GPTHound –
Your Active Directory Security Assistant

John Jiang
Senior Cyber Security Researcher, CyCraft
Whoami

- Senior Cyber Security Researcher @ CyCraft
- Co-founder @ UCCU Hacker
- Speaker at
  - Black Hat USA
  - HITCON
  - CODEBLUE
  - HITB ...
- Focus on Active Directory, Windows Security
Outline

› Intro
  › Current LLM in cybersecurity
  › Active Directory Situation & Current Analysis Method

› LLM with Active Directory Security
  › What Analysts Actually Analyze
  › Determining the Capabilities of the LLM

› Design & Demo:
  › How Do We Design It? Workflow & Tools We Select
  › GPT/LLM Integration Challenge
  › Demo
LLM/AI is Hot Topic this year
How about LLM in cyber security?
LLM in Cyber Security

➢ Red Team Related:
  ➢ Phishing - WormGPT/FraudGPT
  ➢ Vulnerability finder – BurpGPT
  ➢ Attack Command Executor- PentestGPT

➢ Blue Team Related:
  ➢ Alert/Report Interpreter – Some Security Vendor
  ➢ Malware Analysis – Gepetto
  ➢ InterpReTable Incident Inspector – CyCraft IRonMan
I want use LLM, too!
We choose Active Directory as our Target.
Why Active Directory Security?

- Misconfiguration can take over the entire domain
- Compromise AD = Compromise the enterprise
- New attack methods are still being found
Current Blue Team for ad security practices (Cont.)

• Attack Path Visualization:
  • Adalanche (Open Source)
  • BloodHound/BlueHound (Open Source)
  • Forest Druid (Free Use)
  • BloodHound Ent (Commercial)

• Prioritizing the issues & Security Configuration:
  • BlueHound (Open Source)
  • Tenable.ad (Commercial)
  • PingCastle (Free Use + Commercial)
Attack Path Visualization
Analysts need to do...

- Find the attack path to critical object
- Figure out the attack path shouldn’t exist
- Find vulnerable setting
  - Kerberoasting
  - Plaintext in description
  - Certificate Template misconfiguration
  - ...
- Summarize it, write a report
The tasks undertaken by analysts during the analysis process
Attack Path Analysis Tasks

- Find the attack path to critical object
- Figure out the attack path shouldn’t exist
- Find vulnerable setting
  - Kerberoasting
  - Certificate Template misconfiguration
  - Plaintext in description
  - ...
- Summarize it, write a report

- Generate Query to Search Attack Path
- Cluster Users/Hosts
- Search Sensitive Information
- Explain Attack Path and Remediation
We operate as 4 major task types

- **Generate Query to Search Attack Path**
  - Simplify tool usage and report generation for analysts by leveraging LLM to convert natural language into neo4j queries with tools such as Bloodhound

- **Cluster Users/Hosts**
  - Asset classification to group objects or potential attack groups based on job roles and groups, thus identifying assets cluster.

- **Search Sensitive Information**
  - Find information on AD, such as passwords, with the added capability of detecting sensitive information in multiple languages.

- **Explain Attack Path and Remediation**
  - Explain attack paths between different object types and provide appropriate recommendations.
We separate as 4 major task types

- **Generate Query to Search Attack Path**
  - Simplify tool usage and report generation for analysts by leveraging LLM to convert natural language into neo4j queries with tools such as Bloodhound

- **Cluster Users/Hosts**
  - Asset classification to group objects or potential attack groups based on job roles and groups, thus identifying assets cluster.

- **Search Sensitive Information**
  - Find information on AD, such as passwords, with the added capability of detecting sensitive information in multiple languages.

- **Explain Attack Path and Remediation**
  - Explain attack paths between different object types and provide appropriate recommendations.
We operate as 4 major task types

- **Generate Query to Search Attack Path**
  - Simplify tool usage and report generation for analysts by leveraging LLM to convert natural language into neo4j queries with tools such as Bloodhound.

- **Cluster Users/Hosts**
  - Asset classification to group objects or potential attack groups based on job roles and groups, thus identifying assets cluster.

- **Search Sensitive Information**
  - Find information on AD, such as passwords, with the added capability of detecting sensitive information in multiple languages.

- **Explain Attack Path and Remediation**
  - Explain attack paths between different object types and provide appropriate recommendations.
We separate as 4 major task types

- **Generate Query to Search Attack Path**
  - Simplify tool usage and report generation for analysts by leveraging LLM to convert natural language into neo4j queries with tools such as Bloodhound.

- **Cluster Users/Hosts**
  - Asset classification to group objects or potential attack groups based on job roles and groups, thus identifying assets cluster.

- **Search Sensitive Information**
  - Find information on AD, such as passwords, with the added capability of detecting sensitive information in multiple languages.

- **Explain Attack Path and Remediation**
  - Explain attack paths between different object types and provide appropriate recommendations.
System Design
LangChain

- Modular framework centered around chaining language model API calls
- Integrations with popular LLMs
- Tools for prompts, memory, indexing, agents, and more
- Support for common NLP application patterns
- Enables complex workflows by combining language models, data sources, and custom logic
System Design

Core Brain
LangChain w/ GPT3.5

Legend:
Component
Tool

Generate Query to Search Attack Path
Search Sensitive Information
Cluster Users/Hosts
Explain Attack Path and Remediation

Neo4j Connector

BloodHound DB
Cluster Result CSV
Attack Explain & Remediation csv

CSV operator
Generate Query to Search Attack Path
Q: Can you write Cypher query, find out the user might be under Kerberoating attack?
Q: Can you write Cypher query, find out the user might be under Kerberoasting attack?

You need to know following things:

- The bloodhound schema
- Cypher query language
- Kerberoasting is an attack technique that targets Kerberos service tickets to extract and crack the associated service account's plaintext passwords. Kerberoasting need user/object has serviceprincipalname
Generate Query to Search Attack Path

- Goal: Help analyst generate Neo4j query for bloodhound
- Example: generate cypher query, I want all user object with hasspn.

```
MATCH (n:User)
WHERE n.servicePrincipalName IS NOT NULL AND
  NOT n:Computer AND
  NOT n:Service AND
  NOT n:ForeignSecurityPrincipal AND
  NOT n:Trust
RETURN n.displayName AS UserName, n.servicePrincipalName AS ServicePrincipal
```

Sure, here's an example of a Cypher query you can use in BloodHound to identify potential Kerberoasting attack targets:
Generate Query to Search Attack Path

Challenge: ChatGPT not familiar with BloodHound

MATCH (n:User)
WHERE n.servicePrincipalName IS NOT NULL AND
    NOT n:Computer AND
    NOT n:Service AND
    NOT n:ForeignSecurityPrincipal AND
    NOT n:Trust
RETURN n.displayName AS UserName, n.servicePrincipalName AS ServicePrincipal
Generate Query to Search Attack Path

Tasks flowcharts

- Neo4j Schema
- User Input Question
- ChatGPT Prompts Template
- Extract Cypher Query
Generate Query to Search Attack Path

CYPHER_GENERATION_TEMPLATE

› Please consider the possibilities of uppercase, lowercase, and Null values.

› Use MATCH and WHERE as much as possible to filter the data.

› Use only the provided relationship types and properties in the schema.
Generate Query to Search Attack Path
GPTHound - W/ any reference or input data

```
Insert your text. Enter 'q' or press Ctrl-D (or Ctrl-Z on Windows) to end.
generate cypher query, I want all user object might be kerbercoating attack.
q

> Entering new chain...
Action:
```
{
  "action": "cypher query generator",
  "action_input": "I want all user object might be kerbercoating attack."
}
```

Observation:

```cypher
MATCH (u:User) WHERE u.hasspn = true AND u.admincount = true RETURN u;
```
Generate Query to Search Attack Path

GPTHound - W/ any reference or input data

MATCH (u:User) WHERE u.hasspn = true AND u.admincount = true RETURN u;
Cluster Users/Hosts
Cluster similar Users/Hosts

➤ Goal: Given user/computer object, find related object
➤ Example: Find the most similar account with User “John”
Cluster similar Users/Hosts

- Goal: Given user/computer object, find related object
- Example: Find the most similar account with User “John”
- Challenge:
  - What is the definition of "similar" in this context?
Q: How do you identify objects are related?

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>DN</th>
<th>GroupList</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD-WEB-93</td>
<td>2023B53375 lurleen.florinda NB</td>
<td>['CN=Computers']</td>
<td>[515]</td>
</tr>
<tr>
<td>NY-MGMT-145</td>
<td>2023B67544 lurleen.florinda PC</td>
<td>['CN=Computers']</td>
<td>[515]</td>
</tr>
<tr>
<td>TK-DB-821</td>
<td>2023B76452 lurleen.florinda PC</td>
<td>['CN=Computers']</td>
<td>[515]</td>
</tr>
<tr>
<td>john</td>
<td>123</td>
<td>['CN=Users']</td>
<td>[512]</td>
</tr>
<tr>
<td>stevenwang</td>
<td></td>
<td>['CN=Users']</td>
<td>[512]</td>
</tr>
</tbody>
</table>
Q: How do you identify objects are related?

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>DN</th>
<th>GroupList</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD-WEB-93</td>
<td>2023B53375 lurleen.florinda NB</td>
<td>['CN=Computers']</td>
<td>[515]</td>
</tr>
<tr>
<td>NY-MGMT-145</td>
<td>2023B67544 lurleen.florinda PC</td>
<td>['CN=Computers']</td>
<td>[515]</td>
</tr>
<tr>
<td>TK-DB-821</td>
<td>2023B76452 lurleen.florinda PC</td>
<td>['CN=Computers']</td>
<td>[515]</td>
</tr>
<tr>
<td>john</td>
<td>123</td>
<td>['CN=Users']</td>
<td>[512]</td>
</tr>
<tr>
<td>stevenwang</td>
<td></td>
<td>['CN=Users']</td>
<td>[512]</td>
</tr>
</tbody>
</table>
### Summary the classification rules

<table>
<thead>
<tr>
<th>Permission</th>
<th>Organizational Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; Based on the same permissions, we currently use groups as a reference to determine similarity.</td>
<td>&gt; &quot;Organizational Unit&quot; is usually delineated based on departments or levels of authority, and then categorized by function, such as Servers or Service Accounts.</td>
<td>&gt; Apart from passwords, certain Active Directory (AD) objects may also include annotations such as a Ticket Number or an asset code.</td>
</tr>
<tr>
<td>&gt; The vast majority of group members are added manually, making it more aligned with our intention to identify individuals who are considered similar.</td>
<td></td>
<td>&gt; These annotations typically allow us to identify the owner or administrator associated with them.</td>
</tr>
</tbody>
</table>
Clustering these data

Ref: https://platform.openai.com/docs/guides/embeddings
TLDR;
GPT Embedding is unnecessary for this scenario.
We employ our own methodology for clustering
Demo: Clustering Result
Comparing with the results of real analysis...

- Analyzing over 20+ domains and comparing with analyst results.
- Comparing with human analysts, the accuracy of judgment is approximately 70.5%.

For the remaining 29.5%, the reasons for inconsistent classification results can be categorized as follows:

- The domain size is too narrow (less than 1000 user objects), lacking a management structure.
- When using the AD three-tier management structure, objects in Tier 0 and Tier 1 mostly lack group relationships and only have the same DN, resulting in unsatisfactory clustering results.
Comparing with the results of real analysis...

- Analyzing over 20+ domains and comparing with analyst results.
- Comparing with human analysts, the accuracy of judgment is approximately 70.5%.

- For the remaining 29.5%, the reasons for inconsistent classification results can be categorized as follows:
  - The domain size is too narrow (less than 1000 user objects), lacking a management structure.
  - When using the AD three-tier management structure, objects in Tier 0 and Tier 1 mostly lack group relationships and only have the same DN, resulting in unsatisfactory clustering results.
Search Sensitive Information
Which of these might be passwords?

- Station: DB   Password: vfddfview1
- JohnSmith-SERVER2003
- ji32k7hk4g4au48a3
- AIM compile server-l200
- VMware View Composer created account.
- PW: B@lDincaan
- P1aa@Ex01
- System Account Custodian: ben / John
  Purpose: Automation
- Barcode Password: 12343333
- EAn@Backup!! For Exchange Backup
- EIP_MN for C2m$345s#49#231$Ffw
Which of these might be passwords?

- Station: DB  Password: vfddfview1
- JohnSmith-SERVER2003
- ji32k7hk4g4au48a3
- AIM compile server-I200
- VMware View Composer created account.
- PW: B@lDincaan
- P1aa@Ex01
- System Account Custodian: ben / John
  Purpose: Automation
- Barcode Password: 12343333
- EAn@Backup!! For Exchange Backup
- EIP_MN for C2m$345s#49#231$Fffw
LLM can identify potential passwords, significantly reducing manual efforts.
Search Sensitive Information

▶ Goal: find sensitive information, such as password, project name...

▶ Example:
  
  read file `ad_object_descriptions` & check which string may be password
Search Sensitive Information

➢ Goal: find sensitive information, such as password, project name...

➢ Example:
  - read file `ad_object_descriptions` & check which string may be password

➢ Challenge:
  - Length limitation
How to overcome the length limitation?

- Vector
  - Split to Chunks -> Text to Embedding -> Vector store
How to overcome the length limitation?

- Vector
- Split to Chunks
  - Text to Embedding
  - Vector store

- Split the data
  - Split to Chunks
  - Send to LLM
  - Store the result
  - Merge
How to overcome the length limitation?

- Vector
  - Split to Chunks → Text to Embedding → Vector store
- Split the data
  - Split to Chunks → Send to LLM → Store the result → LLM Merge Result
Explain Attack Path and Remediation
Q: Could you explain this attack path? 
[John]{User}->[GenericAll]->[Key Admins]{Group}
Explain Attack Path and Remediation

Goal: Explain specific attack path and get appropriate remediation

Example:

- explain this path in english, keep the tool response format [John]{User}–[GenericAll]–>[Key Admins]{Group}

Challenge:

- ChatGPT don’t know too much active directory security
**Goal:** Explain specific attack path and get appropriate remediation.

**Example:** explain this path in English, keep the tool response format: `$[John]{User}$-[GenericAll]->$[Key Admins]{Group}$`

**Challenge:** ChatGPT doesn't know too much about Active Directory security. No clear path representation method, displaying multiple nodes in a single Cypher path.

**Response:**

Certainly! Let's break down the path you provided:

- `$[John]{User}$`: This represents an individual named "John" who is categorized as a "User".
- `$[GenericAll]->$`: This is a connection or relationship labeled "GenericAll" that originates from "John".
- `$[Key Admins]{Group}$`: This represents a group called "Key Admins" that is categorized as a "Group".

In plain English, the path can be described as:

"John, who is a User, has a 'GenericAll' relationship or connection to the 'Key Admins' group."

Would you like this explanation to be visualized in a flowchart or mind map format?
Explain Attack Path and Remediation Requirements

- **Anonymization**: Implement de-identification measures.
- **Utilize Approved Reference Data**: Minimize instances of misinformation by relying on specified reference materials.
- **Comprehensive Explanation**: Ensure that explanation includes the impact and strategies for mitigation.
Explain Attack Path and Remediation

Comprehensive Explanation

**Description & Recommendation**

<table>
<thead>
<tr>
<th>Edge_en</th>
<th>Description_tw</th>
<th>Recommendation_tw</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenericAll</td>
<td>對於指定物件完全控制，能夠任意修改物件的 DACL (Discretionary Access Control List)、寫入物件屬性。</td>
<td>確認是否需要對目標寫入屬性、修改控制權等權限，若無必要移除其權限</td>
</tr>
</tbody>
</table>

**Impact**

<table>
<thead>
<tr>
<th>Edge_en</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenericAll</td>
<td>[Source_Node] 可控制 [Destination_Node] 修改群組成員。</td>
</tr>
</tbody>
</table>
**Explain Attack Path and Remediation**

**Comprehensive Explanation**

### Description & Recommendation

<table>
<thead>
<tr>
<th>Edge_en</th>
<th>Description</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenericAll</td>
<td>With full authority over the designated object, the capacity to freely alter the object's DACL (Discretionary Access Control List), and the capability to edit object properties at will.</td>
<td>Please verify if it's necessary to assign attributes, modify access rights, or permissions to the target. If not required, then remove the permissions.</td>
</tr>
</tbody>
</table>

### Impact

<table>
<thead>
<tr>
<th>Edge_en</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenericAll</td>
<td>[Source_Node] has the ability to control [Destination_Node] and make modifications to group members.</td>
</tr>
</tbody>
</table>
Explain Attack Path and Remediation

GPTHound Response

**Observation:** AD bot (gpt-3.5-turbo-0301)

**Path:**

[John]{User}→[GenericAll]→[Key Admins]{Group}

**Model Response:**

This path indicates that the user John has GenericAll permissions over the Key Admins group, allowing him to control the members of the Key Admins group. It is recommended to verify if John actually needs this permission, and if not, it should be removed.
1. The LLM facilitates the generation of a cypher query.
2. Execute the cypher query and store the resulting data.
3. Retrieve the stored data to identify sensitive objects that contain passwords.
4. Identify the targeted account of interest and generate an attack path.
5. Explain the attack path and propose mitigation.
Conclusion
Key Takeaway

- There are many security issues in AD that require a lot of information to be memorized. By using LLM as an assistant, it accelerates the efficiency of analysts.

- LLM is particularly suitable for handling tasks that don't require high precision but consume a lot of analyst's energy.
  - For example, writing articles, determining similar texts, or passwords...

- Prepare reference materials for LLM to understand the desired output.
  - Use non-English materials as references; the final output in English can still be effective.

- Avoid spending excessive time adjusting prompts. Instead, focus on processing input data or enabling LLM to directly use tools like Python/csv to extract the necessary information.
Thanks!