Carnegie Mellon University
Software Engineering Institute

CONSIDERATIONS FOR EVALUATING LARGE LANGUAGE MODELS FOR CYBERSECURITY TASKS

Jeff Gennari
Shing-hon Lau
Samuel Perl
Joel Parish (OpenAI)
Girish Sastry (OpenAI)

February 2024

Introduction

Artificial Intelligence (AI) policy discussions in government and industry have been dominated by concerns about the potential societal-scale risks stemming from recent advancements in artificial intelligence technologies like those based on large language models (LLMs). One concern is malicious use, wherein a modestly skilled individual or group with malicious intent could perhaps leverage an LLM to amplify their own capabilities to cause mayhem. Other concerns reflect the potential of future AI systems to autonomously run amuck. For example, discussants worry that such autonomous systems could take control of critical systems or independently orchestrate influence operations. Developing more practically realistic LLM evaluations will likely help to avoid inappropriately extrapolating into implausibly extreme scenarios or dismissing LLM capabilities altogether.

One notable domain where malicious actors have already sought to leverage LLMs is cybersecurity. In an ideal world, we could directly assess the degree to which LLM capabilities amplify the capabilities of malicious actors. Such an assessment is virtually impossible to perform; a next-best option is to assess the cybersecurity capabilities of LLMs and envision how malicious actors might be able to leverage those capabilities.

Part of the challenge in assessing the cybersecurity capabilities of LLMs is that many current evaluations are implemented as simple benchmarks and do not provide the information necessary to truly assess aptitude. Without a clear understanding of how an LLM performs on applied and realistic cybersecurity tasks, decisions makers do not have the information they need to assess opportunities and risks. We contend that practical, applied, and comprehensive evaluations are required to assess cybersecurity capabilities. Realistic evaluations reflect the complex nature of cybersecurity and provide a more complete picture of cybersecurity capabilities. Without a clear understanding of how an LLM performs on applied and realistic cybersecurity tasks, decisions makers do not have the information needed to assess opportunities and risks.

In this paper, we describe an approach for holistically evaluating LLM capabilities, focusing on cybersecurity expertise. An actor with cybersecurity knowledge poses a particular threat given the nature of the technologies that make up the critical infrastructure of the modern world. We have already seen evidence of actors using generative AI technology to aid in cyber intrusion activities. Famous examples of misuse include coercing LLMs to divulge sensitive information or using LLMs to automate and conduct malicious cyber activities [WormGPT 2024, Shimony 2023, Sayak 2023].

Other important cybersecurity-capability-related questions include the following:

- Can an LLM like GPT-4 write novel malware?
- Will LLMs become critical components of large-scale cyber attacks?
- Can we trust that LLMs provide cybersecurity experts with reliable information?

The answer to these questions will depend on the analytic methods chosen and their results. Unfortunately, the quality of current public methods and evaluations for evaluating cybersecurity capabilities of LLMs does not provide a comprehensive assessment of capabilities. Current evaluation strategies, such as having AI systems take cybersecurity certification exams, usually focus only on the factual knowledge that LLMs may have already absorbed and lack other applied knowledge, such as planning realistic goals, identifying new scenarios, or adapting to changing environmental circumstances.

We consider ways to help researchers build cybersecurity evaluations that provide more complete assessments of LLM cyber capabilities. Fundamentally, we argue that practical, real-world evaluations are necessary to understand the cyber capabilities of LLMs; fact-based examinations are insufficient. Better cybersecurity evaluations can help policy and decision makers assess the potential benefits and risks LLMs pose in cybersecurity contexts. To develop recommendations for improving cybersecurity evaluations of LLMs, we first consider the nature of cybersecurity activities and highlight specific opportunities and challenges for using LLMs to perform them. We then explore how LLMs capture and apply cybersecurity knowledge in general. Next, we consider specific cyber-related tasks in terms of how they can be supported by LLMs. Finally, we propose a set of recommendations to help cybersecurity practitioners effectively evaluate LLMs' cybersecurity capabilities.

1 The Nature of Cybersecurity Expertise

Expertise is a concept that is well studied in cognitive science. The Theory of Expert Competence notes that expertise requires good domain knowledge, the skills needed to make tough decisions, the ability to select appropriate decision strategies, and the match of the task itself to the experts' skills [Shanteau 1992]. Experts are confident in their abilities, can clearly communicate, and can adapt to changing circumstances. Experts can generally handle real-world situations, which often have problems that require strategic thinking, risk assessment, and tradeoffs. This definition works well in the cybersecurity domain, where it is not enough to simply have a command of facts and concepts;

experts must also be able to apply their knowledge to real-world situations, use complex tools, and balance risks and tradeoffs to achieve security goals.

Cybersecurity is a wide-ranging field that includes many concerns. Subfields may be technical in nature, such as vulnerability analysis, exploitation, reverse engineering, computer forensics, and cryptography. Other concerns are more business and policy oriented, such as risk management, insider threat mitigation, and creating policies for incident response. Some roles are defensive in nature and focus on detecting and preventing compromise; for instance, network defenders need deep knowledge of an organization's assets and information technology (IT) infrastructure in addition to an understanding of business concerns and goals. Other roles, such as penetration testers, require an attacker mindset and a deep technical knowledge of software exploitation techniques and tools for a variety of systems. Regardless of the role, basic factual knowledge of cybersecurity is necessary. For example, a malware analyst should have a basic knowledge of malware.

However, to demonstrate expertise, a more nuanced understanding is needed than textbook knowledge combined with a little common sense. For example, malware analysts are more helpful if they can connect technical analysis of digital artifacts to perceived adversary goals and to the circumstances of the compromise before deciding what new actions to conduct in their assessment. For example, it may be the case that understanding the malware's capabilities is a more important goal than actually stopping the specific attack (or vice versa). In other areas of cybersecurity, red team testers must balance the always-present risk of detection with their ability to perform actions on machines to achieve their objectives. They need to understand when to push ahead and when to back off, and what they might be giving up when doing so. These are but a few examples of the nuances that cybersecurity expertise requires.

Seasoned cybersecurity professionals possess not only cybersecurity knowledge, but also sound judgement and often creativity to apply what they know to real tasks in operational scenarios. Security experts can quickly process new information and change tactics to adapt. Such strategic thinking may be counterintuitive when viewed in isolation but reasonable when considered as part of a larger goal. For example, when deciding how to balance system defense with business operations, security engineers carefully consider both concerns in the full context of the organization. They may decide to accept additional risk to fulfill a business objective, choose certain technologies due to outside regulatory requirements, or select a technique that appears inefficient but maintains stealth to monitor a compromise. Such risk appetite is highly dependent on context, however. In other scenarios, such as medical software, the risk calculation is much different, given the possible threat to health and safety.

The ability to contextualize, apply, and adapt knowledge, tools, and skills in a variety of real situations is the essence of cybersecurity expertise. Factual knowledge of cybersecurity is necessary but insufficient to convey expertise. Human evaluations often test basic knowledge as a prerequisite for deeper assessments. More sophisticated evaluations emulate real-world situations. These can take the form of game-like competitions, such as capture-the-flag contests or penetration tests. In these exercises, there are multiple, layered, and sometimes contradictory objectives designed to test the expert's ability to apply their technical knowledge and think strategically. The specifics of the exercise matter, but in general, these types of assessments are well regarded because they provide a more comprehensive assessment of cybersecurity expertise in a realistic context. This type of evaluation

could provide similar insights into the cybersecurity expertise of AI systems, as described in the following section.

2 Cybersecurity Evaluation Challenges

Assessing cybersecurity is a wide-ranging endeavor that incorporates a diverse set of measurements, including cyber-operator cognitive complexity [Paul 2017, Nyre-Yu 2020, Bellovin 1992], computational complexity [Huang 2015], and foundational computer science knowledge [Ingram 2022]. Unlike traditional methods for evaluation (e.g., accuracy) that skew towards fact-based knowledge, assessing cybersecurity aptitude often depends on context regardless of the specific task. For example, evaluating the severity of a buffer overflow vulnerability requires experts to consider factors beyond the technical conditions that cause the vulnerability. They must also understand vulnerable software configuration (e.g., level of privileges, operating system), where and how widely the vulnerable component is deployed, the consequences of exploitation, possible mitigations, and so on. On the defensive side, high-level concepts, such as secure design principles, require the ability to comprehend and relate abstractions to software elements in conjunction with other quality attributes, possibly unrelated to or in tension with security goals.

Perhaps the most challenging aspect of evaluating real cybersecurity tasks is that they are often complex, dynamic, and require broader context to fully assess. Consider a traditional network intrusion where an attacker seeks to compromise a system. During the attack, adversaries may repeatedly change tactics based on defender actions and vice versa. Depending on the attacker's goals, they may emphasize stealth or attempt to quickly maximize damage. Defenders may choose to simply observe the attack to learn adversary tendencies or gather intelligence or immediately expel the intruder. The variations of attack and response are impossible to enumerate in isolation. This inherent complexity means that cybersecurity experts rarely face the exact task more than once. Perhaps this is why cybersecurity is often described as a "wicked problem" [Rittel 1973] that inherently possesses characteristics that make solutions difficult to measure.

Evaluations must be multifaceted and comprehensive because cybersecurity expertise requires a variety of skills, including highly specialized technical knowledge, effective communication skills [Sushlia 2023], the ability to balance competing goals, and more. Adding to the complexity is the fact that cybersecurity is an umbrella term that includes numerous domains, each of which requires specific knowledge and skills. For example, the skillset needed to be a computer forensics expert is different than that of a network security engineer. The diversity of cybersecurity concepts combined with the technical aptitude required by each domain forces cybersecurity professionals to become highly specialized. Despite these challenges, cybersecurity professionals tend to possess general knowledge but demonstrate expertise through their ability to perform hands-on, specialized, and domain-specific tasks. These tasks generally demonstrate cybersecurity experts' ability to make tradeoffs, adapt to changing conditions, and convert theory to practice. Thus, cybersecurity evaluations should include a component for assessing general knowledge and another for demonstrating proficiency via applied tasks in specific contexts.

3 Evaluating Cybersecurity Expertise in LLMs

Evaluating LLMs for factual cybersecurity knowledge is necessary, but insufficient, to truly assess aptitude and ultimately determine how reliable (or dangerous) an AI agent may be in the context of cybersecurity. Thus, when evaluating LLMs for cybersecurity, it is crucial to go beyond tasks that rely on rote memorization. Such evaluations cater to the strengths of LLMs by their nature—namely factual recall and in-context learning—while overlooking the complexity of performing these tasks in the real world.

As an analogy, consider how college-level students are evaluated. Written exams may be used to assess foundational cybersecurity knowledge, but courses typically include a significant amount of applied homework. Often, these assignments require chaining together and applying theory in new circumstances. For example, consider an assignment in a college-level reverse engineering class that requires students to perform dynamic analysis of a software binary to discover and extract a cryptographic key. This task requires basic factual knowledge, like the definition of a cryptographic key, an understanding of assembly language and the underlying operating system, and a notion of what it means to perform dynamic analysis. However, knowing the definition of dynamic analysis is not sufficient to achieve the assignment goal. Students need to appreciate that running the software in a controlled way will allow them to find and recover the key. To do this, students must know the telltale signs that data in the program is actually a cryptographic key versus the vast amounts of other program data. Recognizing the relevance of data represents a deeper understanding of the context and application. This understanding then needs to be connected to concrete activities to recover the key (e.g., selecting the proper tools to run and monitor the software). Finally, debugging often requires a significant planning and trial-and-error approach because it is common to restart the process as more information is gained. Understanding when, where, and how to restart the process is perhaps the most sophisticated use of knowledge—one often earned through experience and frustration.

Complex tasks, like the key recovery described above, require a fusion of basic information with the ability to perform more abstract tasks in service of larger goals. Using such a task to evaluate an LLM provides a much more complete picture of cybersecurity capabilities. Clever prompting aside, the connection and synthesis of concepts and tactics should be a core part of any evaluation. With these considerations in mind, we propose that evaluations of LLMs mirror the approach used in higher education by breaking assessment techniques into three levels:

1. **Theoretical Knowledge Evaluation**: This level examines the textbook understanding that an individual possesses. In the context of LLMs, this might involve assessing the model's ability to accurately define cybersecurity terms, explain core concepts, or correctly solve multiple-choice questions about scenarios. This is analogous to a quiz or multiple-choice section of a college-level midterm exam.
2. **Practical Knowledge Evaluation**: This level focuses on testing the ability of the student, or the LLM, to provide practical solutions to self-contained cybersecurity problems. For example, a practical knowledge evaluation might involve free-form response-writing code that exploits a vulnerability within a given code snippet.

3. **Applied Knowledge Evaluation**: At this level, the student or LLM is assessed based on their ability to provide practical solutions to achieve higher level objectives in open-ended situations. An example of an applied knowledge evaluation might involve gaining root access to a system from outside the network, performing lateral movement, and obtaining a flag on a second system.

Our sense is that there are few current LLM evaluations that assess practical and applied knowledge. This deficiency can result in an incomplete picture of LLM cybersecurity capabilities and could lead to biased assessments which cater to the strengths of LLMs while ignoring their weaknesses. The nature of many cybersecurity tasks makes all three of these evaluation strategies important. Like the debugging example above, many cybersecurity tasks require foundational knowledge and the ability to apply that knowledge towards achieving a goal. Real-world cybersecurity tasks are rarely limited to information retrieval; rather, they are often complex, multifaceted, and full of indirect nuance.

4 Practical Hurdles to Evaluating LLMs

Successful and accurate evaluation of LLM performance within a specific domain of expertise, such as cybersecurity, requires overcoming three practical hurdles:

1. Creating a set of questions to be answered by the LLM.
2. Posing that set of questions to the LLM and collecting the responses.
3. Defining and computing metrics that define how well the LLM responses answer the questions.

Note that these hurdles are the same general ones that must be cleared when evaluating humans that are performing these tasks. A core difference is that, while we have a well-established understanding of human cognization—particularly with respect to how humans generalize knowledge—the same is not true for LLMs. For humans, evaluating on a set of dozens or hundreds of well-chosen questions or practical exercises is generally sufficient to gauge competency in a particular topic. For LLMs, evaluations are typically conducted across many thousands, if not millions, of questions. Two major drivers lead to this expanded dataset size. First, LLMs are physically capable of answering such a vast number of questions, whereas a human is not. Second, questions that are semantically similar to those that a human might ask may lead to dissimilar responses from an LLM, necessitating far more questions to cover the knowledge space.

The large number of required questions to evaluate the theoretical knowledge of an LLM is a sizeable obstacle when questions must be limited to a narrow domain of expertise. The common dataset-creation approach of scraping large volumes of text (e.g., from Wikipedia, X, or other large repositories) is less accessible in a narrow domain since the quantity of available material is more limited. Moreover, content that is readily available is likely already incorporated into the training dataset of the LLM, rendering it unsuitable as test data. While some of the test questions and answers will assuredly need to be crafted by human experts, it is likely that tools can be employed to create variants of expert-crafted questions by adjusting character, variable, or function names or by rewording questions while preserving the semantics. Crafting enough suitable exercises for evaluating

the practical or applied knowledge of an LLM presents an even larger challenge than crafting questions for a theoretical knowledge evaluation. Creating semantically similar and relevant variants of vulnerable code, network configurations, or entire network deployments necessitates specialized tooling that does not currently exist.

Presuming that an appropriate set of questions or exercises can be developed, the second hurdle that must be cleared is the actual administration of the evaluation. Simple question-and-answer evaluations are already relatively well-supported by existing tools, such as MLFlow [MLFlow 2024], particularly for multiple choice questions. Evaluations on practical exercises can be considerably more difficult as these may necessitate interactivity between the LLM and the evaluation environment. This interactivity is necessary to support sequences, such as the LLM attempting to execute a command on a machine, receiving feedback that a permission-denied error message was generated as a result, and then attempting to execute a modified command. Assessing troubleshooting skills is, of course, part of evaluating practical and applied knowledge.

The third and final practical hurdle to overcome is the actual computation of metrics based on the responses (or sequence of responses) provided by the LLM. Some of these metrics are well supported by various tooling, particularly for theoretical knowledge questions—question correctness on multiple-choice questions or BiLingual Evaluation Understudy (BLEU) and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores between LLM and human responses. Evaluation of output code, sequences of responses, or long-form responses are less well supported, though applying LLMs to grade responses is an emerging trend. Ascertaining the correctness of output code may require a test harness that permits executing that code and determining whether it is an adequate response to the prompt. When using an LLM to grade responses, an operator needs to ensure that the grading LLM itself has the necessary understanding of the domain. Not all errors in long-form responses are equal; errors regarding substantive facts are likely more consequential than minor errors. For humans, evaluating such responses would typically involve developing a rubric that human graders then apply. There is a need for tools that support the application of rubrics on large volumes of responses. Ultimately, perhaps the best evaluations of LLMs are those that focus on whether they assist a human in performing a task. Although the resources and planning needed to evaluate how well LLMs assist humans perform realistic tasks are generally greater than other types of assessments, they are the gold standard for determining how useful an LLM is to a cybersecurity expert; all other metrics are approximations of this approach.

5 Recommendations for Cybersecurity Evaluations

To properly judge the risks and appropriateness of using LLMs for cybersecurity tasks, evaluators need to carefully consider the design, implementation, and interpretation of their assessments. Favoring tests based on practical and applied cybersecurity knowledge is preferred to general fact-based assessments. However, creating these types of assessments can be a formidable task that encompasses infrastructure, task/question design, and data collection. The following list of

recommendations are meant to inform assessors so they can craft meaningful and actionable evaluations that accurately capture LLM cybersecurity capabilities.

Define the real-world task that you would like your evaluation to capture.

Starting with a clear definition of the task helps clarify decisions about complexity and assessment. The following recommendations are meant to help define real-world tasks:

1. *Consider how humans do it*: Existing evaluations and benchmarks are often overfitted to what was possible with models a decade ago. Starting from first principles, think about how the task you would like to evaluate is accomplished by humans, and write down the steps involved.
2. *Use caution with existing datasets:* Current evaluations within the cybersecurity domain have largely leveraged existing datasets, which can influence the type and quality of tasks evaluated. Evaluations may focus on narrow areas where datasets do exist. Moreover, these datasets are not necessarily representative of their narrow area as they are samples of convenience and may have been used in the training data of the LLM under evaluation. New datasets should be defined to support the realistic tasks. There is a need for new datasets—consisting of theoretical, practical, and applied knowledge tests and exercises—for explicitly evaluating LLM performance on real-world cybersecurity tasks.
3. *Define tasks based on intended use:* Carefully consider whether you are interested in autonomy or human-machine teaming when planning evaluations. If an LLM is used in an advisory role, then tasks may be more nuanced given the presence of human judgement. Furthermore, evaluation could skew towards the utility of answers rather than task completion in its entirety. Conversely, tasks performed autonomously may incur more risk (especially in a cybersecurity context), be judged on success or failure, and require a more substantial explanation from the LLM to properly assess it.

Represent tasks appropriately.

Most tasks worth evaluating in cybersecurity are too nuanced or complex to be represented with simple queries, such as multiple-choice questions. Rather, queries need to reflect the nature of the task without being unintentionally or artificially limiting. The following guidelines can help ensure evaluations incorporate the complexity of the task:

1. *Define an appropriate scope:* While subtasks of complex tasks are usually easier to represent and measure, their performance does not always correlate with the larger task. Ensure that you do not represent the real-world task with a narrow subtask. For example, consider vulnerability discovery: the task of identifying vulnerabilities in a single function could be very different from the task of identifying vulnerabilities in a whole real-world program.
2. *Develop an infrastructure to support the evaluation:* Practical and applied tests will generally require significant infrastructure support, particularly in supporting interactivity between the LLM and the test environment. Without adequate tooling and infrastructure for expeditiously evaluating an LLM in varied test environments, evaluating the capabilities of an LLM will be difficult. Evaluators should consider the supporting infrastructure when designing evaluations.

3. *Incorporate affordances to humans where appropriate:* Ensure your assessment mirrors real-world affordances and accommodations given to humans. For example, in real-world tasks, humans often can use tools, instrument a program using a debugger, and read documentation, etc. We do not expect humans to perform exploit development or reverse engineering in a single pass without errors. Allow LLMs multiple attempts and an ability to self-correct.

4. *Avoid affordances to humans where inappropriate:* Evaluations of humans in higher education and professional-certification settings may ignore real-world complexity. This is useful for measuring educational progress and career progression, especially in a time-limited examination. However, this fails to adequately capture the difficulty of many tasks. For instance, generating an exploit for a memory corruption vulnerability (e.g., an unconstrained *memcpy*) involves interpreting tool results, creating a proof-of-concept exploit, and dealing with various program manipulations and defenses like ASLR, stack canaries, DEP, or W^X memory protection. These tasks require significant troubleshooting, hypothesis generation, and testing. Evaluations should accurately reflect this complexity or be framed appropriately.

Make your evaluation robust.

Care must be taken when designing evaluations to avoid spurious results. Assessors should consider the following guidelines when creating assessments:

1. *Use preregistration:* Preregister how you will grade the task, possibly/ideally including partial credit. This is especially important if you are using subjective human or model-driven judgements of success.

2. *Apply realistic perturbations to inputs*: Changing the wording, ordering, or names in a question would have minimal effects on a human but can result in dramatic shifts in LLM performance. Even relatively innocuous, realistic variations, such as misspellings, could have an impact. Validate your evaluation by doing things like shuffling the input (which verifies the importance of order), assigning random labels, and assessing performance, and replace phrases with semantic equivalents.

3. *Beware of training data contamination*: LLMs are frequently trained on large corpora, including news of vulnerability feeds, Common Vulnerabilities and Exposures (CVE) websites, and code and online discussions of security. While it is a standard practice to ensure the exact language of an evaluation is not included in a model's training data, often tasks are made easier by information indirectly included in the training data. For example, LLMs can memorize information about software; therefore, evaluating an LLM's ability to identify vulnerabilities that were discussed or patched in the training data is partially testing the LLMs ability to synthesize that information, not to identify vulnerabilities in arbitrary code. While training data might not include an exact prompt and response for identifying the 'goto fail' vulnerability in Apple's CoreCrypto library, discussion (and even the name) of the vulnerability significantly biases test performance [NVD 2014]. Data about security phenomena may be difficult to isolate entirely in datasets. For example, information about a new vulnerability may be gleaned from evidence present in previous reports. However, no efforts to shield training data from contamination will undoubtedly increase the risk of skewed results. Thus, evaluators should maintain private

security-relevant information that is not discussed at all in the training data, or that was added after the training-data cutoff date.

Frame results appropriately.

Evaluations with a sound methodology can still be misleading with how they frame results. The following guidelines should be considered when interpreting results:

1. *Avoid overgeneralized claims:* Evaluations are often published along with benchmarks for current models of the new evaluation. Avoid making sweeping claims about capabilities from the task or subtask evaluated. Strong model performance in an evaluation measuring vulnerability identification in a single function does not mean that a model is good at discovering vulnerabilities in a real-world web application; it only means that it is good at vulnerability discovery in a single function. Most real-world vulnerability discovery is done without access to source code. Claiming a model has "human-level vulnerability discovery" when it requires access to source code and is only "human level" when compared to humans evaluating *single functions* is an overgeneralized claim unsupported by the evaluation. On the other hand, claiming that a system can definitely *not* do a task requires using the best elicitation methods available.
2. *Estimate best-case and worst-case performance:* LLMs may have wide variations in evaluation performance due to different prompting strategies or because they use additional test-time compute techniques (e.g., Chain-of-Thought prompting [Wei 2022]). For best-case performance estimates, consider fine-tuning a model (even on a small number of examples) or building an "agent" that can help build confidence that some level of performance is out of reach. For worst-case performance estimates, you can use measures like zero-shot performance to evaluate how LLMs handle complex, bespoke problems with no prior experience.
3. *Be careful with model selection bias:* Any conclusions drawn from evaluations should be put into the proper context. For example, if a paper introduces a new vulnerability identification evaluation and tests only small models, claiming that "LLMs are poor at vulnerability identification" is unsupported. If possible, run tests on a variety of contemporary models, or qualify claims appropriately.
4. *Clarify whether you are evaluating risk or evaluating capabilities*. A judgement about the risk of models requires a threat model. But, in general, the capability profile of the model is only one source of uncertainty about the risk. Task-based evaluations can help understand the capability of the model.

Conclusion

AI and LLMs have the potential to be both an asset to cybersecurity professionals and a boon to malicious actors unless risks are properly managed. To better understand and assess the cybersecurity capabilities and risks of LLMs, we propose developing evaluations that are grounded in real and complex scenarios with competing goals. Assessments that are based on standard, factual knowledge skew towards the type of reasoning LLMs are inherently good at (i.e., factual information recall). Instead, to get a more complete sense of cybersecurity expertise, evaluations should consider applied security concepts in realistic scenarios. This is not to say that a basic command of cybersecurity knowledge is not valuable to evaluate; rather, more realistic and robust assessments are required to judge cybersecurity expertise accurately and comprehensively. Understanding how an LLM performs on real cybersecurity tasks will provide policy and decision makers with a clearer sense of capabilities and the risks of using these technologies in such a sensitive context.

References

URLs are valid as of the publication date of this report.

[Bellovin 1992]
Bellovin, Steven Michael & Merritt, Michael. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. Pages 72-84. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*. May 1992. https://doi.org/10.7916/D8833ZSK

[Huang 2015]
Huang, Zequn; Shen, Chien-Chung; Doshiy, Sheetal; Thomasy, Nimmi; & Duong, Ha. *Difficulty-Level Metric for Cyber Security Training.* In *Proceedings of the 2015 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision*. March 2015. http://dx.doi.org/10.1109/COGSIMA.2015.7108194

[Ingram 2022]
Ingram, Lieutenant-Commander Kenneth. *Unpacking 'Wicked Problems' of Cyberspace: Conceptual Approaches for Novice Practitioners.* 2022. https://www.cfc.forces.gc.ca/259/290/308/305/ingram.pdf

[MLFlow 2024]
MLFlow—An Open Source Platform for the Machine Learning Lifecycle. *MLFlow Website*. February 1, 2024 [accessed]. https://mlflow.org/

[NVD 2014]
National Vulnerability Database. *CVE-2014-1266.* National Institute of Standards and Technology (NIST). 2014. https://nvd.nist.gov/vuln/detail/CVE-2014-1266

[Nyre-Yu 2020]
Nyre-Yu, Megan. *Comparing Cognitive Demands in Two Cybersecurity Tasks.* SAND2020-2167C. Sandia National Laboratories. 2020.

[Paul 2017]
Paul, C. L. & Dykstra, J. Understanding Operator Fatigue, Frustration, and Cognitive Workload in Tactical Cybersecurity Operations. *Journal of Information Warfare*. Volume 16. Number 2. Spring 2017.Pages 1-11. https://www.jstor.org/stable/26502752

[Rittel 1973]
Rittel, Horst W. J. & Webber, Melvin. Dilemmas in a General Theory of Planning. *Policy Sciences*. Volume 4. Issue 2. Pages 155-169. 1973. https://link.springer.com/article/10.1007/BF01405730

[Sayak 2023]
Roy, Sayak Saha; Naragam, Krishna Vamsi; & Nilizadeh, Shirin. Generating Phishing Attacks Using ChatGPT. *Cryptography and Security*. https://doi.org/10.48550/arXiv.2305.05133

[Shanteau 1992]
Shanteau, James. Competence in Experts: The Role of Task Characteristics. *Organizational Behavior and Human Decision Processes*. Volume 53. Issue 2. Pages 252–266. November 1992. https://doi.org/10.1016/0749-5978(92)90064-E

[Shimony 2023]
Shimony, Eran & Tsarfati, Omer. Chatting Our Way Into Creating a Polymorphic Malware [blog post]. *CyberArk Blog*. January 17, 2023. https://www.cyberark.com/resources/threat-research-blog/chatting-our-way-into-creating-a-polymorphic-malware

[Sushlia 2023]
Nair, Sushila. Five Ways for Digital Trust Professionals to Improve Soft Skills. *@ISACA.* Volume 46. November 2023. https://www.isaca.org/resources/news-and-trends/newsletters/atisaca/2023/volume-46/five-ways-to-improve-soft-skills

[Wei 2022]
Wei, Jason; Wang, Xuezhi; Schuurmans, Dale; Bosma, Maarten; Ichter, Brian; Xia, Fei; Chi, Ed; Le, Quoc; & Zhou, Denny. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2022. https://arxiv.org/abs/2201.11903

[WormGPT 2024]
WormGPT. FlowGPT Website. February 1, 2024 [accessed]. https://flowgpt.com/p/wormgpt-6

Legal Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM24-0025

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu