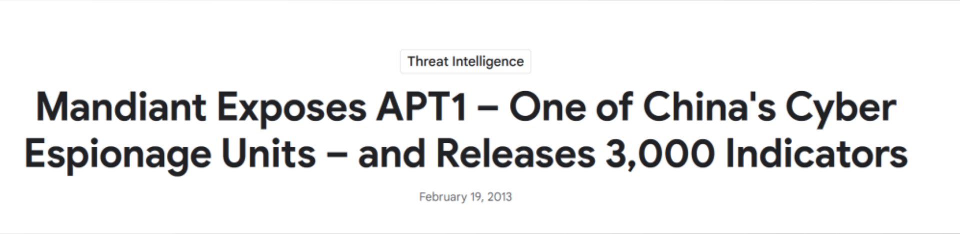
From Data Deluge to actionable Insights with LLMs: Introducing "TI Mindmap"

Antonio Formato
June 29, 2024

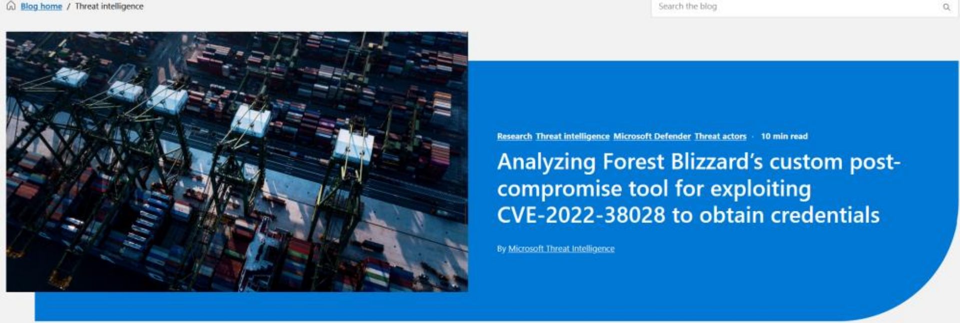
Analyzing open-source intelligence and generating reports, as well as extracting immediate value from write-ups, takes hours each day.


Threat Intelligence

Mandiant Exposes APT1 – One of China's Cyber Espionage Units – and Releases 3,000 Indicators

February 19, 2013

https://cloud.google.com/blog/topics/threat-intelligence/mandiant-exposes-apt1-chinas-cyber-espionage-units/


Blog home / Threat intelligence

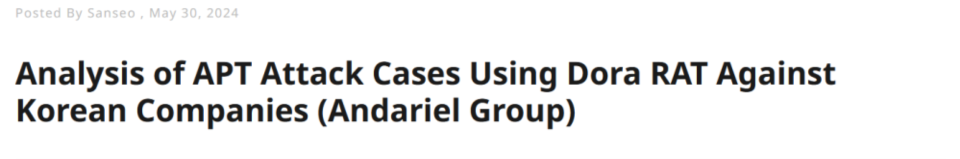
Search the blog

Research Threat intelligence Microsoft Defender Threat actors · 10 min read

Analyzing Forest Blizzard's custom post-compromise tool for exploiting CVE-2022-38028 to obtain credentials

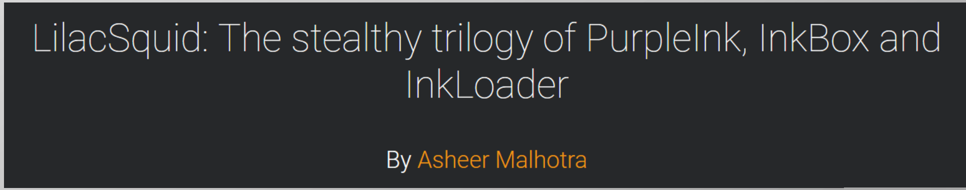
By Microsoft Threat Intelligence

https://www.microsoft.com/en-us/security/blog/2024/04/22/analyzing-forest-blizzards-custom-post-compromise-tool-for-exploiting-cve-2022-38028-to-obtain-credentials/


Posted By Sanseo , May 30, 2024

Analysis of APT Attack Cases Using Dora RAT Against Korean Companies (Andariel Group)

https://asec.ahnlab.com/en/66088/


LilacSquid: The stealthy trilogy of PurpleInk, InkBox and InkLoader

By Asheer Malhotra

https://blog.talosintelligence.com/lilacsquid/

Who am I?

Antonio Formato

Cybersecurity Technical Specialist @ Microsoft

https://medium.com/@antonio.formato

@anformato

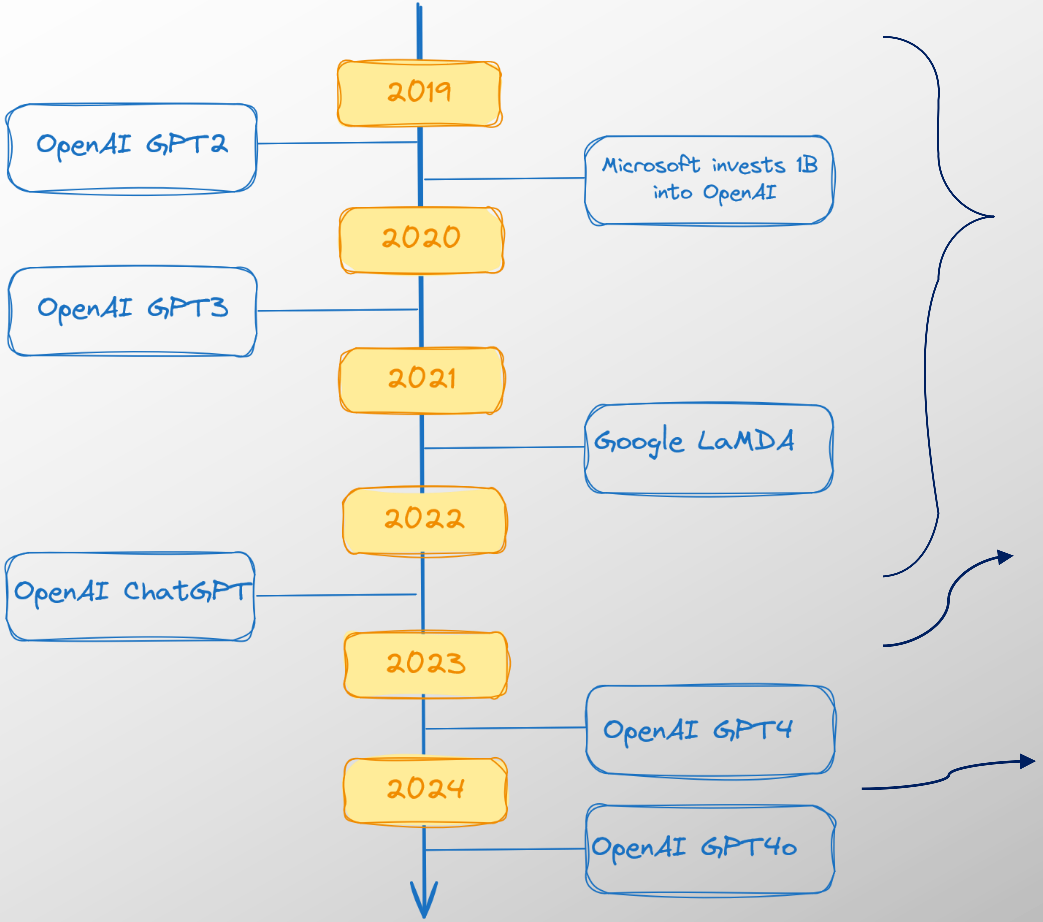
https://www.linkedin.com/in/antonioformato/

Special mentions and acknowledgments:

Oleksyi Meletski main project contributor

Thomas Roccia source of inspiration for the project

LLMs: More than Just Chatbots!



2019

OpenAI GPT2

Microsoft invests 1B into OpenAI

2020

OpenAI GPT3

2021

Google LaMDA

2022

OpenAI ChatGPT

2023

OpenAI GPT4

2024

OpenAI GPT4o

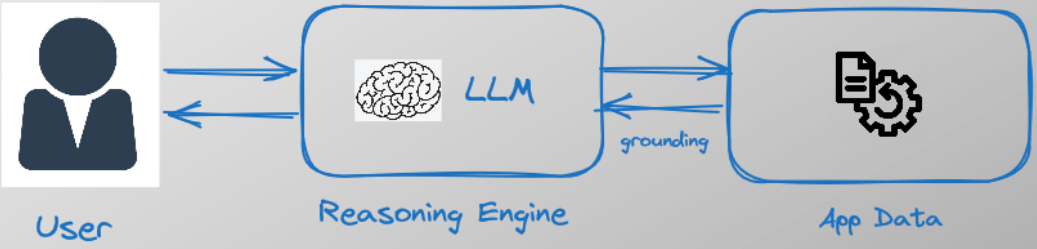
The timeline is not exhaustive and does not claim to be complete.

LLM opportunity

> LLMs are more than language generators, they can be seen as reasoning engines becoming brains of apps

> LLMs are not only for conversational tasks, they can also:

- > Extract structured data from unstructured text or images
- > Generate synthetic structured data or unstructured data
- > Help humans make decisions
- > Make decisions and interact with other systems
- > Grasp general patterns and relationships within the data
- > Not just text. Foundational understanding across different domains

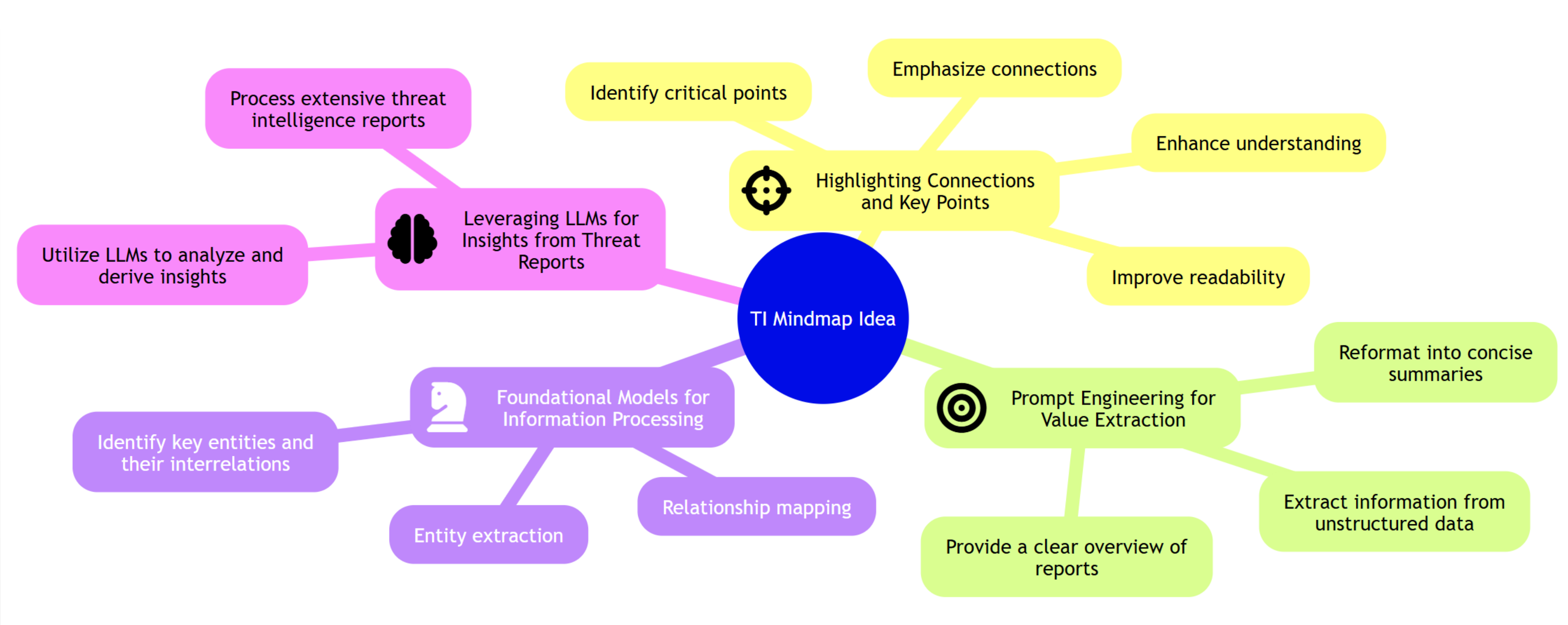


Why LLMs in Cyber Threat Intelligence

- Semantic Analysis: 🔍
  - Understand threat reports more in depth
  - Uncover context and relationships
- Bridging Analyst Skill Gaps: 🌉
  - Technical and strategic insights
  - Operate in both realms effectively
- Entity Extraction: 🗂️
  - Identify and catalog key entities
- Summarization: ✂️
  - Concise summaries of lengthy reports
- Automated Report Generation: 📝
  - Create human-readable threat reports
- Vulnerability Analysis: 🔧
  - Extract info on vulnerabilities
- Knowledge Base Enrichment: 📚
  - Update with latest threat info
- Anomaly Detection: ⚠️
  - Spot unusual patterns
- Pattern Recognition: 🔄
  - Predict future threats
- Language Translation: 🌐
  - Facilitate global collaboration

https://www.rsaconference.com/library/blog/how-large-language-models-are-changing-threat-intelligence-report-analysis
https://socradar.io/how-can-open-source-llms-be-used-in-cti/

TI Mindmap Idea decoded with Mermaid.js



TI Mindmap Idea

Leveraging LLMs for Insights from Threat Reports
- Process extensive threat intelligence reports
- Utilize LLMs to analyze and derive insights

Highlighting Connections and Key Points
- Identify critical points
- Emphasize connections
- Enhance understanding
- Improve readability

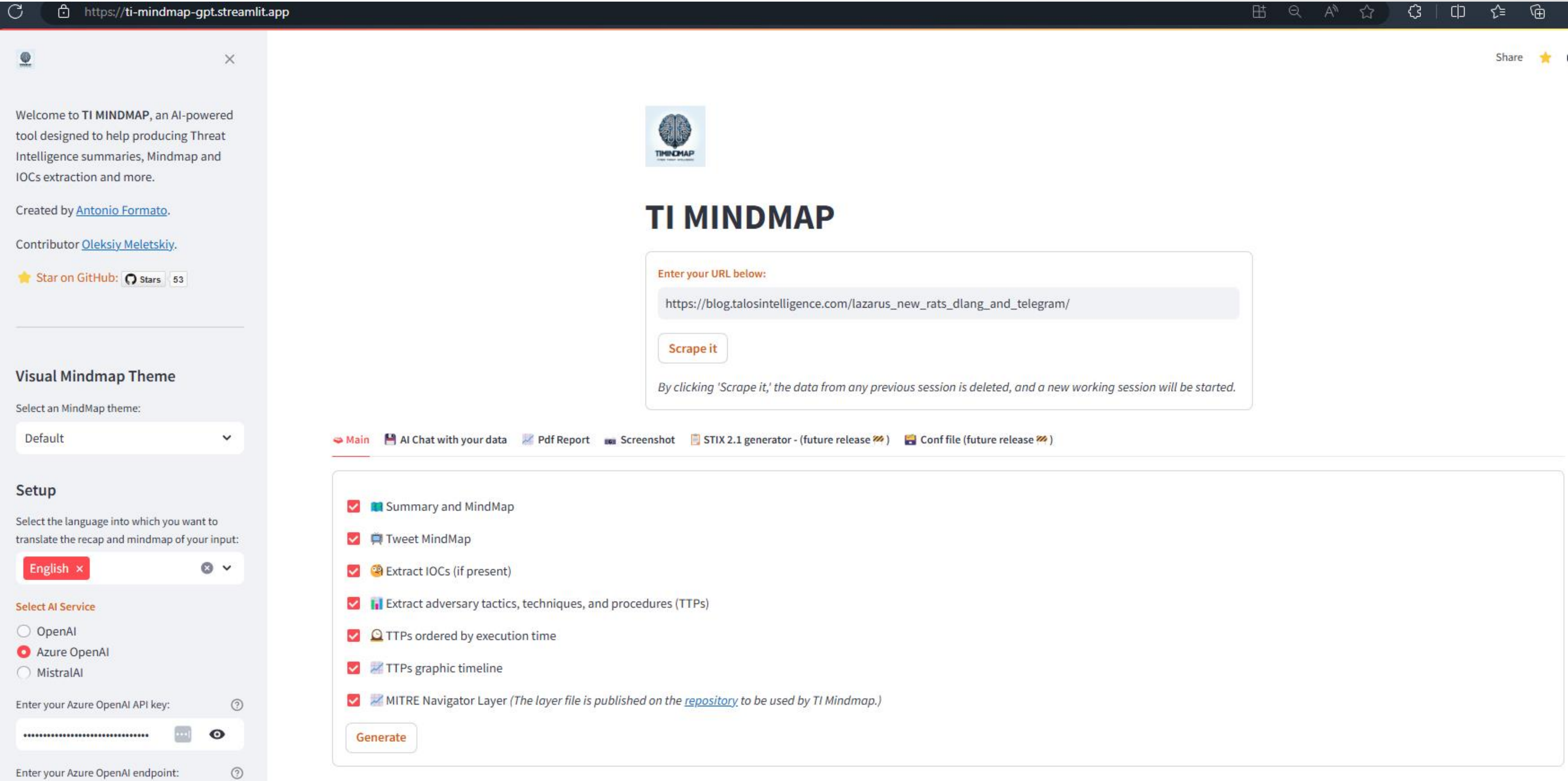
Foundational Models for Information Processing
- Identify key entities and their interrelations
- Entity extraction
- Relationship mapping

Prompt Engineering for Value Extraction
- Reformat into concise summaries
- Extract information from unstructured data
- Provide a clear overview of reports

What TI Mindmap is

- Open-source Python project
- Cyber Threat Intelligence Analysis prototype app
- Tool powered by Streamlit
- LLMs (OpenAI, Azure OpenAI, MistralAI) to:
  - 🧠Interpret complex cyber threat data
  - 🗺️Create Mindmap
  - 📝Generate summaries and insights
  - 🔍Extracts IOCs (VT enrichment)
  - 📈Extract and track TTPs overtime
  - 🗂️ Generate MITRE Navigator Layer
  - 🤖AI chat on threat reports
  - 📄Provide pdf report
  - 🐦Tweet your Mindmap

TI Mindmap App – https://ti-mindmap-gpt.streamlit.app/



Welcome to **TI MINDMAP**, an AI-powered tool designed to help producing Threat Intelligence summaries, Mindmap and IOCs extraction and more.

Created by Antonio Formato.

Contributor Oleksiy Meletskiy.

Star on GitHub: Stars 53

Visual Mindmap Theme

Select an MindMap theme: Default

Setup

Select the language into which you want to translate the recap and mindmap of your input: English

Select AI Service

○ OpenAI
● Azure OpenAI
○ MistralAI

Enter your Azure OpenAI API key: ••••••••••••••••••••••••••••••••

Enter your Azure OpenAI endpoint:

TI MINDMAP

Enter your URL below: https://blog.talosintelligence.com/lazarus_new_rats_dlang_and_telegram/

By clicking 'Scrape it,' the data from any previous session is deleted, and a new working session will be started.

Main
AI Chat with your data
Pdf Report
Screenshot
STIX 2.1 generator - (future release)
Conf file (future release)

☑ Summary and MindMap
☑ Tweet MindMap
☑ Extract IOCs (if present)
☑ Extract adversary tactics, techniques, and procedures (TTPs)
☑ TTPs ordered by execution time
☑ TTPs graphic timeline
☑ MITRE Navigator Layer *(The layer file is published on the repository to be used by TI Mindmap.)*

Mindmap Example

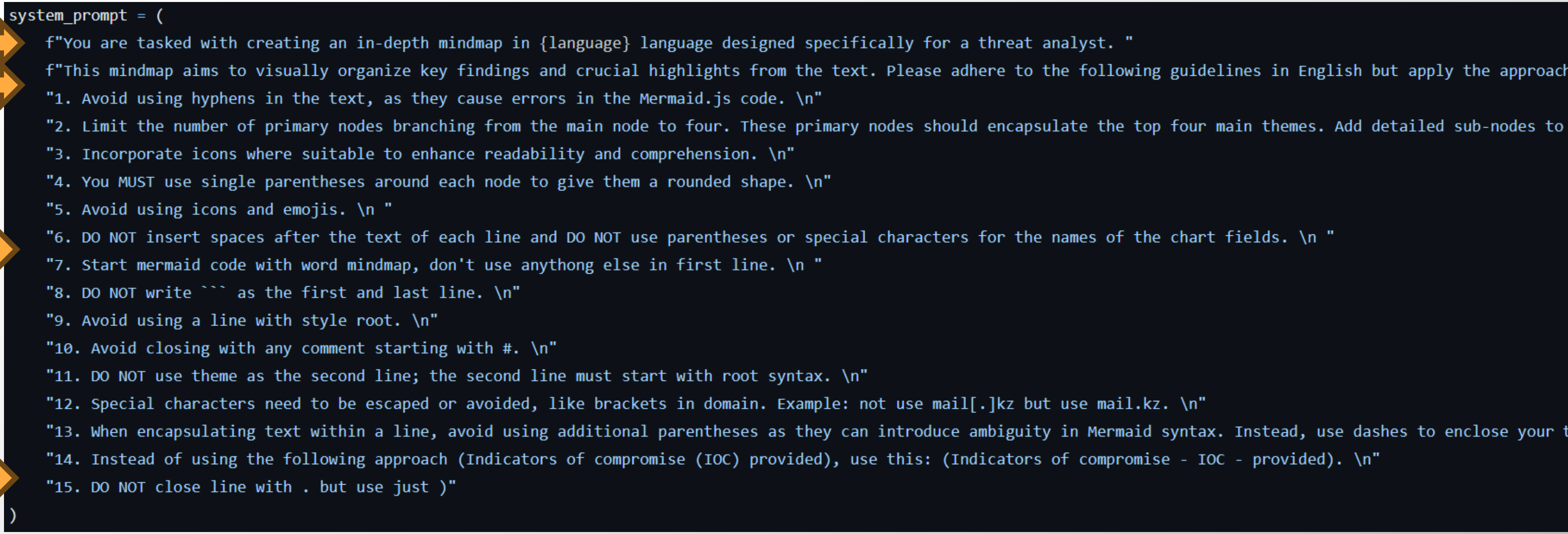
Metaprompt

Instructions

Task-specific context

Supporting information

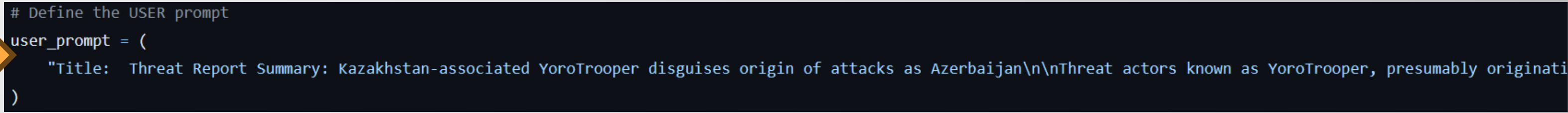
Model's output format



```
system_prompt = (
    f"You are tasked with creating an in-depth mindmap in {language} language designed specifically for a threat analyst. "
    f"This mindmap aims to visually organize key findings and crucial highlights from the text. Please adhere to the following guidelines in English but apply the approac
    "1. Avoid using hyphens in the text, as they cause errors in the Mermaid.js code. \n"
    "2. Limit the number of primary nodes branching from the main node to four. These primary nodes should encapsulate the top four main themes. Add detailed sub-nodes to
    "3. Incorporate icons where suitable to enhance readability and comprehension. \n"
    "4. You MUST use single parentheses around each node to give them a rounded shape. \n"
    "5. Avoid using icons and emojis. \n "
    "6. DO NOT insert spaces after the text of each line and DO NOT use parentheses or special characters for the names of the chart fields. \n "
    "7. Start mermaid code with word mindmap, don't use anythong else in first line. \n "
    "8. DO NOT write ``` as the first and last line. \n"
    "9. Avoid using a line with style root. \n"
    "10. Avoid closing with any comment starting with #. \n"
    "11. DO NOT use theme as the second line; the second line must start with root syntax. \n"
    "12. Special characters need to be escaped or avoided, like brackets in domain. Example: not use mail[.]kz but use mail.kz. \n"
    "13. When encapsulating text within a line, avoid using additional parentheses as they can introduce ambiguity in Mermaid syntax. Instead, use dashes to enclose your t
    "14. Instead of using the following approach (Indicators of compromise (IOC) provided), use this: (Indicators of compromise - IOC - provided). \n"
    "15. DO NOT close line with . but use just )"
)
```

User prompt

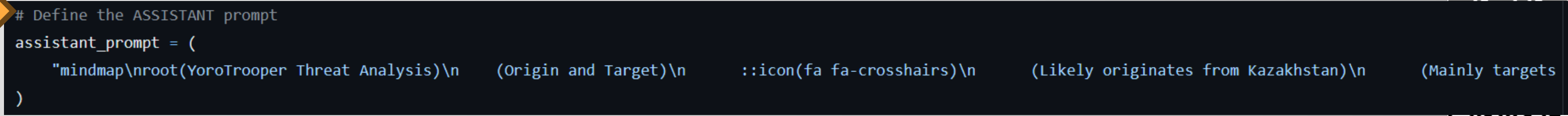
Example of user inputs



```
# Define the USER prompt
user_prompt = (
    "Title:  Threat Report Summary: Kazakhstan-associated YoroTrooper disguises origin of attacks as Azerbaijan\n\nThreat actors known as YoroTrooper, presumably originati
)
```

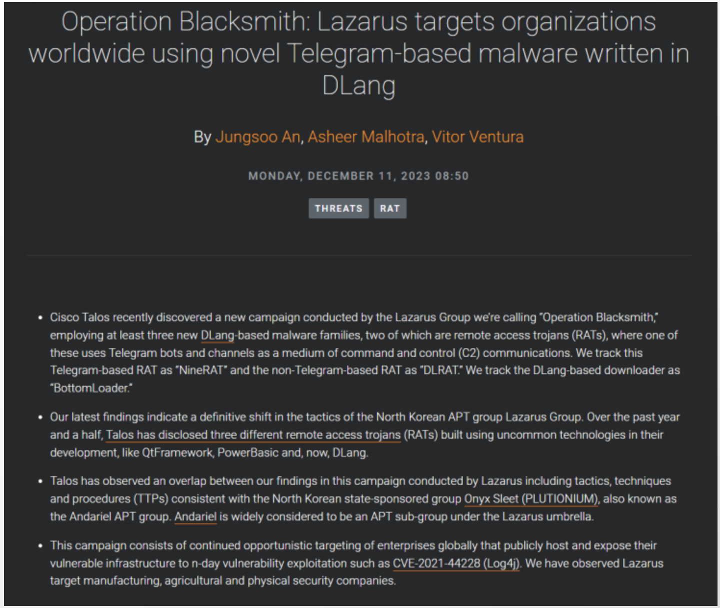
Assistant prompt

Illustration of the expected output



```
# Define the ASSISTANT prompt
assistant_prompt = (
    "mindmap\nroot(YoroTrooper Threat Analysis)\n    (Origin and Target)\n      ::icon(fa fa-crosshairs)\n      (Likely originates from Kazakhstan)\n      (Mainly targets
)
```

From article…



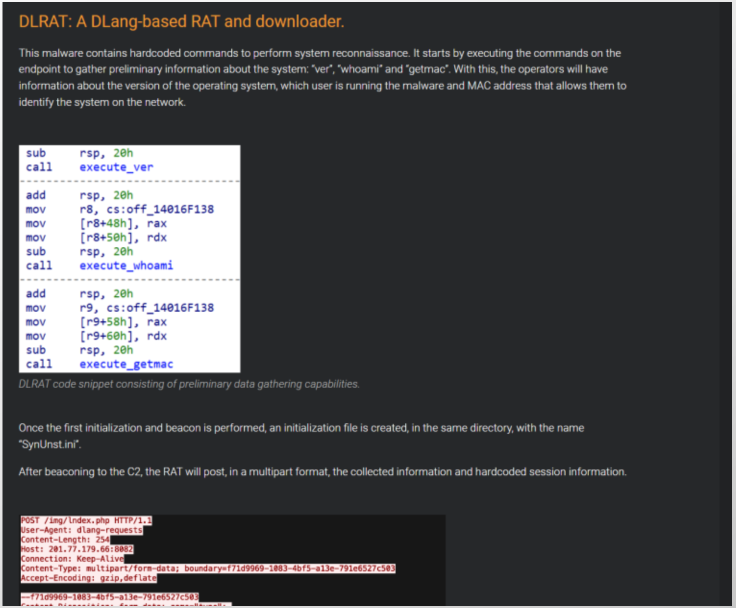
Operation Blacksmith: Lazarus targets organizations worldwide using novel Telegram-based malware written in DLang

By Jungsoo An, Asheer Malhotra, Vitor Ventura

MONDAY, DECEMBER 11, 2023 08:50

THREATS RAT

- Cisco Talos recently discovered a new campaign conducted by the Lazarus Group we're calling "Operation Blacksmith," employing at least three new DLang-based malware families, two of which are remote access trojans (RATs), where one of these uses Telegram bots and channels as a medium of command and control (C2) communications. We track this Telegram-based RAT as "NineRAT" and the non-Telegram-based RAT as "DLRAT." We track the DLang-based downloader as "BottomLoader."
- Our latest findings indicate a definitive shift in the tactics of the North Korean APT group Lazarus Group. Over the past year and a half, Talos has disclosed three different remote access trojans (RATs) built using uncommon technologies in their development, like QtFramework, PowerBasic and, now, DLang.
- Talos has observed an overlap between our findings in this campaign conducted by Lazarus including tactics, techniques and procedures (TTPs) consistent with the North Korean state-sponsored group Onyx Sleet (PLUTONIUM), also known as the Andariel APT group. Andariel is widely considered to be an APT sub-group under the Lazarus umbrella.
- This campaign consists of continued opportunistic targeting of enterprises globally that publicly host and expose their vulnerable infrastructure to n-day vulnerability exploitation such as CVE-2021-44228 (Log4j). We have observed Lazarus target manufacturing, agricultural and physical security companies.



DLRAT: A DLang-based RAT and downloader.

This malware contains hardcoded commands to perform system reconnaissance. It starts by executing the commands on the endpoint to gather preliminary information about the system: "ver", "whoami" and "getmac". With this, the operators will have information about the version of the operating system, which user is running the malware and MAC address that allows them to identify the system on the network.

```
sub     rsp, 20h
call    execute_ver

add     rsp, 20h
mov     r8, cs:off_14016F138
mov     [r8+48h], rax
mov     [r8+50h], rdx
sub     rsp, 20h
call    execute_whoami

add     rsp, 20h
mov     r9, cs:off_14016F138
mov     [r9+58h], rax
mov     [r9+60h], rdx
sub     rsp, 20h
call    execute_getmac
```

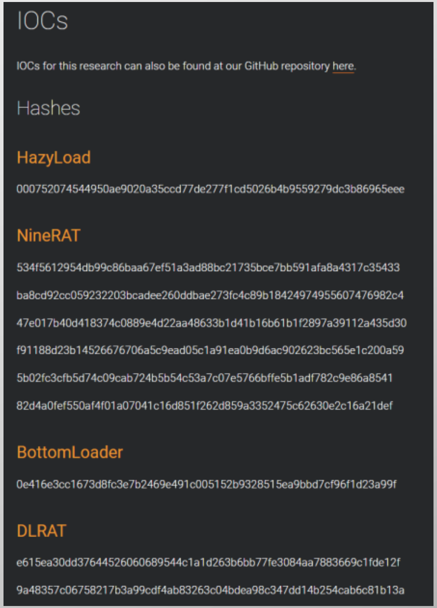
DLRAT code snippet consisting of preliminary data gathering capabilities.

Once the first initialization and beacon is performed, an initialization file is created, in the same directory, with the name "SynUnst.ini".

After beaconing to the C2, the RAT will post, in a multipart format, the collected information and hardcoded session information.

```
POST /img/lndex.php HTTP/1.1
User-Agent: dlang-requests
Content-Length: 254
Host: 201.77.179.66:8082
Connection: Keep-Alive
Content-Type: multipart/form-data; boundary=f71d9969-1083-4bf5-a13e-791e6527c503
Accept-Encoding: gzip,deflate

--f71d9969-1083-4bf5-a13e-791e6527c503
```



IOCs

IOCs for this research can also be found at our GitHub repository here.

Hashes

HazyLoad

000752074544950ae9020a35ccd77de277f1cd5026b4b9559279dc3b86965eee

NineRAT

534f5612954db99c86baa67ef51a3ad88bc21735bce7bb591afa8a4317c35433

ba8cd92cc059232203bcadee260ddbae273fc4c89b18424974955607476982c4

47e017b40d418374c0889e4d22aa48633b1d41b16b61b1f2897a39112a435d30

f91188d23b14526676706a5c9ead05c1a91ea0b9d6ac902623bc565e1c200a59

5b02fc3cfb5d74c09cab724b5b54c53a7c07e5766bffe5b1adf782c9e86a8541

82d4a0fef550af4f01a07041c16d851f262d859a3352475c62630e2c16a21def

BottomLoader

0e416e3cc1673d8fc3e7b2469e491c005152b9328515ea9bbd7cf96f1d23a99f

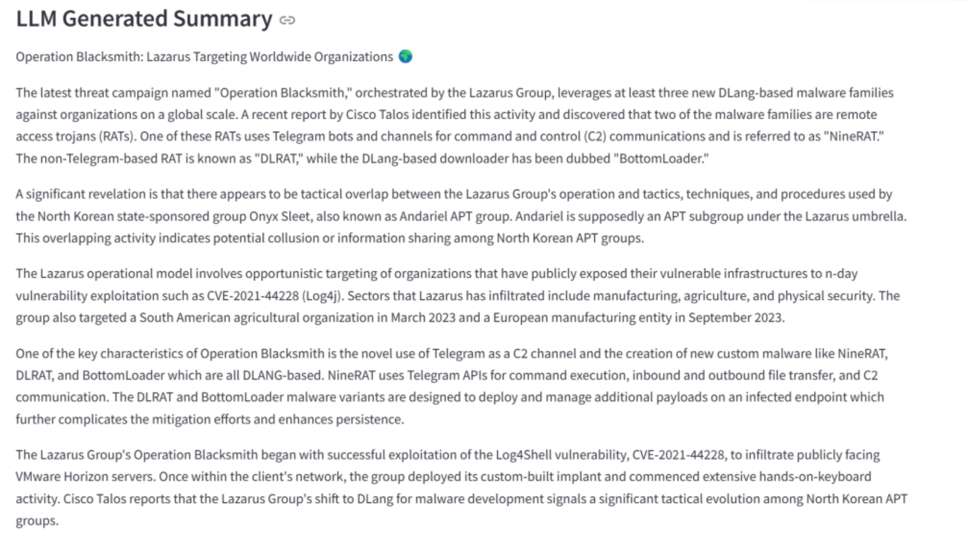
DLRAT

e615ea30dd37644526060689544c1a1d263b6bb77fe3084aa7883669c1fde12f

9a48357c06758217b3a99cdf4ab83263c04bdea98c347dd14b254cab6c81b13a

Operation Blacksmith: Lazarus targets organizations worldwide using novel Telegram-based malware written in Dlang, by Cisco Talos

… to TI Mindmap



LLM Generated Summary

Operation Blacksmith: Lazarus Targeting Worldwide Organizations 🌍

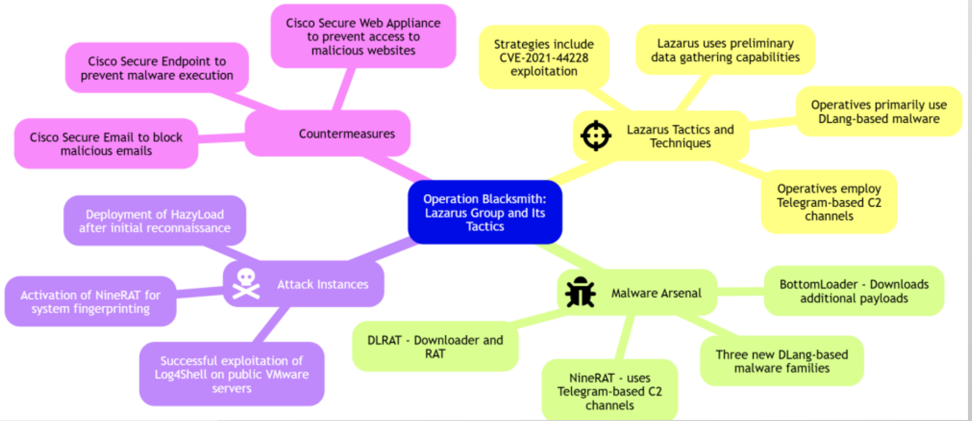
The latest threat campaign named "Operation Blacksmith," orchestrated by the Lazarus Group, leverages at least three new DLang-based malware families against organizations on a global scale. A recent report by Cisco Talos identified this activity and discovered that two of the malware families are remote access trojans (RATs). One of these RATs uses Telegram bots and channels for command and control (C2) communications and is referred to as "NineRAT." The non-Telegram-based RAT is known as "DLRAT," while the DLang-based downloader has been dubbed "BottomLoader."

A significant revelation is that there appears to be tactical overlap between the Lazarus Group's operation and tactics, techniques, and procedures used by the North Korean state-sponsored group Onyx Sleet, also known as Andariel APT group. Andariel is supposedly an APT subgroup under the Lazarus umbrella. This overlapping activity indicates potential collusion or information sharing among North Korean APT groups.

The Lazarus operational model involves opportunistic targeting of organizations that have publicly exposed their vulnerable infrastructures to n-day vulnerability exploitation such as CVE-2021-44228 (Log4j). Sectors that Lazarus has infiltrated include manufacturing, agriculture, and physical security. The group also targeted a South American agricultural organization in March 2023 and a European manufacturing entity in September 2023.

One of the key characteristics of Operation Blacksmith is the novel use of Telegram as a C2 channel and the creation of new custom malware like NineRAT, DLRAT, and BottomLoader which are all DLANG-based. NineRAT uses Telegram APIs for command execution, inbound and outbound file transfer, and C2 communication. The DLRAT and BottomLoader malware variants are designed to deploy and manage additional payloads on an infected endpoint which further complicates the mitigation efforts and enhances persistence.

The Lazarus Group's Operation Blacksmith began with successful exploitation of the Log4Shell vulnerability, CVE-2021-44228, to infiltrate publicly facing VMware Horizon servers. Once within the client's network, the group deployed its custom-built implant and commenced extensive hands-on-keyboard activity. Cisco Talos reports that the Lazarus Group's shift to DLang for malware development signals a significant tactical evolution among North Korean APT groups.

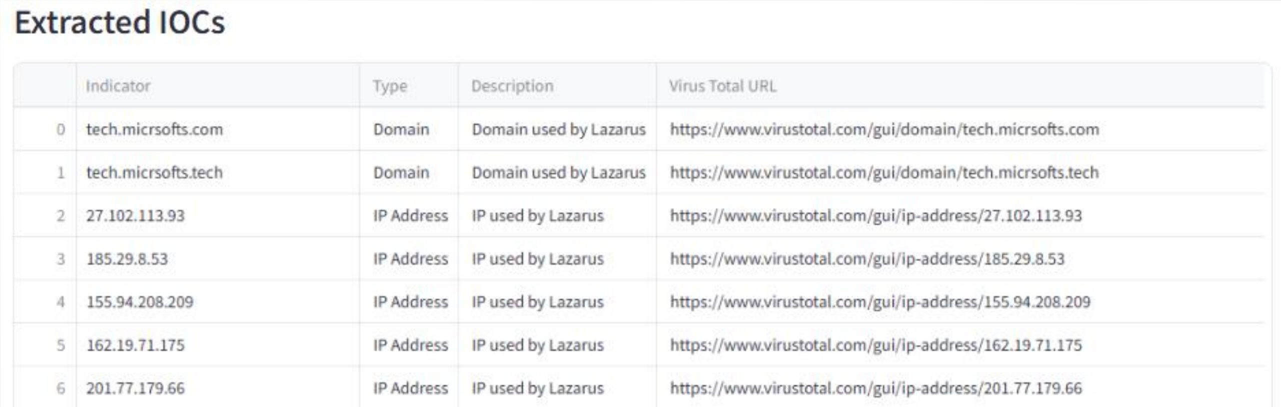

Operation Blacksmith: Lazarus Group and Its Tactics
Countermeasures
Cisco Secure Web Appliance to prevent access to malicious websites
Cisco Secure Endpoint to prevent malware execution
Cisco Secure Email to block malicious emails
Lazarus Tactics and Techniques
Strategies include CVE-2021-44228 exploitation
Lazarus uses preliminary data gathering capabilities
Operatives primarily use DLang-based malware
Operatives employ Telegram-based C2 channels
Attack Instances
Deployment of HazyLoad after initial reconnaissance
Activation of NineRAT for system fingerprinting
Successful exploitation of Log4Shell on public VMware servers
Malware Arsenal
BottomLoader - Downloads additional payloads
Three new DLang-based malware families
NineRAT - uses Telegram-based C2 channels
DLRAT - Downloader and RAT

LLM Generated Summary
Avg Latency: 34 sec
Prompt technique: *Task-Specific*
Input: *scraped text, language selection*

Mindmap
Avg Latency: 18 sec
Prompt technique: *few-shot learning*
Input: *scraped text, system, user, assistant prompts*

… to TI Mindmap

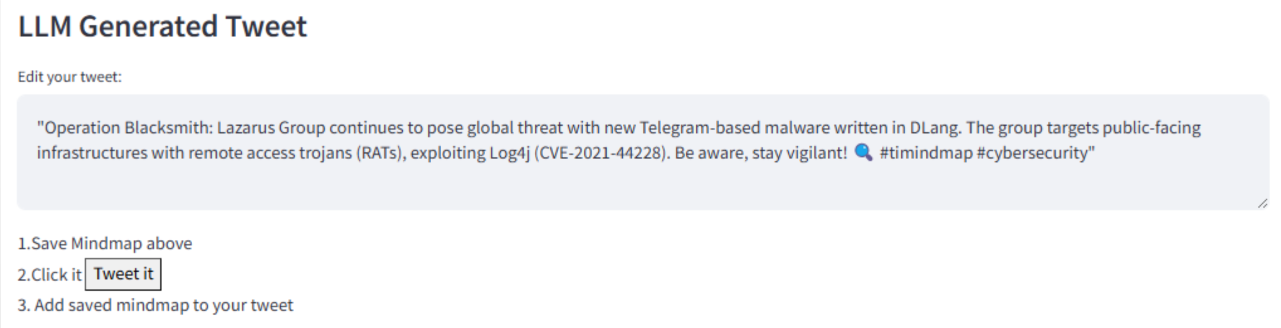
IOCs Extraction
AVG Latency: 44 sec
Prompt technique: *few-shot learning*
Input: *system, user prompts*



Extracted IOCs

	Indicator	Type	Description	Virus Total URL
0	tech.micrsofts.com	Domain	Domain used by Lazarus	https://www.virustotal.com/gui/domain/tech.micrsofts.com
1	tech.micrsofts.tech	Domain	Domain used by Lazarus	https://www.virustotal.com/gui/domain/tech.micrsofts.tech
2	27.102.113.93	IP Address	IP used by Lazarus	https://www.virustotal.com/gui/ip-address/27.102.113.93
3	185.29.8.53	IP Address	IP used by Lazarus	https://www.virustotal.com/gui/ip-address/185.29.8.53
4	155.94.208.209	IP Address	IP used by Lazarus	https://www.virustotal.com/gui/ip-address/155.94.208.209
5	162.19.71.175	IP Address	IP used by Lazarus	https://www.virustotal.com/gui/ip-address/162.19.71.175
6	201.77.179.66	IP Address	IP used by Lazarus	https://www.virustotal.com/gui/ip-address/201.77.179.66

Tweet generation
Avg Latency: 7 sec
Prompt technique: *Task-Specific*
Input: *scraped text, system, user prompts*



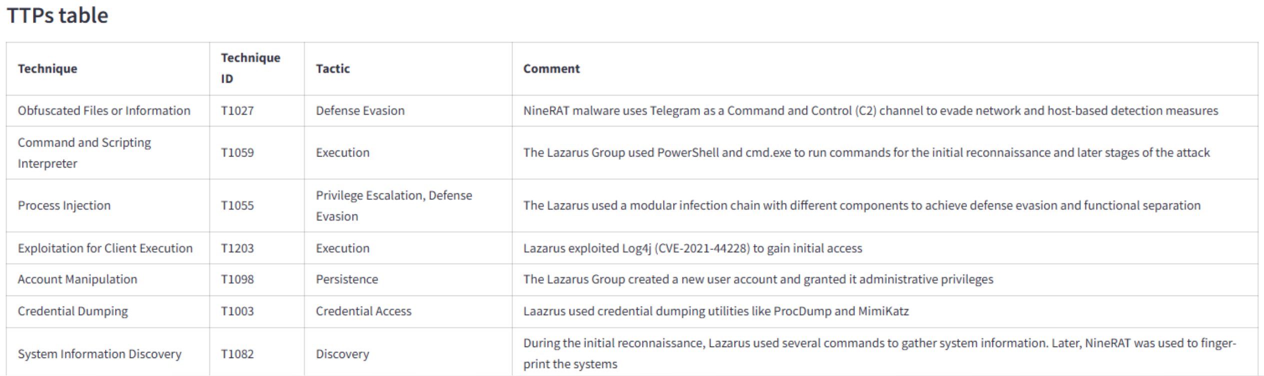
LLM Generated Tweet

Edit your tweet:

"Operation Blacksmith: Lazarus Group continues to pose global threat with new Telegram-based malware written in DLang. The group targets public-facing infrastructures with remote access trojans (RATs), exploiting Log4j (CVE-2021-44228). Be aware, stay vigilant! 🔍 #timindmap #cybersecurity"

1.Save Mindmap above
2.Click it Tweet it
3. Add saved mindmap to your tweet

… to TI Mindmap

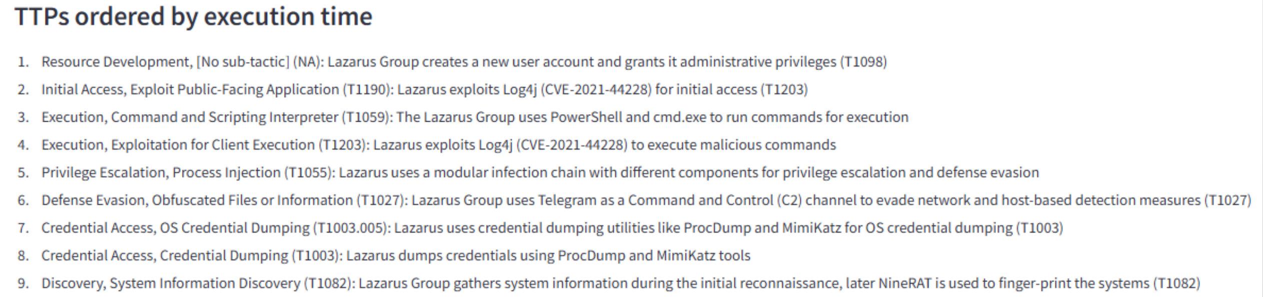


TTPs table

Technique	Technique ID	Tactic	Comment
Obfuscated Files or Information	T1027	Defense Evasion	NineRAT malware uses Telegram as a Command and Control (C2) channel to evade network and host-based detection measures
Command and Scripting Interpreter	T1059	Execution	The Lazarus Group used PowerShell and cmd.exe to run commands for the initial reconnaissance and later stages of the attack
Process Injection	T1055	Privilege Escalation, Defense Evasion	The Lazarus used a modular infection chain with different components to achieve defense evasion and functional separation
Exploitation for Client Execution	T1203	Execution	Lazarus exploited Log4j (CVE-2021-44228) to gain initial access
Account Manipulation	T1098	Persistence	The Lazarus Group created a new user account and granted it administrative privileges
Credential Dumping	T1003	Credential Access	Laazrus used credential dumping utilities like ProcDump and MimiKatz
System Information Discovery	T1082	Discovery	During the initial reconnaissance, Lazarus used several commands to gather system information. Later, NineRAT was used to finger-print the systems

TTPs Table

Avg Latency: 23 sec
Prompt technique: *Task-Specific*
Input: *scraped text, user prompt*

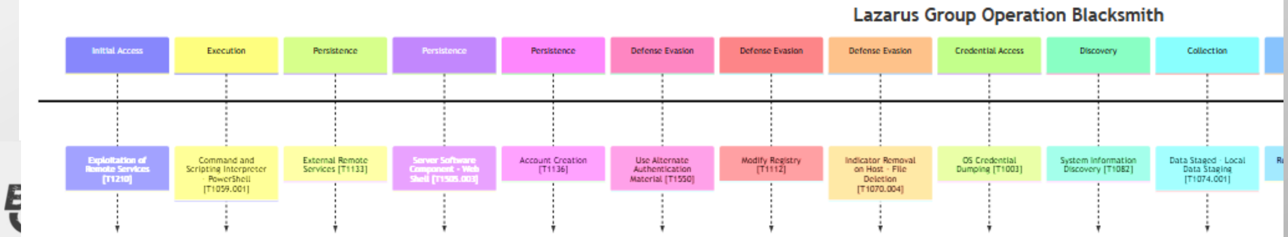


TTPs ordered by execution time

1. Resource Development, [No sub-tactic] (NA): Lazarus Group creates a new user account and grants it administrative privileges (T1098)
2. Initial Access, Exploit Public-Facing Application (T1190): Lazarus exploits Log4j (CVE-2021-44228) for initial access (T1203)
3. Execution, Command and Scripting Interpreter (T1059): The Lazarus Group uses PowerShell and cmd.exe to run commands for execution
4. Execution, Exploitation for Client Execution (T1203): Lazarus exploits Log4j (CVE-2021-44228) to execute malicious commands
5. Privilege Escalation, Process Injection (T1055): Lazarus uses a modular infection chain with different components for privilege escalation and defense evasion
6. Defense Evasion, Obfuscated Files or Information (T1027): Lazarus Group uses Telegram as a Command and Control (C2) channel to evade network and host-based detection measures (T1027)
7. Credential Access, OS Credential Dumping (T1003.005): Lazarus uses credential dumping utilities like ProcDump and MimiKatz for OS credential dumping (T1003)
8. Credential Access, Credential Dumping (T1003): Lazarus dumps credentials using ProcDump and MimiKatz tools
9. Discovery, System Information Discovery (T1082): Lazarus Group gathers system information during the initial reconnaissance, later NineRAT is used to finger-print the systems (T1082)

TTPs by exec time

Avg Latency: 14 sec
Prompt technique: *Task-Specific*
Input: *scraped text,system, user prompts*


Lazarus Group Operation Blacksmith
Initial Access
Exploitation of Remote Services [T1210]
Execution
Command and Scripting Interpreter · PowerShell [T1059.001]
Persistence
External Remote Services [T1133]
Persistence
Server Software Component · Web Shell [T1505.003]
Persistence
Account Creation [T1136]
Defense Evasion
Use Alternate Authentication Material [T1550]
Defense Evasion
Modify Registry [T1112]
Defense Evasion
Indicator Removal on Host · File Deletion [T1070.004]
Credential Access
OS Credential Dumping [T1003]
Discovery
System Information Discovery [T1082]
Collection
Data Staged · Local Data Staging [T1074.001]

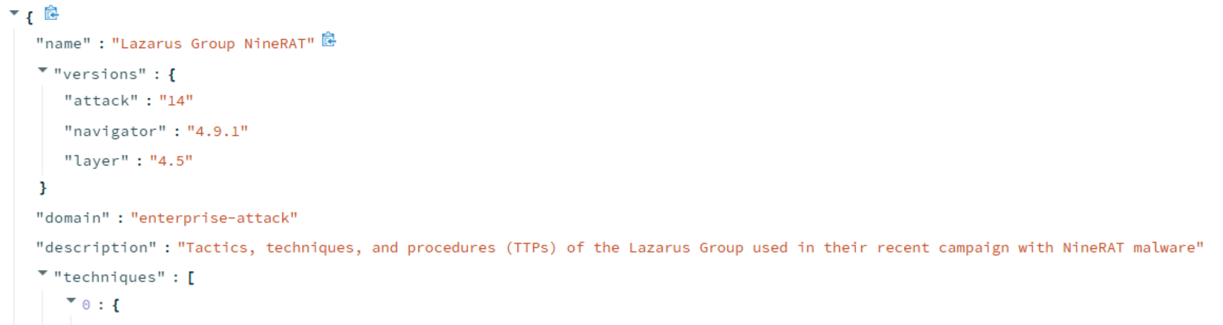
TTP timeline

Avg Latency: 17 sec
few-shot learning
Input: *scraped text, system, user, assistant prompts*

… to TI Mindmap

Mitre layer file
Avg Latency: 45 sec
Prompt technique:
few-shot learning
Input: *scraped text, system, user assistant prompts*

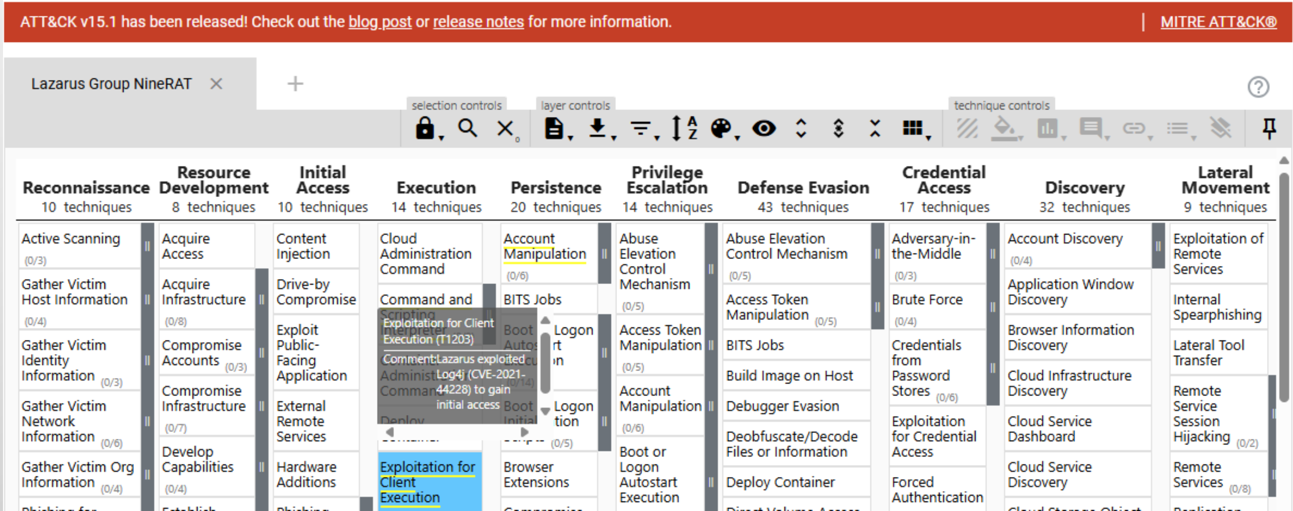
MITRE ATT&CK Navigator layer json



```
{
  "name" : "Lazarus Group NineRAT"
  "versions" : {
    "attack" : "14"
    "navigator" : "4.9.1"
    "layer" : "4.5"
  }
  "domain" : "enterprise-attack"
  "description" : "Tactics, techniques, and procedures (TTPs) of the Lazarus Group used in their recent campaign with NineRAT malware"
  "techniques" : [
    0 : {
```

Mitre Navigator embedded as iFrame into Streamlit app.

Mitre Navigator

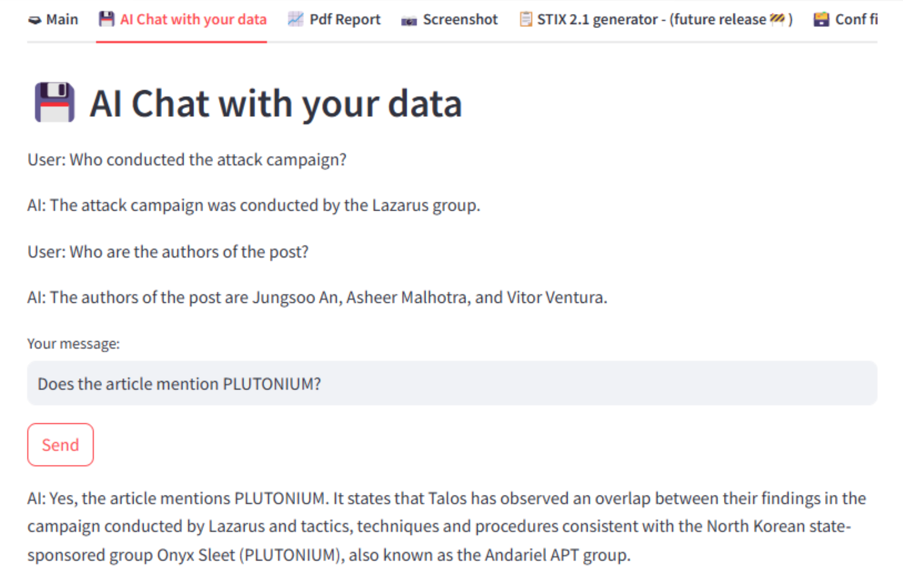


ATT&CK v15.1 has been released! Check out the blog post or release notes for more information. | MITRE ATT&CK®

Reconnaissance 10 techniques	Resource Development 8 techniques	Initial Access 10 techniques	Execution 14 techniques	Persistence 20 techniques	Privilege Escalation 14 techniques	Defense Evasion 43 techniques	Credential Access 17 techniques	Discovery 32 techniques	Lateral Movement 9 techniques
Active Scanning (0/3)	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation (0/6)	Abuse Elevation Control Mechanism (0/5)	Abuse Elevation Control Mechanism (0/5)	Adversary-in-the-Middle (0/3)	Account Discovery (0/4)	Exploitation of Remote Services
Gather Victim Host Information (0/4)	Acquire Infrastructure (0/8)	Drive-by Compromise	Command and Scripting Interpreter	BITS Jobs	Access Token Manipulation (0/5)	Access Token Manipulation (0/5)	Brute Force (0/4)	Application Window Discovery	Internal Spearphishing
Gather Victim Identity Information (0/3)	Compromise Accounts (0/3)	Exploit Public-Facing Application		Boot or Logon Autostart Execution	Account Manipulation (0/6)	BITS Jobs	Credentials from Password Stores (0/6)	Browser Information Discovery	Lateral Tool Transfer
Gather Victim Network Information (0/6)	Compromise Infrastructure (0/7)	External Remote Services		Boot or Logon Initialization Scripts (0/5)	Boot or Logon Autostart Execution	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (0/2)
Gather Victim Org Information (0/4)	Develop Capabilities (0/4)	Hardware Additions	Exploitation for Client Execution	Browser Extensions		Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services (0/8)
						Deobfuscate/Decode Files or Information		Cloud Service Discovery	
						Deploy Container			

Exploitation for Client Execution (T1203)
Comment: Lazarus exploited Log4j (CVE-2021-44228) to gain initial access

AI Chat



AI Chat with your data

User: Who conducted the attack campaign?

AI: The attack campaign was conducted by the Lazarus group.

User: Who are the authors of the post?

AI: The authors of the post are Jungsoo An, Asheer Malhotra, and Vitor Ventura.

Your message:

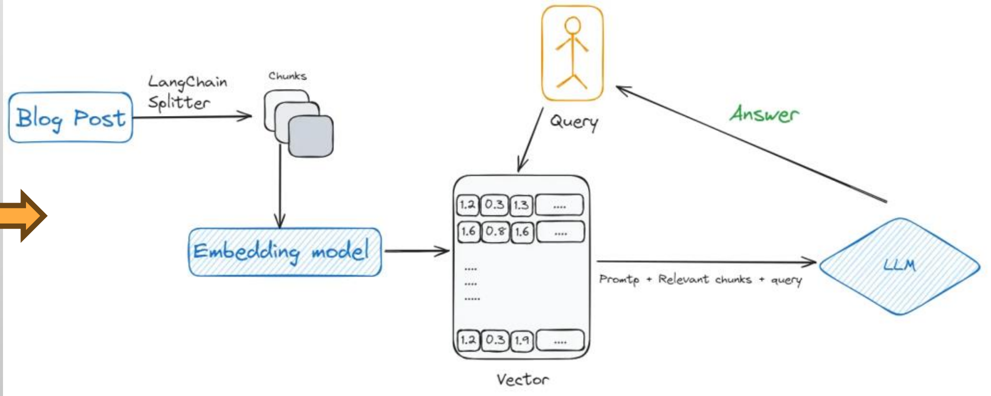
Does the article mention PLUTONIUM?

AI: Yes, the article mentions PLUTONIUM. It states that Talos has observed an overlap between their findings in the campaign conducted by Lazarus and tactics, techniques and procedures consistent with the North Korean state-sponsored group Onyx Sleet (PLUTONIUM), also known as the Andariel APT group.

AI Chat

> RAG (Retrieval Augmented Generation)
> Chunking
> Embedding
> Embedding vector processed in memory
> FAISS for similarity search
> LangChain

TI Mindmap chatbot architecture



Challenges in developing TI Mindmap

> **LLM response times**

> **Variable and probabilistic output**

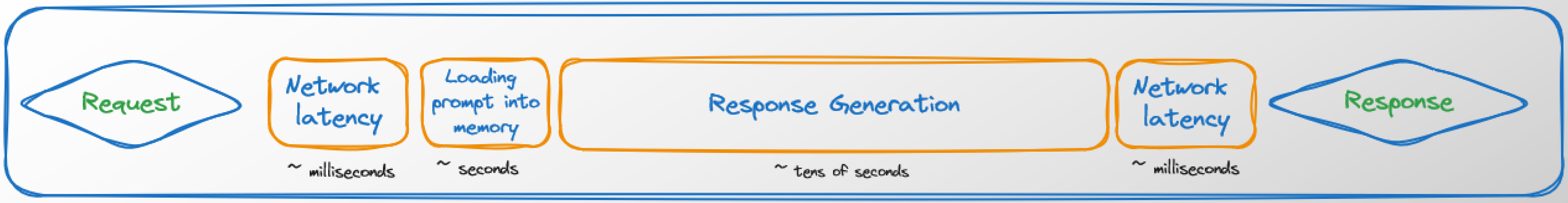
> **Cost due to token usage**

> LLMs Don’t Have Memory

> LLMs Might Hallucinate

> Willingness to minimize the need for a backend during the prototyping phase

Latency -> per-call response times

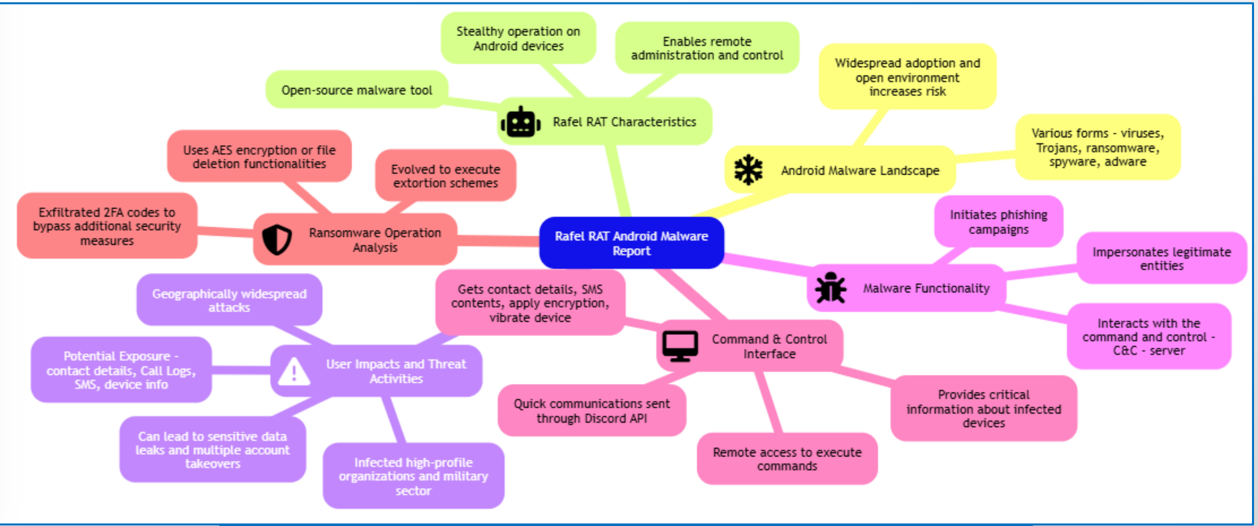


The latency depends on:
> the model used
> the number of tokens in the prompt.
> the number of tokens generated.
> the overall load on the deployment
        & system

Techniques for improving latency:
> Select the appropriate model size
> Don't use LLMs for extensive
        predefined text output.
> Prompt optimization
> Batching or parallelizing API calls
> Limit the use of LLMs to only when
        strictly necessary (traditional
        methods are always valid)☺

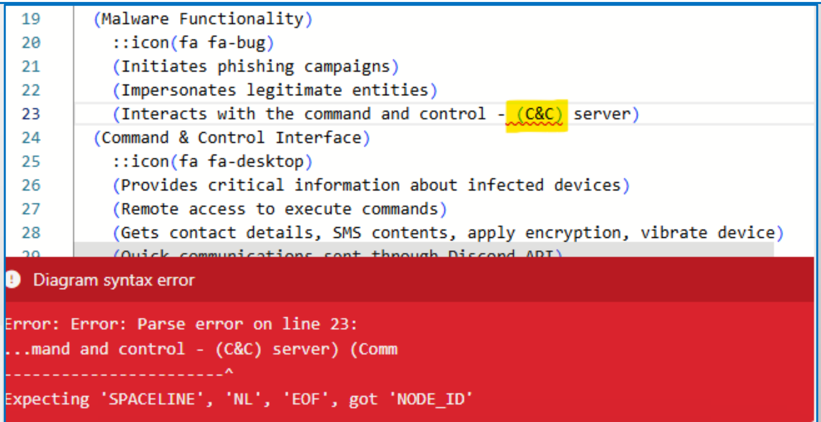
https://techcommunity.microsoft.com/t5/ai-azure-ai-services-blog/the-llm-latency-guidebook-optimizing-response-times-for-genai/ba-p/4131994

Variable and probabilistic output - example


Rafel RAT Android Malware Report
Rafel RAT Characteristics
Open-source malware tool
Stealthy operation on Android devices
Enables remote administration and control
Android Malware Landscape
Widespread adoption and open environment increases risk
Various forms - viruses, Trojans, ransomware, spyware, adware
Ransomware Operation Analysis
Uses AES encryption or file deletion functionalities
Evolved to execute extortion schemes
Exfiltrated 2FA codes to bypass additional security measures
Malware Functionality
Initiates phishing campaigns
Impersonates legitimate entities
Interacts with the command and control - C&C - server
Command & Control Interface
Gets contact details, SMS contents, apply encryption, vibrate device
Quick communications sent through Discord API
Remote access to execute commands
Provides critical information about infected devices
User Impacts and Threat Activities
Geographically widespread attacks
Potential Exposure - contact details, Call Logs, SMS, device info
Can lead to sensitive data leaks and multiple account takeovers
Infected high-profile organizations and military sector

1st run

OK ☺



```
19  (Malware Functionality)
20    ::icon(fa fa-bug)
21    (Initiates phishing campaigns)
22    (Impersonates legitimate entities)
23    (Interacts with the command and control - (C&C) server)
24  (Command & Control Interface)
25    ::icon(fa fa-desktop)
26    (Provides critical information about infected devices)
27    (Remote access to execute commands)
28    (Gets contact details, SMS contents, apply encryption, vibrate device)
29    (Quick communications sent through Discord API)
```

Diagram syntax error

```
Error: Error: Parse error on line 23:
...mand and control - (C&C) server) (Comm
-----------------------^
Expecting 'SPACELINE', 'NL', 'EOF', got 'NODE_ID'
```

Nth run

Not OK ☹ due to mermaid.js syntax error

Mitigating Non-Determinism in LLMs

While it's impossible to eliminate the risks of non-determinism entirely, there are some strategies that can help you achieve more consistent results

> Craft Specific Prompts
> Use templates
> temperature: this controls the randomness of the model's response
> Agents to augment LLMs
> …

Cost due to token usage

> Commercial LLM are ready to use, do not require your infrastructure, but have significant costs.
> Currently, the main cost is associated with input tokens rather than output tokens.
> The average cost to process a writeup is about 1$.

Possible optimizations

> Prompt engineering
> Memory, caching
> Selecting the right LLM model
> Local Small Language Models

Roadmap & new ideas

> 🚀 App **Agentification**
> 📝 **STIX** 2.1 threat reports
> 📓 Jupyter **notebooks** generator
> 💎 **Diamond** model
> 📝 **5W1H** report (WHAT, WHEN, WHERE, WHO, WHY, and HOW)
> 🔌 **API** access
> 📄 Extending **input** types (pdf, docx, etc…)
> 🌐 Open-source **SML** Small Language Models

How to get involved

- The project is open to external contributions.
- GitHub: https://github.com/format81/TI-Mindmap-GPT/
- Streamlit App: https://ti-mindmap-gpt.streamlit.app/

TI Mindmap

GitHub Repo