intelligence of this workflow. Its main responsibilities are:
    - **Understanding User Queries:** It interprets the user's request for engagement metrics.
    - **Data Retrieval:** It uses its access to the "Engagement Metrics Sheet" tool to fetch the requested information.
    - **Information Provision:** It presents the retrieved engagement metrics to the user in a helpful and understandable manner.
3. **Internal Tools & Their Functions:**
    - **Engagement Metrics Sheet (Google Sheets):** This is the core data source. It's a Google Sheet that stores detailed engagement metrics for our previously posted tweets. The sheet is explicitly stated to contain the following columns:

        - **ID:** The unique identifier for each tweet.
        - **Retweets:** The number of times the tweet has been retweeted.
        - **Replies:** The number of replies the tweet has received.
        - **Likes:** The number of likes the tweet has garnered.
        - **Quotes:** The number of times the tweet has been quoted (retweeted with a comment).
        - **Views:** The total number of times the tweet has been viewed (impressions).
        - **Posted Time:** The timestamp of when the tweet was originally posted.
        - **URL:** The direct URL to the tweet.
        - **Content:** The actual text content of the tweet.
    - **Google Gemini Chat Model:** This is the large language model that drives the AI Agent's ability to understand natural language queries and formulate responses.
    - **Simple Memory:** This node helps the AI Agent maintain context throughout the conversation with the user, allowing for more fluid and coherent interactions, especially when discussing multiple tweets or metrics.
    - **When Executed by Another Workflow:** This node acts as the entry point, allowing the "X Engagement Metrics" workflow to be called and integrated into a larger system (the "Mega X Agent").

Key Operational Rule:

- **Data Provision:** The AI Agent's primary goal is to provide the user with the requested engagement metrics from the "Engagement Metrics Sheet." It acts as an information retrieval system for historical tweet performance.

In essence, the "X Engagement Metrics" tool acts as a dedicated analytics reporting system for tweets. It allows a user to query historical performance data for their tweets, providing insights into their content's reach and interaction.

Tweet Analyser

.

This n8n workflow, named "Tweets Analyser," is an **automated system designed to periodically fetch and store engagement metrics for tweets** in a Google Sheet. The data stored by this tool is available through Maya

How it Works:

1. **Trigger ("Google Sheets Trigger"):** This workflow is set to **trigger every minute**. It monitors a specific Google Sheet . It seems to be looking for new rows or changes in this sheet, specifically expecting "Tweet ID" values to be present.
2. **Wait:** After being triggered, the workflow introduces a **24-hour delay** using the "Wait" node. This suggests that the engagement metrics for a tweet are collected one day after the tweet's ID appears in the trigger sheet, allowing time for initial engagement to accumulate.
3. **HTTP Request ("Get Tweet Data"):** After the 24-hour wait, this node makes an **API call to a Twitter API endpoint** (`https://api.twitterapi.io/twitter/tweets`).
    - It uses the `tweet_ids` query parameter, taking the "Tweet ID" directly from the data that triggered the workflow (e.g., the new row in "Sheet1").
    - The purpose of this call is to fetch the current, real-time engagement metrics for the specified tweet(s).
4. **Append row in sheet ("Store Engagement Metrics"):** The final step is to **append a new row to another Google Sheet** (specifically "Sheet3" of the "Twitter Agent" spreadsheet, `1btnAsLtaNminq-eXzc_jkYInpXeaq9ZoBwNAPfMA4Ic`). This sheet is likely the "Engagement Metrics Sheet" referred to by the "X Engagement Metrics" tool.
    - It populates the columns of this sheet with the data retrieved from the Twitter API call:
        - **ID:** Tweet ID (`$json.tweets[0].id`)
        - **Retweets:** Retweet count (`$json.tweets[0].retweetCount`)
        - **Replies:** Reply count (`$json.tweets[0].replyCount`)
        - **Likes:** Like count (`$json.tweets[0].likeCount`)
        - **Quotes:** Quote count (`$json.tweets[0].quoteCount`)
        - **Views:** View count (`$json.tweets[0].viewCount`)
        - **Posted Time:** Creation timestamp (`$json.tweets[0].createdAt`)
        - **Url:** The tweet's content (`$json.tweets[0].text`) - **Note:** There seems to be a mapping error here. `Url` should likely be `$json.tweets[0].twitterUrl` and `Content` should be `$json.tweets[0].text`.
        - **Content:** The tweet's URL (`$json.tweets[0].twitterUrl`) - **Note:** This also seems like a mapping error as stated above.

Key Operational Flow:

- **Automated Monitoring:** The workflow continuously monitors a Google Sheet for new tweet IDs.
- **Delayed Data Collection:** It waits 24 hours before fetching engagement metrics, assuming this delay allows for a significant portion of initial tweet engagement to occur.
- **API Integration:** It uses a Twitter API to get detailed engagement statistics for specific tweets.

Automated Twitter Workflow

Automatic and Manual (Maya) Twitter workflows will be working parallely. The "Tweet Creator" is a sophisticated n8n workflow designed to **automatically generate and publish original tweets, as well as intelligent replies to trending cybersecurity discussions**, all while maintaining a consistent and professional brand voice. This tool leverages advanced AI models and integrates with Twitter (now X) and Google Sheets to streamline social media engagement for a cybersecurity thought leader.

How it Works: Two Main Workflows

This tool appears to have two primary, distinct, and parallel workflows, both aimed at automating Twitter activity:

1. **Automated Tweet Generation from News Articles**
2. **Automated Tweet Replies to Trending Discussions**

---

1. Automated Tweet Generation from News Articles

This branch of the workflow focuses on creating original tweets based on trending cybersecurity news.

- **Triggers (RSS Feeds):**
    - **"RSS Feed Trigger"** and **"RSS Feed Trigger1"**: These nodes are configured to check RSS feeds from prominent cybersecurity news sources:
        - `https://feeds.feedburner.com/TheHackersNews`
        - `https://cyberscoop.com/feed/`
        - They poll daily at 11:01 PM (23:01). This suggests the tool aims to get the latest news at the end of the day to prepare content for the next.
- **"Edit Fields":** This node extracts the `contentSnippet` from the RSS feed, which likely contains the summary or main text of the articles. This is then passed as `text` to the next stage.
- **"AI Agent" (Tweet Drafting):** This is the core of the tweet generation.
    - It takes the content of up to **three articles** as input (Article 1, Article 2, Article 3).
    - It uses a **Google Gemini 2.5 Flash model** ("Google Gemini Chat Model") to generate the tweet.
    - The **system message** provides highly detailed instructions for the AI:
        - **Persona:** A "skilled Twitter content creator" who writes like a human, with clarity, emotional intelligence, and an engaging tone. This persona also

has 10 years of experience in cybersecurity and has worked in the Middle East and India.
        - **Task:** Summarize key insights from the three articles into a **single tweet**.
        - **Style:** Human-written (no em-dashes), professional yet personal, warm, conversational, and no jargon or robotic phrasing. It should sound like it's from a thought leader.
        - **Length:** 1-3 sentences, **never exceeding 200 characters** (with a stricter reminder of 220 characters).
        - **Audience:** Working professionals who value authenticity and insight.
        - **Constraints:** Do not mention "Article 1", "Article 2", etc. Integrate insights naturally. No hashtags.
        - **Learning:** The AI is instructed to "Learn from all the tweets that are present in that source" using a "Memory" tool (though a direct "Memory" node connection isn't explicitly visible in this path, it implies an underlying knowledge base for tweet style).
- **"X" (Tweet Posting):** This node takes the `Content` generated by the AI Agent and posts it directly to the connected X (Twitter) account.
- **"Update row in sheet1":** This node updates a Google Sheet (Sheet1 of the "Twitter Agent" spreadsheet, `1btnAsLtaNminq-eXzc_jkYInpXeaq9ZoBwNAPfMA4Ic`) to log the created tweet.
    - It marks the `Status` of the original "Topic" as "Done".
    - It records the `Content` (the tweet text) and the `Tweet ID` of the newly posted tweet.
    - The `Topic` is pulled from the triggering row in the Google Sheet, indicating that this process likely creates tweets for topics that were previously identified as "Not Done."

---

2. Automated Tweet Replies to Trending Discussions

This parallel workflow is designed to find trending cybersecurity tweets and automatically craft replies.

- **Triggers (Schedule & Twitter Scrapping):**
    - **"Schedule Trigger1" and "Schedule Trigger3":** These nodes initiate the process every 12 hours.
    - **"Twitter Scrapping" and "Twitter Scrapping1":** These HTTP Request nodes query the Twitter API for trending tweets related to cybersecurity:
        - `"Cybersecurity" min_retweets:50 min_faves:100 min_replies:20` (from "Twitter Scrapping")
        - `"Information security" min_retweets:50 min_replies:20` (from "Twitter Scrapping1")

        - They specifically look for **"Latest"** tweets with high engagement (minimum retweets, likes, and replies). This ensures the tool interacts with active discussions.
- **"Code":** This node processes the scraped tweet data.
    - It **filters out non-English tweets and tweets containing "$"** (likely to avoid irrelevant content or financial spam).
    - It extracts key information like `text`, `url`, `userName`, `screenName`, `retweetCount`, `replyCount`, `likeCount`, `quoteCount`, `viewCount`, `createdAt`, and `lang`.
    - It also formats the `createdAt` timestamp.
- **"Append row in sheet" (Memory Storage):** This node appends the details of the scraped, high-engagement tweets to a Google Sheet (Sheet2 of the "Twitter Agent" spreadsheet). This sheet likely serves as the "Memory" referred to in the AI Agent prompts, storing examples of successful or relevant tweets for the AI to learn from.
- **"AI Agent1" (Trending Topic Analysis):**
    - This AI Agent takes the aggregated text of the trending tweets.
    - It's tasked with analyzing these tweets and **suggesting "exactly three trending topics."**
    - This step helps identify the most relevant themes from the scraped data.
- **"Information Extractor":** This node extracts specific `Topic` information from the output of "AI Agent1." It's configured to extract "A topic other than just cybersecurity," suggesting a refinement to identify more specific sub-topics.
- **"Code1":** This node processes the extracted topics, preparing them for the next step.
- **"Append row in sheet1" (Topic Storage):** The extracted topics are then appended to Sheet1 of the "Twitter Agent" spreadsheet, with their `Status` initially set to "Not Done." This creates a queue of topics for which new tweets might be generated.
- **"Schedule Trigger" (Disabled):** This trigger is currently disabled, but if active, it would initiate the main tweet creation process periodically (every 12 hours).
- **"Get row(s) in sheet":** This node retrieves a row from Sheet1 where the `Status` is "Not Done." This ensures the workflow picks up a new topic to tweet about.
- **"AI Agent2" (Reply Generation):**
    - This AI Agent is specialized in crafting tweet replies. It takes the `content` of a tweet to reply to as input.
    - It uses an **OpenAI Chat Model (gpt-4.1-mini)** for generation.
    - The **system message** defines its persona and task:
        - **Persona:** A "young yet seasoned cybersecurity professional" with 10+ years of multinational experience in threat intelligence, incident response, security architecture, and emerging technologies. The tone is confident, concise, professional with a "modern swagger," combining elite technical insight with approachability.
        - **Task:** Reply to tweets like a human expert—insightful, engaging, sometimes witty, never condescending.

        - **Style:** Add value (practical insight, expert opinion, resources), maintain respectful and community-driven tone, avoid promotion, use clear human-like sentences (no em-dashes), **never exceed 220 characters**, and **no hashtags**.
        - **Learning:** Similar to the other AI Agent, it's instructed to use the "Memory" tool (the Google Sheet where trending tweets are saved) to get an idea of how reply tweets should be styled.
- **"Create Tweet":** This node takes the generated reply from "AI Agent2" and posts it to X (Twitter) as a reply, using the `inReplyToStatusId` to link it to the original tweet's URL.
- **"Update row in sheet":** This node updates the Google Sheet (Sheet1) to mark the `Status` of the "Topic" as "Done" after a reply has been successfully generated and posted. It also records the `Content` (the reply tweet text) and the `Tweet ID` of the reply.