

# Prompting LLM to Enforce and Validate CIS Critical Security Control

**Mohiuddin Ahmed**

Software and Information Systems  
University of North Carolina at Charlotte  
Charlotte, NC, USA



**Jinpeng Wei**

Associate Professor  
Software and Information System  
University of North Carolina at Charlotte  
Charlotte, NC, USA



**Ehab Al-Shaer**

Distinguished Professor  
School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA, USA

The logo of Carnegie Mellon University, consisting of a solid red rectangle with the text 'Carnegie Mellon University' in white, serif font, stacked in three lines: 'Carnegie', 'Mellon', and 'University'.

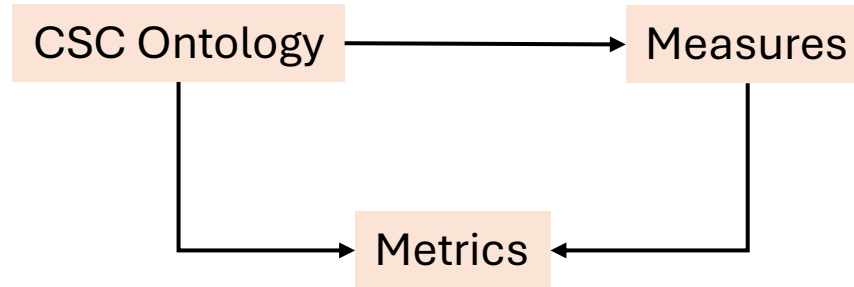
# Automated Critical Security Control Enforcement Assessment

- Events generated in end-host devices provide insights about both **ongoing attacks** and the **security posture** of the system under protection.
- Organizations like NIST (National Institute of Standards and Technology) and CIS (Center for Internet Security) provide security controls to enforce defense-in-depth and defend against attacks.
- CIS provides 18 security controls and 153 safeguards

# Outline

- CSC Ontology, Measures and Metrics
- Traditional Approach of CSC Assessment
- Motivation and Challenges
- Problem Statement
- Approach Overview
- Prompt Engineering
- CSC Ontology
- Measures and Metrics Generation
- Measures and Metrics Implementation Steps Generation
- Evaluation
- Conclusions

# CIS CSC Enforcement Assessment: CSC Ontology, Measures and Metrics



- **CSC Ontology:** Critical information that will help to identify Key Measurement Indicator (KMI or measure) and Key Enforcement Indicator (KEI or metric) for a safeguard.
  - What are the threat actions and corresponding cyber observables each safeguard targets?
  - What are the KMI or cyber-measurable features that need to be measured for each safeguard?
  - What is the measurement category, and what is the measurement approach?
- **Measure:** A concrete and objective attribute for the corresponding CSC safeguard, such as
  - Number of detected malicious IP addresses
  - Number of the unused IP addresses in the target organization.
- **Metric:** An abstract and subjective attribute (composition of multiple measures) such as
  - Coverage or percentage of malicious IP that can be detected.
  - Freshness or how fast a new asset is discovered

# CIS CSC Enforcement Assessment: Traditional Approach

## **CSC Safeguard 5.3: Disable Dormant Accounts**

Delete or disable any dormant accounts after a period of 45 days of inactivity, where supported.

### **CSC Ontology:**

**Observable:** Dormant account activities.

### **Measurement Approach:**

- Passive monitoring of account activity audit logs (ETW).
- Scripting to check last activity time of an account.

### **Measures:**

- M1 = Count of dormant accounts.
- M2 = Count of dormant accounts still enabled.
- M3 = Number of timely (45 days) deactivated dormant accounts.

### **Metrics:**

- **Compliance Completeness** =  $M2 / M1$ .
- **Timeliness of the deactivation** =  $M3 / (M1 - M2)$ .

### **Measure Monitoring Implementation:**

- System account configuration to check for last account activity.
- Account activity ETW event ID to monitor.

# CSC Enforcement Assessment: Motivation and Challenges

## Motivation

- Frequent updates of CSC require repetitive manual task of measures and metrics generation.
- No well-defined automated measures and metrics are developed to validate the enforcement of these CSCs.
- Analyzing the implementations of security products to validate CSC enforcement is non-trivial.
- No measures and metrics exist to check enforcement quality.
- Manual approach to develop measures and metrics, and implementation steps are time-consuming and highly dependent on the analyst's expertise.

## Challenges

- Automated extraction of critical information from safeguard description.
- Accurate knowledge base of facts (e.g., security best practices, event monitoring approach, and event ID in event logs).



## Solution

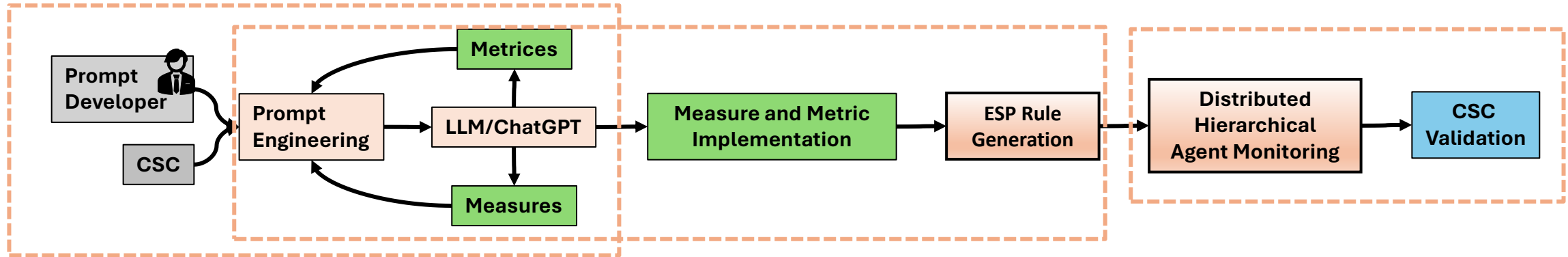
- Large Language Model (LLM) as a knowledge base and reasoner.
- Prompt engineering to elicit reasoning from LLM.
  - Zero-shot Prompting.
  - Few-shot Prompting.
  - Chain-of-Thought Prompting.

# CSC Enforcement Assessment: Problem Statement

If a company X has invested in Y products to implement Z CSCs, our goal is to identify measures, metrics and measurement procedures to test and evaluate the quality of CSC enforcement quantitatively.

- Automated CSC ontology Extraction
- Automated measures, metrics and measurement procedures generation

# CIS CSC Enforcement Assessment Approach Overview



- Automated Measures and Metrics Development
  - CSC Ontology Generation
    - Manual Approach
    - Chain-of-Thought prompting for automation

- Automated Measures and Metrics Development
  - Chain-of-Thought prompting for automated measures and metrics

- Measures and Metric Implementation steps generation
  - Generated Knowledge prompting

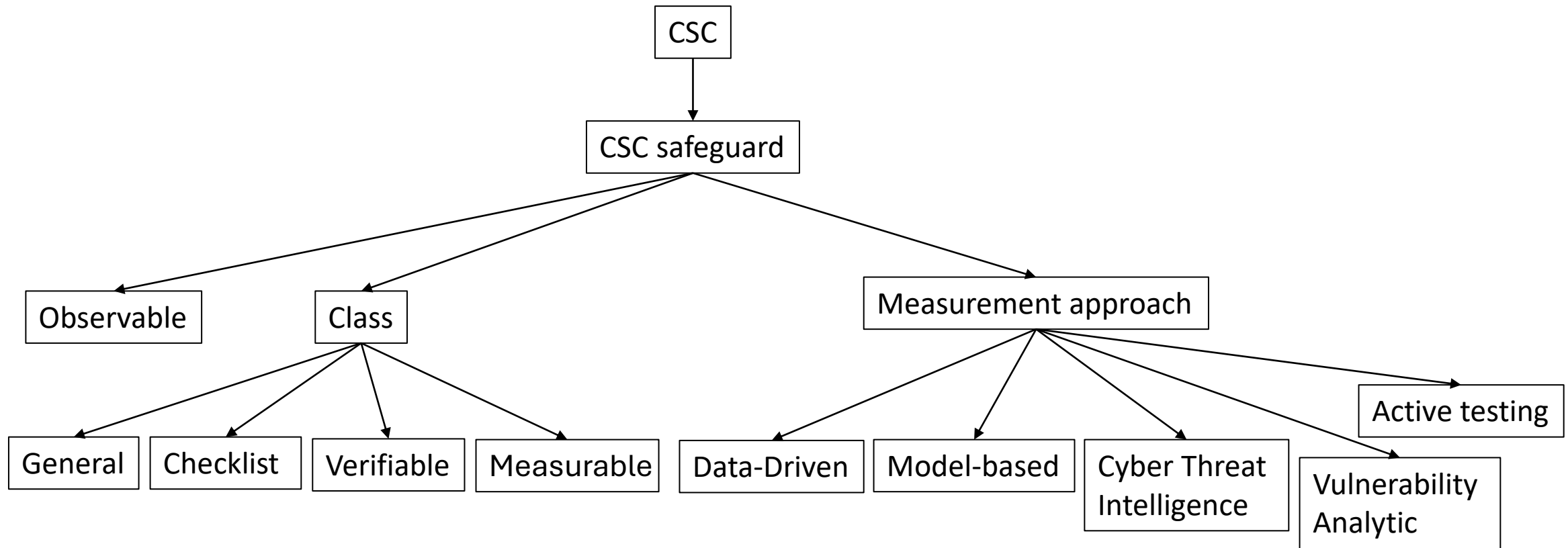


# CIS CSC Enforcement Assessment: Prompt Engineering

- **Prompt Engineering:** A prompt is a task instruction described in natural language text that requests the LLM to perform a specific task. Prompt development strategy to follow [1]-
  - **Use low-level pattern:** Use various patterns about the expected output instead of using terms that require background knowledge
  - **Itemize instruction:** Turn descriptive attributes into bulleted lists. If there are any negation statements, turn them into assertion statements.
  - **Break down a task** into multiple simpler tasks.
  - **Enforce constraint:** Add explicit textual statements of output constraints.
  - **Specialize the instruction:** Customize the instructions so that they directly speak to the intended output.

# CSC Ontology: Manual Approach

Analyzing the guidelines of 153 CIS CSC safeguard, we identify following critical information to extract from each safeguard description: Cyber observable, Measurement class and Measurement approach.



# CSC Ontology: Zero-shot vs Few-shot Prompting

- **Zero-shot prompting:** No examples of task is presented to the LLM, rather a query as task is provided to the LLM.

Zero-shot prompting does not able to differentiate different critical information in the CSC Ontology.

Question: For the following CIS CSC safeguard description enclosed in the curly braces, extract the following CSC Ontology:  
CIS CSC Safeguard: {...}

CSC Ontology to extract:  
Cyber Observable: ...  
Class: ...  
Evaluation Method: ...  
Measures:  
Metrics:

Format the output as JSON with the following keys:

Output format:  
Observable  
Class  
Evaluation Method  
Measures  
Metrics

LLM

Answer: {  
 "Observable": ["Frequency of inventory updates", "Completeness of inventory", "Accuracy of asset information",  
 "Utilization of automated discovery tools", "Timeliness of updates", "Adherence to policy", "Incident response metrics",  
 "Compliance monitoring", "Number of unauthorized devices", "Security breach metrics", "Asset lifecycle management", "Training effectiveness"],  
 "Class": [ "Checklist", "Verifiable", "Measurable"],  
 "Evaluation Method": ["Data-driven", "Model-based", "Active testing"],  
 "Measures": ["Percentage of known assets", "Accuracy of information", "Utilization percentage of automated tools",  
 "Time taken for updates", "Adherence to documented policies", "Response time to security incidents", "Compliance with control and regulations",  
 "Number of unauthorized devices", "Frequency and impact of security breaches", "Effectiveness of asset lifecycle management", "Effectiveness of training programs"],  
 "Metrics": ["Completeness = (Known Assets / Total Expected Assets) \* 100", "Accuracy = (Correctly Identified Assets / Total Identified Assets) \* 100",  
 "Utilization Percentage = (Utilized Automated Tools / Total Tools Available) \* 100", "Timeliness = Time taken for updates",  
 "Adherence to Policy = (Adhered Instances / Total Instances) \* 100", "Response Time = Time taken to respond to security incidents",  
 "Compliance Percentage = (Compliant Instances / Total Instances) \* 100", "Number of Unauthorized Devices", "Security Breach Index = (Frequency \* Impact) / 2",  
 "Effectiveness of Asset Lifecycle Management = (Effective Instances / Total Instances) \* 100", "Effectiveness of Training Programs = (Effective Instances / Total Instances) \* 100"]  
}

# CSC Ontology: Zero-shot vs Few-shot Prompting

- **Few-shot prompting:** A demonstration of the task at hand is provided to the LLM as an example.
  - Chain-of-Thought (CoT) is a form of few-shot prompting where the reasoning steps are divided into multiple thought steps and provided to the LLM as example.
- LLM are few-shot learners [1].

For the following security control safeguard description enclosed in the curly braces, extract the following information:

**Observable:** which cyber detectable observable to monitor to assess the safeguard enforcement

**Class:** Checklist or verifiable or Measurable (A safeguard can belong to multiple classes). A checklist class means we can assess the safeguard through scripting, A verifiable class means we can verify the safeguard by checking system configuration, and A measurable class means we have to use data-driven analytics to assess the safeguard. You must also give the reason for why you think a safeguard belongs to the detected class.

**Evaluation Method:** Data-driven or Model-based or Active testing (A safeguard can be evaluated using multiple evaluation methods). A Data-driven evaluation method indicates that we have to generate statistics from event logs, traffic flows, and user activities to evaluate the enforcement of the safeguard; A model-based evaluation approach indicates that we have to use the configuration of the system to evaluate the enforcement of the safeguard; A active testing evaluation method indicates that we have to probe the system to evaluate the enforcement of the safeguard. You must also explain why you think a safeguard can be evaluated using the detected evaluation method.

Format the output as JSON with the following keys:

Observable, Class.type, Class.explanation, Evaluation.method

**Question:** Safeguard: ... CIS CSC 1.1 description.

**Thought 1:** If this safeguard is implemented, the organization should have an asset inventory list containing asset details. If it is not enforced, there will be no asset inventory, so no asset details. Thus, observables will be the presence/absence of asset inventory and asset attribute details. We can assess the safeguard by checking the presence/absence of asset inventory and asset attributes recorded in the inventory. Thus, the class will be checklist.

**Thought 2:** Since we have to check recorded asset details and how fast an asset is recorded in the inventory to measure the enforcement of this safeguard, the evaluation method will be Data-driven.

**Question:** {Safeguard: ... CIS CSC 12.2 description}

LLM

**Answer:**

```
{ "Observable": "Insecure network architecture",  
  "Class": { "type": ["Measurable"],  
    "explanation": "The safeguard requires data-driven analytics to assess the effectiveness of  
segmentation, least privilege, and availability measures." },  
  "Evaluation" : {  
    "Method": { "type": ["Data-driven"],  
      "explanation": "The evaluation method involves generating statistics from event logs,  
traffic flows, and user activities to assess the enforcement of the safeguard." } }
```

# Measures and Metrics Generation using CoT Prompting

**Chain-of-Thought (CoT) Prompting:** A form of few-shot prompting for reasoning tasks where a prompt consists of a triplet <input, chain of thought, output> that is presented to the LLM as a question.

- According to Wei *et al.* [2], CoT prompting significantly improves the complex reasoning capabilities of LLMs. Reasoning is an **emergent** property of LLM.

**Question:** Safeguard: ... (CIS CSC safeguard 1.5 description).

Answer:

**Thought 1:** If this safeguard is implemented, we will see the passive discovery tool's scan result. Thus, we can measure the following things-

Measures: M1 = Count of properly configured passive asset discovery tools, M2 = Count of improperly configured passive asset discovery tools, M3 = Count of organization's networks that are covered by properly configured passive discovery tools, M4 = Count of organization's networks that are not covered by properly configured passive discovery tools, M5 = Count of enterprise networks, M6 = last scan time, M7 = 2nd last scan time, M8 = Time when asset inventory is updated.

**Thought 2:** To calculate the coverage score, you can combine measure M3 and M4; to calculate the Configuration compliance quality, you can combine measure M1 and M2; To calculate scan rate, we can use measure M6 and M7; To calculate Asset inventory update rate, we can use measure M8 and M6.

Metrics: Coverage =  $M3 / (M3 + M4)$ , Configuration compliance quality =  $M1 / (M1 + M2)$ , Scan rate = time difference between two consecutive scan =  $M6 - M7$ , Assent inventory update rate (freshness) =  $1 / (\text{the time when asset inventory is updated} - \text{last scan time}) = 1 / (M8 - M6)$ .

Now, for the following security control safeguard description enclosed in the curly braces, extract the following information:

Measures: what to measure to assess the safeguard?

Metrics: How to measure the effectiveness of the safeguard by using extracted measures?

After extracting measures, Use the extracted measures to generate metrics. Format the output as JSON with the following keys: Measures, Metrics.

**Question:** {Safeguard: ... (CIS CSC safeguard 1.1 description)}

LLM

**Answer:**{

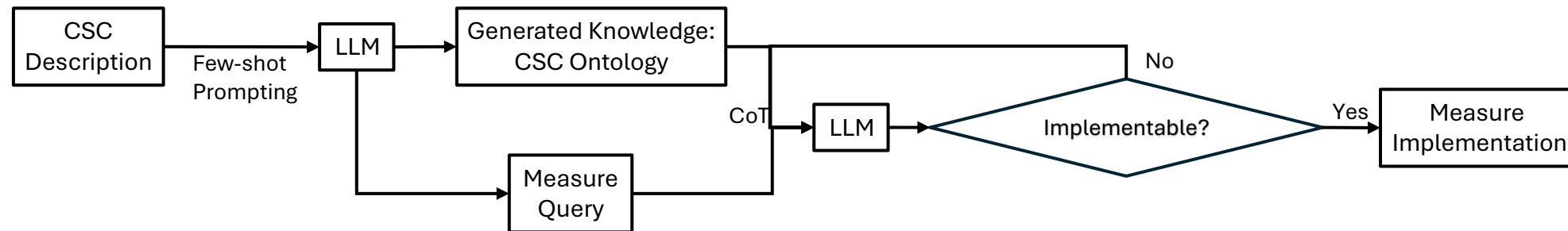
"Measures": {"M1": "Count of end-user devices in the inventory", "M2": "Count of network devices in the inventory", "M3": "Count of non-computing/IoT devices in the inventory", "M4": "Count of servers in the inventory", "M5": "Count of mobile end-user devices managed by MDM tools", "M6": "Count of assets with static network addresses", "M7": "Count of assets with approved network connectivity", "M8": "Count of assets with up-to-date records", "M9": "Count of assets physically connected to the infrastructure", "M10": "Count of assets virtually connected to the infrastructure", "M11": "Count of assets remotely connected to the infrastructure", "M12": "Count of assets within cloud environments", "M13": "Count of assets regularly connected to the network, even if not under control", "M14": "Count of assets reviewed and updated bi-annually or more frequently"},

"Metrics": {"Inventory Completeness": " $(M1 + M2 + M3 + M4) / \text{Total Expected Assets}$ ", "Approval Rate": " $M7 / (M1 + M2 + M3 + M4)$ ", "MDM Coverage": " $M5 / \text{Total Mobile End-User Devices}$ ", "Static Address Rate": " $M6 / (M1 + M2 + M3 + M4)$ ", "Up-to-date Record Rate": " $M8 / (M1 + M2 + M3 + M4)$ ", "Connection Method Distribution": {"Physical": " $M9 / (M1 + M2 + M3 + M4)$ ", "Virtual": " $M10 / (M1 + M2 + M3 + M4)$ ", "Remote": " $M11 / (M1 + M2 + M3 + M4)$ ", "Cloud": " $M12 / (M1 + M2 + M3 + M4)$ ", "Regularly Connected": " $M13 / (M1 + M2 + M3 + M4)$ "}, "Review and Update Rate": " $M14 / \text{Total Assets}$ "}}

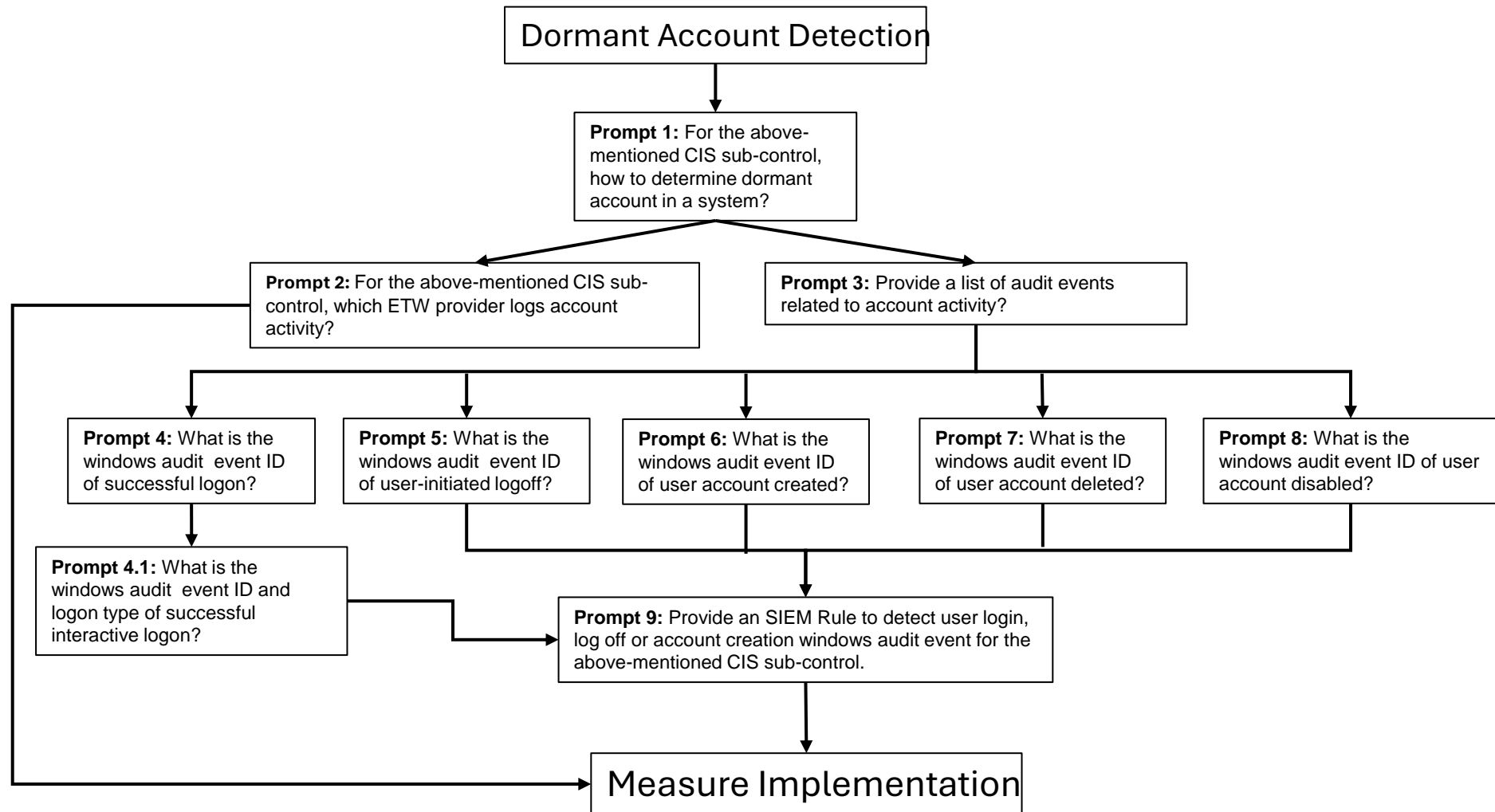
# Measures and Metrics Implementation Steps Generation using Generated Knowledge Prompting

**Generated Knowledge Prompting:** A form of prompting where knowledge is generated and extracted from the LLM first, then the generated knowledge is provided as additional input to the LLM to answer a question.

- According to Liu *et al.* [3], generated knowledge prompting facilitates common sense reasoning tasks in LLMs.



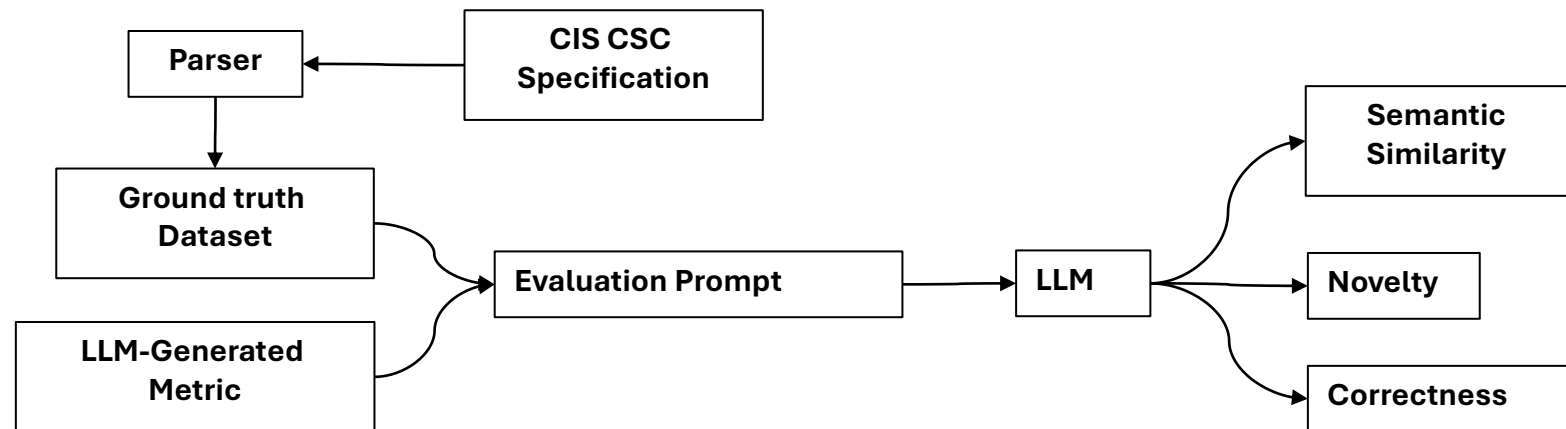
# Measures and Metrics Implementation Steps Generation using Generated Knowledge prompting



# Evaluation of Generated Measures and Metrics using LLM

- Time-consuming and resource-intensive task of manual comparison.
- Criteria to evaluate.
- Reliability of the automated evaluation.
- Re-usability of prompts for different usecases.

RLHF (Reinforcement Learning from Human Feedback) involves creating an (LLM) reward model that scores model responses during training, this shows a new direction to **use LLMs themselves as evaluators**.





# Evaluation of Generated Measures and Metrics using LLM: Evaluation Prompt

## Evaluation Prompt

- Context such as definition of each evaluation criteria, safeguard description,
- LLM-generated measures and metrics,
- Human-generated measures and metrics.

Given the context, LLM generated response and human-annotated response, evaluate LLM-generated response with respect to annotated response based on the following criteria:

1. Semantic similarity: Is the LLM-generated response semantically similar to the annotated response? While calculating the semantic similarity score, ignore the measure name mismatch. Provide a score between 1 to 10.
2. Novelty: Does the LLM-generated response contain new metrics that are not mentioned in the annotated response? Provide a score between 1 to 10.
3. Metrics correctness: are the measures composed correctly to generate metrics? Provide a score between 1 to 10.

Context: safeguard description

LLM-generated response: "..."

Human-annotated response: "..."

Provide the score with explanation for each criterion in JSON format.

LLM

**Answer:**

Semantic similarity: {Score:..., Explanation: ...}

Novelty: {Score:..., Explanation: ...}

Metrics Correctness: {Score:..., Explanation: ...}

# Evaluation of Generated Measures and Metrics using LLM: **Semantic Similarity, Novelty and Correctness**

## Semantic Similarity:

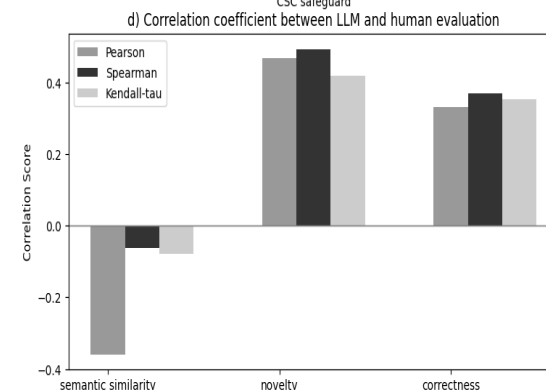
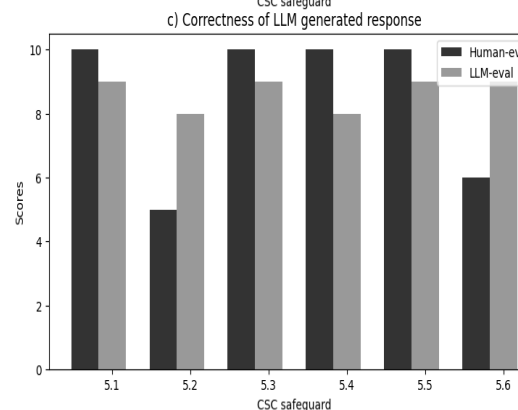
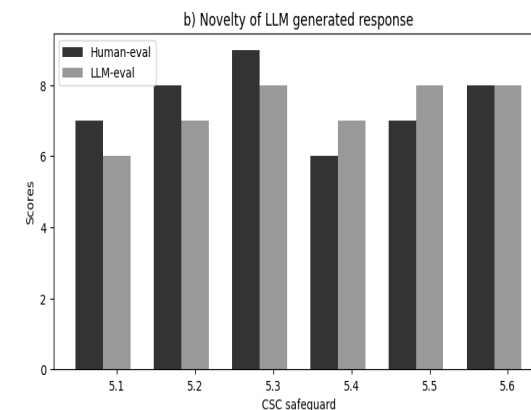
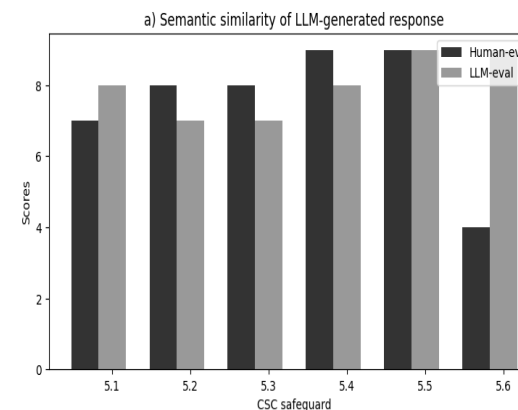
- LLM-generated metrics cover all the human-generated metrics, when any dependent safeguard consideration is unnecessary.
- LLM-generated metrics differ from the human-labeled ones when limited context is given. (safeguard 5.6).

## Novelty:

- LLM generated metrics provide new metrics for all of the safeguard we evaluated.
  - One of the major findings of this paper.

## Correctness:

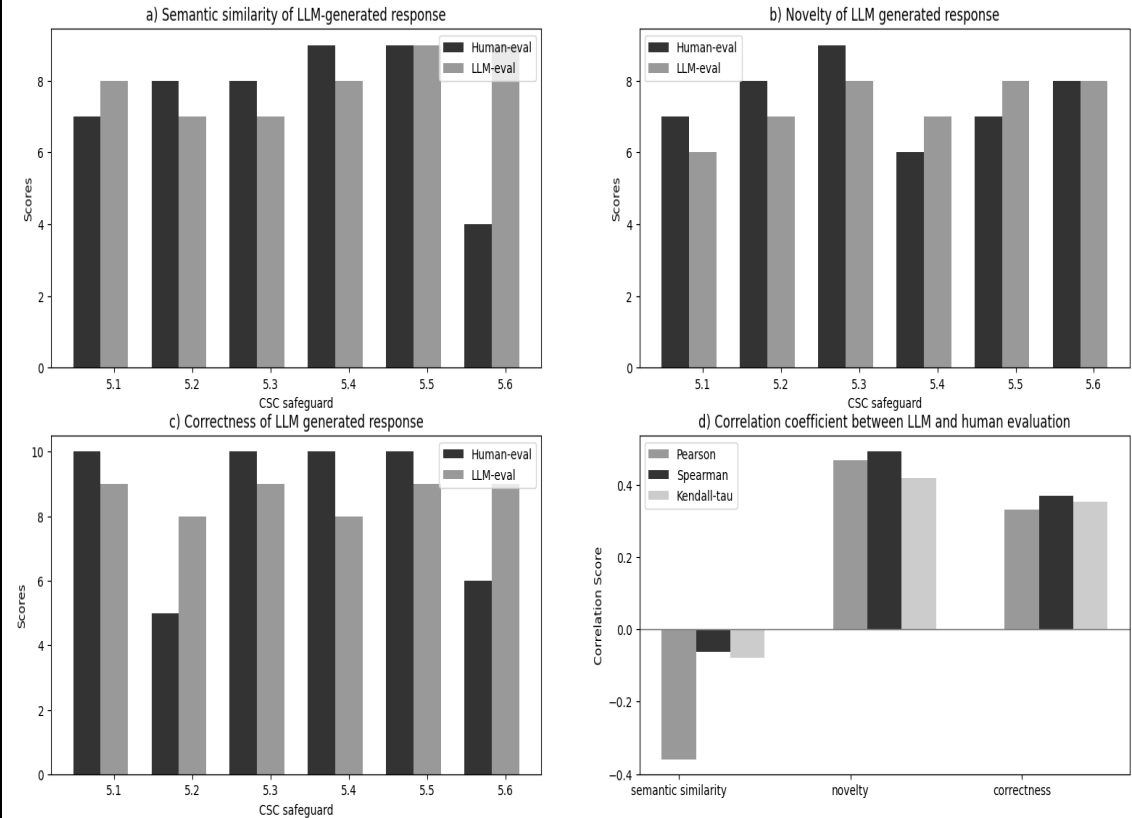
- LLM evaluation prompt can detect whether measures are composed correctly to generate metrics.



# Evaluation of Generated Measures and Metrics using LLM: Reliability

To determine whether LLM evaluation aligns with human evaluation

- Pearson, Spearman, and Kendall Tau correlation coefficient to determine the correlation between human and LLM evaluation.
- Pearson, Spearman, and Kendall Tau correlation coefficient of -0.39, -0.06, and -0.08 for *semantic similarity*.
- Pearson, Spearman, and Kendall Tau correlation coefficient of 0.47, 0.49, and 0.42 for *novelty*.
- Pearson, Spearman, and Kendall Tau correlation coefficients of 0.33, 0.37, and 0.35 for *correctness*.



# Evaluation of Generated Measures and Metrics using LLM: **Re-usability of prompt**

## **Security Control from Different Organizations**

- Our evaluation is based on CIS CSC.
- We tested NIST CSF. Original prompt can generate accurate Measures and Metrics.

## **Different LLM Family**

- ChatGPT 3.5 and ChatGPT 4.0: Original prompt works well. No fine-tuning required.
- LLAMA, Gemini: Fine-tune prompt for specific LLM family.

# Conclusions

In Summary, we present:

- **Automated CIS CSC enforcement assessment** using prompt engineering:
  - Automate the extraction of **CSC Ontology** from CSC safeguard descriptions using Few-shot prompting.
  - Generate **novel Measures and Metrics** from CSC safeguard descriptions using CoT prompting.
  - Generate **Measures and Metrics** implementations steps using Generated Knowledge prompting.
  - **Demonstrate CIS CSC enforcement assessment** using CSCMonitor.

**Thank You.**

**Questions?**

# References

1. How much does a data breach cost? <https://www.ibm.com/security/data-breach>. [Online; accessed 15 March 2024]
2. The state of ransomware 2022. <https://www.sophos.com/en-us/content/state-of-ransomware>. [Online; accessed 15 March 2024]
3. Special report m-trends 2021. <https://www.mandiant.com/resources/m-trends-2021>. [Online; accessed 15 March 2024]
4. 2024. Center for Internet Security- Critical Security Control, 2021. <https://www.cisecurity.org/controls/cis-controls-list>.
5. 2024. CIS Benchmark. <https://www.cisecurity.org/cis-benchmarks>.
6. 2024. CIS Controls Assessment Specification. <https://controls-assessment-specification.readthedocs.io/en/stable/index.html>.
7. 2024. CIS Controls Measurement Companion Guide. <https://www.cisecurity.org/insights/white-papers/a-measurement-companion-to-the-cis-critical-controls>.
8. 2024. CIS Controls Self Assessment Tool (CIS CSAT). <https://www.cisecurity.org/controls/cis-controls-self-assessment-tool-cis-csat>.
9. S. Mishra, D. Khashabi, C. Baral, Y. Choi, and H. Hajishirzi, "Reframing instructional prompts to gptk's language," ArXiv, vol. abs/2109.07830, 2021.
10. S. Mishra, D. Khashabi, C. Baral, Y. Choi, and H. Hajishirzi, "Reframing instructional prompts to gptk's language," ArXiv, vol. abs/2109.07830, 2021.
11. Mohiuddin Ahmed, Jinpeng Wei, Ehab Al-Shaer. 2024. "Prompting LLM to Enforce and Validate CIS Critical Security Control." Accepted at ACM SACMAT 2024.
12. Mohiuddin Ahmed, Jinpeng Wei, Ehab Al-Shaer. 2023. "SCAHunter: Scalable Threat Hunting Through Decentralized Hierarchical Monitoring Agent Architecture." In: Arai, K. (eds) Intelligent Computing. SAI 2023. Lecture Notes in Networks and Systems, vol 739. Springer, Cham. [https://doi.org/10.1007/978-3-031-37963-5\\_88](https://doi.org/10.1007/978-3-031-37963-5_88)
13. Mohiuddin Ahmed, Ehab Al-Shaer. 2019. "Measures and Metrics for the Enforcement of Critical Security Controls: a Case Study of Boundary Defense." In Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security (Nashville, Tennessee, USA) (HotSoS '19). Association for Computing Machinery, New York, NY, USA, Article 21. <https://doi.org/10.1145/3314058.3317730>
14. T. Brown, B. Mann, N. Ryder, ..., D. Amodei, "Language models are few-shot learners," in Advances in Neural Information Processing Systems (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020
15. J. Wei, X. Wang, D. Schuurmans, M. Bosma, brian ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," in Advances in Neural Information Processing Systems (A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds.), 2022.