Fluent but Hollow: The Operational Limits of LLMs in DNS Security

Derek Wegerbauer
Touro University
Berlin, Germany
dwegerbauer@student.touro.edu

Revital Marbel
School of Computer Science
Holon Institute of Technology (HIT)
Holon, Israel
marbelr@hit.ac.il

Harel Berger
Ariel University, Israel
harelb@ariel.ac.il

Florian Alt
LMU Munich
Munich, Germany
florian.alt@ifi.lmu.de

***Abstract*—Security practitioners increasingly use Large Language Models (LLMs) to extract operational parameters from attack literature for IDS configuration and incident response. However, the reliability of these parameters is poorly understood. We evaluate five state-of-the-art LLMs on sixteen canonical DNS cache-poisoning papers, comparing citation-only and document-grounded prompting via independent human annotation. We identify a systematic semantic-procedural gap: while models excel at narrative attack descriptions, they frequently hallucinate or fail to recover the precise quantitative parameters required for defensive deployment. Our analysis reveals a 'Temporal Erosion' of technical accuracy, characterized by a performance drop of up to 20% when processing historical literature. This creates a 'Silent Failure' mode—producing technically plausible but operationally incorrect parameters for legacy primitives. We demonstrate that while document grounding significantly improves performance, ungrounded LLMs introduce systemic risk by generating 'operationally hollow' defenses. Our findings establish that document grounding is a mandatory but insufficient prerequisite for safety. Even with grounding, persistent stochastic errors in quantitative extraction mean that LLMs cannot yet be deployed in high-assurance security workflows without rigorous human-in-the-loop verification.**

I. Introduction

Developing network attack detection systems traditionally requires manually extracting precise parameters, like packet counts and protocol conditions, from research papers, a slow and error-prone process. Large language models (LLMs) streamline this workflow, allowing defenders to instantly query and extract technical summaries, attack characteristics, and implementation details.

Recent work has demonstrated that LLMs can support a wide range of cybersecurity mitigation tasks, including phishing detection [1], [2], vulnerability detection and automated patch generation [3], [4], automated penetration testing workflows [5], with alternative LLM architectures extending these capabilities to complex malware detection [6]. Comprehensive surveys highlight their rapid adoption across defensive and analytical workflows [7]. These capabilities suggest that LLMs may serve as efficient interfaces to security knowledge, potentially accelerating defensive analysis and reducing the burden of manual literature review. However, their usefulness in operational security contexts depends critically on the correctness of the parameters they produce. Unlike narrative summaries, operational parameters such as packet counts, timing constraints, and affected implementations must be precise to support attack reproduction, intrusion detection system (IDS) configuration, and defensive validation.

Despite their demonstrated utility in security applications, the reliability of LLMs in recovering precise operational parameters from technical literature remains largely unexplored. Prior evaluations [3], [4], [5] have primarily focused on qualitative tasks, where approximate correctness may still provide value. In contrast, defensive workflows often require exact procedural details, and even small inaccuracies may result in ineffective detection rules. This distinction highlights a critical gap between narrative understanding and operational reliability that has not been systematically measured.

To address this gap, we conduct a systematic empirical evaluation of LLM reliability in extracting operational attack parameters from canonical security literature. We focus on DNS cache poisoning attacks as a case study. DNS cache poisoning is a class of protocol-level attacks that require precise packet-level and implementation-specific parameters for successful execution and mitigation [8], [9], [10], [11]. These attacks have been extensively studied over several decades, with well-documented operational characteristics compiled in prior work [12], [13]. This historical corpus provides a structured and authoritative benchmark for evaluating whether LLMs can accurately recover operational security information from technical documents.

We evaluate five state-of-the-art LLMs across sixteen canonical DNS cache poisoning papers using structured prompts designed to extract attack vectors, packet counts, and affected vendors. Our evaluation compares prompting without the actual papers, which reflects realistic workflows in which practitioners query models using paper titles or references, with document-grounded prompting, in which the original paper is provided to the model. Using independent human annotation and statistical analysis, we measure model accuracy, error rates, and abstention behavior across both conditions.

Our results reveal a consistent discrepancy between narrative competence and operational reliability. While models accurately summarize attack concepts and identify affected systems, they frequently produce incorrect, incomplete, or unstable quantitative parameters when operating without explicit document

grounding. We characterize this discrepancy as a *semantic–procedural gap*, in which LLMs demonstrate strong narrative understanding but unreliable recovery of procedural details required for operational security tasks. we characterize the quantitative collapse in historical accuracy as a 'Silent Failure.' Unlike a model refusal, which prompts human intervention, these fluent hallucinations can lead a security practitioner to misconfigure IDS thresholds or rate-limiting rules, resulting in a defense that is syntactically valid but operationally hollow. These findings highlight a fundamental limitation in using LLMs as autonomous sources of operational security knowledge and underscore the importance of document grounding and human verification when integrating LLMs into defensive workflows. We make the following contributions:

- **Systematic measurement of LLM operational reliability in security workflows.** We evaluate five state-of-the-art LLMs on their ability to extract operational attack parameters from sixteen canonical DNS cache poisoning papers spanning three decades, using structured prompts and independent human annotation.
- **Identification of the 'Temporal Erosion' of security knowledge.** We demonstrate a systematic *semantic–procedural gap* where LLMs excel at narrative understanding but fail at quantitative recovery. We identify a 'temporal erosion' effect, where extraction accuracy for foundational attack primitives (pre-2020) is significantly lower than for emerging threats. We argue this creates a blind spot, as historical primitives are frequently reused in modern attacks, as formalized in recent work [12].
- **Empirical demonstration of grounding dependence for operational correctness.** We show that providing source documents significantly improves extraction accuracy, establishing document grounding as a functional requirement. However, we deduce that grounding primarily improves narrative recall rather than procedural reasoning, as quantitative accuracy remains significantly below the thresholds required for automated defensive configuration.
- **Characterization of the 'Silent Failure' mode.** We identify a high-risk failure mode where models produce 'fluent hallucinations' - narratively correct summaries paired with incorrect quantitative parameters. We deduce that using these parameters leads to 'operationally hollow' IDS and firewall configurations, providing a false sense of security in automated defensive pipelines.

II. Background & Related Work

A. DNS Resolution and Cache Poisoning

When a recursive resolver R does not possess a cached resource record for a queried domain D, R issues iterative queries up the DNS hierarchy until it contacts the authoritative name server A for D. Server A responds with the requested address record; R accepts and caches that response only after validating the ephemeral parameters of the packet: the domain name D, the source IP, the ephemeral source port p, and the transaction identifier t of the packet. An adversary M who queries R to perform a DNS lookup can attempt to inject a forged response that impersonates A's response. The resolver will add the malicious record to its cache only if it is validated, as explained above. Once stored, the malicious record causes all subsequent client queries for D to be resolved to the malicious IP address.

Practical types of DNS cache poisoning attacks reduce the need to correctly guess both p and t. A statistical brute-force type transmits numerous forged responses containing unique (p', t') combinations until the right combination is achieved. Fragmentation-based types exploit IP fragmentation by inserting a malicious fragment that will be reassembled with a legitimate fragment; because the UDP header appears only in the initial fragment, the attacker needs only match the 16-bit IP fragment identifier IPID rather than the p and t parameters. Separately, resolvers that indiscriminately cache out-of-bailiwick resource records permit an attacker to include additional records for domains outside the queried zone, producing cached mappings for entirely different domains supplied within the same forged response.

B. Large Language Models

LLMs are transformer-based models trained on large text corpora and increasingly used as interfaces to technical knowledge [14]. Their usefulness in security workflows depends not only on fluent summarization, but also on reliable recovery of exact operational details. Two failure modes are especially relevant here: hallucinations, where models produce plausible but incorrect claims [15], and refusals, where models decline to answer due to alignment constraints or safety mechanisms [16], [17], [18]. In high-assurance security settings, both behaviors matter: hallucinations can produce unsafe configurations, while refusals limit analyst utility. We evaluate five state-of-the-art models: ChatGPT-4o, ChatGPT-5, Claude Sonnet 4.5, Gemini 2.5 Pro, and DeepSeek V3.1.

C. LLMs for Security

Previous research has demonstrated the potential of LLMs in various security tasks, including phishing and vulnerability detection, automated patching, and penetration testing [19], [3], [4]; however, their reliability in extracting the precise operational parameters required for defensive high-assurance workflows remains largely unexplored. Within network security, previous work has targeted human-centered threats and protocol fuzzing. In the phishing domain, KnowPhish [1] and Ji et al. [2] show that LLMs can detect notorious campaigns, leverage contextual and visual cues, and generalize to varied contexts. For protocol assurance, Meng et al. [20] and Ma et al. [21] use LLMs to automate input generation and harness semantic feedback to guide network-protocol fuzzing, improving path coverage and bug discovery.

Overall, prior work establishes that LLMs can support a wide range of security tasks, including vulnerability detection, penetration testing, phishing detection, and protocol analysis. However, existing evaluations have focused primarily on qualitative capabilities such as classification, summarization, and

exploit generation. Far less is known about whether LLMs can reliably recover precise operational parameters required for defensive workflows, such as packet counts, protocol conditions, and affected targets. This distinction is critical, as operational parameters must be exact to support attack reproduction, intrusion detection configuration, and defensive validation.

To address this gap, we conduct a systematic empirical evaluation of LLM reliability in extracting operational attack parameters from canonical security literature. We use DNS cache poisoning as a case study, leveraging its well-documented attack corpus and explicitly specified operational characteristics. This setting enables us to quantify model accuracy, characterize failure modes, and identify the semantic–procedural gap between narrative understanding and operational parameter recovery.

III. Methodology

Research Question. Our study asks: *How accurately can LLMs recover key operational parameters of DNS cache-poisoning attacks from research literature, with and without access to the original paper?*

We focus on attacks targeting recursive resolvers[1], using the POPS corpus [12] (see Appendix B)[2], which catalogs the top 18 resolver-centric attacks most attractive to adversaries [24], [25]. Papers were included when they described a concrete DNS cache-poisoning technique, targeted recursive resolvers or resolver-adjacent behavior, and reported sufficient operational detail to derive the attack vector, malicious packet count, or affected vendors. Papers were excluded when they focused primarily on unrelated DNS topics, such as tunneling, denial-of-service, DNSSEC deployment, or general DNS measurement, unless they contained a clearly defined cache-poisoning mechanism. We evaluate state-of-the-art LLMs through prompts designed to test three key capabilities: (1) understanding the end-to-end flow of each attack; (2) estimating the packet volume required for a successful attack; and (3) articulating the expected impact on DNS vendors, thereby assessing whether models accurately recover literature-reported operational parameters of DNS cache poisoning attacks. Human annotators with expertise in DNS infrastructure review model outputs and score them along these dimensions. We next describe the prompt design, annotation schema, evaluation metrics, and statistical tests used to quantify model improvements.

A. Prompts

We began with context-less zero-shot prompts [26], [27] to assess each model's inherent knowledge of DNS cache-poisoning attack classes and mechanics [4]. The evaluation was then repeated with prompts augmented by the full paper text, enabling models to ground their answers in both prior knowledge and the provided document[3]. Below, we present examples of prompts for a single attack paper: Prompt A (zero-shot) omits the paper; Prompt B includes the paper.

Prompt A (W/O Paper)

Model: GPT-4o
User Prompt:

```
Following your knowledge on the paper "The maginot
line: Attacking the  boundary of DNS caching
protection", please answer the following questions:
(1) What is the attack vector?
(2) How many packets were used to run the attack?
(3) Which vendors were affected by the attack?
```

Prompt B (With Paper)

Model: GPT-4o
User Prompt:

```
Following your knowledge on the paper "The maginot
line: Attacking the boundary of DNS caching
protection", and the attack paper,
please answer the following questions:
(1) What is the attack vector?
(2) How many packets were used to run the attack?
(3) Which vendors were affected by the attack?
```

Prompt A uses the paper title as an identifier, reflecting a realistic workflow where practitioners query an assistant using a citation string (e.g., from a bibliography or memory) without immediate access to the PDF. This design tests whether models can answer *paper-specific factual questions* from prior exposure or generalization. We treat this as distinct from open-ended inquiries on DNS, and interpret citation-only results accordingly.

B. Annotations

After collecting the model outputs, two independent annotators (an undergraduate and a faculty member experienced in DNS security research) evaluated each response against the three capabilities. Disagreements were resolved by discussion. Ground truth labels are defined using the per-paper fact sheets provided in an anonymous artifact repository released alongside this submission. For each paper, the fact sheet records the attack vector, packet count (or formula), and affected DNS vendors. These elements correspond to operational characteristics reported in the original papers.

- **Attack Vector** The annotator issued a binary judgment (match/no match) on whether the response correctly identified the attack vector described in the paper. Semantically equivalent descriptions were accepted as correct.
- **Attack Packets** The annotator checked whether the response included the correct packet count to execute the attack, accepting exact numbers or paper, consistent ranges. This gauges if the model identifies parameters relevant to `PCAP` construction for IDS or manual analysis.

[1]Attacks on DNS clients and forwarders usually affect only one endpoint [22], [23]; hardening DNS resolvers shields all users that rely on them.

[2]Table III lists 18 attacks drawn from 16 papers.

[3]Per vendors' published privacy policies, these models do not train on user prompts or auxiliary materials; thus, supplying the paper does not raise ethical concerns.

- **Affected Vendors** The annotator evaluated whether the response correctly identified which DNS vendors were impacted. Responses were deemed correct if they matched any vendor from the list in the paper.

Responses that provided no relevant information or declined to answer were recorded separately as *no-answer*. For accuracy calculations, such cases were counted as incorrect, but they were also tracked to capture abstention behavior.

C. Metrics

Performance of the LLMs was compared under two conditions: prompts without access to the paper (*text-only*) and prompts where the full paper PDF was attached (*attachment*). We evaluated the performance using three metrics for each model, condition, and target capability: accuracy, wrong-response rate, and no-answer rate. Accuracy was defined as the proportion of correct responses over the total number of responses, with no-answer responses counted as incorrect. The wrong-response rate measured the proportion of incorrect responses among attempts, reflecting cases where the model produced an answer, but it was wrong (i.e., hallucinations). No-answer rate captured the proportion of abstentions, reflecting cases where the model refused to answer. Reporting both error types separately made it possible to distinguish failures caused by hallucinations from those caused by refusals.

Statistical Analysis. To evaluate the impact of document grounding, we employed a within-subject paired design comparing *text-only* and *attachment* conditions across all model-item pairs ($n = 16$). We tested the null hypothesis ($H_0 : \mu_{\text{with}} = \mu_{\text{without}}$) using a paired t-test ($\alpha = 0.05$). To ensure robustness against non-normal distributions, we supplemented this with a Wilcoxon signed-rank test. The magnitude of the effect was quantified using Cohen's d, where values of 0.2, 0.5, and 0.8 represent small, medium, and large effects, respectively.

Historical Analysis To account for the evolving landscape of LLM capabilities and the dynamic nature of technical discourse [28], we stratify the research corpus into pre-2020 and post-2020 cohorts. This division stems from the architectural shift initiated by the release of GPT-3 [29], which marked the transition toward generalized in-context learning. By isolating literature from the 'pre-foundation model' era, we can evaluate whether LLMs maintain operational rigor when processing historical technical frameworks or if they exhibit a 'temporal affinity' for contemporary research.

IV. Experiments

In this section, we describe the experimental design. We evaluated five LLMs on extracting key facts from DNS cache poisoning research papers. The task focused on identifying three target items from each paper: (i) attack vector, (ii) number of attack packets, and (iii) vendors affected by the attack.

A. Experimental Setup

The initial plan was to use API calls for automation and systematic logging. However, inconsistent API availability and differences in attachment handling across models would have created unfair advantages. To ensure comparability, all experiments were conducted via the free web-based interfaces provided by each model developer. For transparency and replication, we additionally will release all prompts, raw outputs, and annotations in a repository. However, evaluation criteria and ground-truth labels are fully specified in this paper.

We used each vendor's public web UI to upload the PDF prior to prompting. This reflects a practitioner-facing workflow, but it is less controlled than an API benchmark because web interfaces may include hidden prompts, changing model configurations, and proprietary document-processing pipelines. Thus, the attachment condition means "paper provided through the UI," not guaranteed identical PDF parsing across models. We treat this as a threat to validity and do not claim to isolate model weights from interface-level effects.

1) Evaluated Models: Five models were included in the study: ChatGPT-4o, ChatGPT-5-o1 (ChatGPT-5), Claude Sonnet 4.5 (Claude), Gemini 2.5 Pro (Gemini), and DeepSeek V3.1 (DeepSeek). These models were chosen as they are top-shelf models and are currently accessible through public web interfaces. We used the default parameters (e.g., temperature 1.0, Top-P 1.0) for all models.

2) Paper Collection: We evaluated 16 DNS cache-poisoning papers under *text-only* and *attachment* conditions.

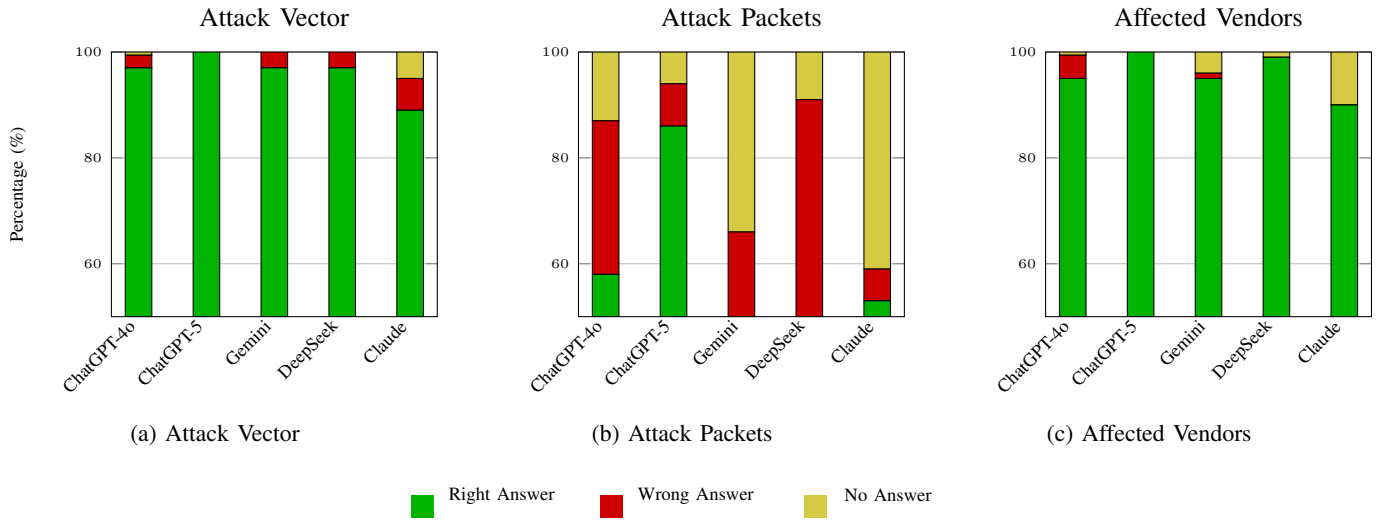
3) Response Gathering: For every paper and condition, five independent responses were collected from each model in separate sessions to capture nondeterminism in outputs. This yielded 160 responses per model: 16 papers $\times$ 2 prompting conditions $\times$ 5 repetitions. Across the five evaluated models, the study produced 800 total responses. Each response was later annotated along three target fields: attack vector, malicious packet count, and affected vendors. Responses were copied directly from the web interface into plain-text logs for later analysis. Prompts were kept identical across all models and conditions, with the only modification being the reference to the uploaded PDF in the attachment condition.

V. Results

We evaluated 800 model responses across three target fields: attack vector, malicious packet count, and vendor identification. A response denotes one model output for a given paper, condition, and repetition; a field-level judgment denotes the annotation of one target field within that response. Outcomes were classified as correct, incorrect, or no answer. Table I and Figure 1 summarize performance across models and fields..

1) Attack Vector: All models demonstrated high accuracy on the *Attack Vector* element. ChatGPT–5 achieved perfect accuracy (100%), followed by ChatGPT–4o, Gemini, and DeepSeek (each 97%). Claude reached 89%, with 6% incorrect and 5% unanswered responses. No other model exceeded 1% in no-answer responses.

2) Number of Attack Packets: Performance varied strongly across models in the *Attack Packets* element. ChatGPT–5 attained the highest accuracy (86%), while ChatGPT–4o, Gemini, DeepSeek, and Claude achieved 58%, 49%, 46%, and 54%, respectively. DeepSeek showed the highest rate of incorrect



(a) Attack Vector (b) Attack Packets (c) Affected Vendors

Fig. 1: Comparative evaluation results of model refusal (no answer) and wrong answers across three categories: Attack Vector, Attack Packets, and Affected Vendors. Green indicates correct identification of the number of attack packets, red denotes incorrect description, and yellow represents unanswered cases.

TABLE I: Model Performance (Accuracy ± SD) and Contextual Improvement (Δ) providing the perspective paper

Model	**Attack Vector**		**Attack Packets**		**Attacked Vendors**	
	Accuracy	Δ	**Accuracy.**	Δ	**Accuracy.**	Δ
ChatGPT-5	100%	0.0	86.2%	+11.3	100%	0.0
ChatGPT-4o	96.8%	+6.3	58.1%	+62.5	96.8%	+1.3
DeepSeek	96.8%	0.0	45.6%	+81.3	98.7%	+3.8
Gemini	96.8%	0.0	48.7%	+46.3	95.6%	+8.8
Claude	91.6%	+10.0	54.0%	+42.5	89.6%	+18.8
Mean	–	**+3.3**	–	**+48.7**	–	**+6.5**

answers (45%), while ChatGPT–5 maintained the lowest combined error and refusal rate (14%).

3) Affected Vendors: In the *Affected Vendors* element, ChatGPT–5 and DeepSeek achieved near-perfect accuracy (100% and 99%), followed by ChatGPT–4o and Gemini (both 96%) and Claude (90%). Claude exhibited a 10% no-answer rate, the highest among models in this category, while others remained below 1% no-answer and 4% incorrect outputs.

4) No Answer: Refusal behavior (*"No Answer"* responses) differed notably between models. As shown in Figure 1, Claude displayed the highest refusal frequency overall, reaching 41% in the *Attack Packets* task and 10% in *Affected Vendors*. Gemini followed with 34% refusals in *Attack Packets*. DeepSeek and ChatGPT–4o exhibited lower refusal rates (typically below 10%), while ChatGPT–5 showed the lowest refusal rate across all tasks ($\leq$6%).

A paired t-test showed a significant improvement across all fifteen model–element pairs, $t(14) = 2.91$, $p = 0.0115$. The corresponding Cohen's $d = 0.75$ indicates a medium-to-large effect size. A Wilcoxon signed-rank test yielded a consistent result ($p = 0.0033$). The statistical significance across these tests allows us to reject the null hypothesis (H_0), confirming that providing the original research papers leads to a non-zero, positive improvement in the models' extraction accuracy.

Across all models and categories, accuracy increased on average by $19.5 \pm 26.0\%$. The largest relative gain occurred for *Attack Packets* (+48.7%), followed by smaller increases in *Attack Vector* (+3.3%) and *Affected Vendors* (+6.5%). DeepSeek and ChatGPT–4o showed the largest contextual improvements, while ChatGPT–5 exhibited minimal change due to its already high baseline accuracy. This does not imply that grounding is unnecessary for reasoning-heavy models; rather, we think that it may suggest that ChatGPT–5's baseline performance is constrained by a more conservative refusal strategy and a different internal representation of technical data. Despite its stability, ChatGPT–5 still struggled with the 'Semantic-Procedural Gap,' failing to achieve the near-100% precision seen in its narrative tasks.

5) Timeline Effect: Table II provides a comparative analysis of model performance across two distinct eras: Historical Papers (published between 1993–2020) and Emerging Threats (published between 2020–2024). Most models demonstrated higher accuracy on emerging threats (2020–2024) compared

TABLE II: Temporal Stratification of LLM Performance (%): Historical Papers (Pre-2020) vs. Emerging Threats (2020-2024)

Model	**Attack Vector**	**Malicious Packets**	**Vendors**
Emerging Threats (2020–2024)			
Claude	90.0 ± 30.0	56.7 ± 45.4	90.0 ± 30.0
ChatGPT-5	100.0 ± 0.0	90.0 ± 30.0	100.0 ± 0.0
ChatGPT-4o	100.0 ± 0.0	63.3 ± 45.4	100.0 ± 0.0
Gemini	90.0 ± 30.0	56.7 ± 45.4	90.0 ± 30.0
DeepSeek	100.0 ± 0.0	40.0 ± 47.1	90.0 ± 30.0
Historical Papers (Pre-2020)			
Claude	82.7 ± 33.3	36.0 ± 44.6	80.0 ± 36.6
ChatGPT-5	100.0 ± 0.0	85.0 ± 35.8	100.0 ± 0.0
ChatGPT-4o	96.7 ± 18.3	53.3 ± 46.2	98.3 ± 8.2
Gemini	82.7 ± 33.3	42.0 ± 45.4	80.0 ± 36.6
DeepSeek	93.3 ± 23.6	40.0 ± 47.1	93.3 ± 23.6

to historical papers when provided with the full text. For example, Claude and Gemini both show an approximate 8% drop in Attack Vector identification when moving from modern to historical literature. This performance gap may suggest that while LLMs are trained on massive corpora that include historical security archives, the precise operational parameters of foundational attacks (e.g., pre-2000s DNS attacks) appear to be less probable to have correct resolution than modern threats.

This suggests a 'Temporal Erosion' of technical accuracy. As the 'semantic distance' from the training cutoff increases, models struggle to reconcile legacy technical terminology with modern security taxonomies. For instance, Claude's accuracy for packet counts drops from 56.7% on emerging threats to a mere 36% on historical papers, a significant relative decrease that highlights the fragility of LLM memory regarding foundational technical specifications.

Across both timelines, models maintained high accuracy (often reaching 100%) and zero or low standard deviation (SD) in semantic tasks such as identifying the Attack Vector and Attacked Vendors. However, the extraction of Malicious Packets remains highly stochastic, with standard deviations reaching as high as 47.14% for models like DeepSeek in both eras. While models like ChatGPT-4o and DeepSeek are almost perfect ($93.3\% - 100\%$) at naming the attacked vendors in historical papers, their accuracy in extracting packet counts for those same papers hovers around 40%-53%.

From an intrusion detection perspective, this represents a 'Silent Failure' mode. An LLM might correctly identify the 'narrative' of a legacy attack vector (e.g., a birthday attack) but provide a packet-rate parameter that is off by an order of magnitude. One might argue that the operational details of decades-old attacks are of diminishing relevance; however, recent research [12] demonstrates that DNS vulnerabilities exhibit significant longitudinal continuity. Modern 'emerging' threats often rely on fundamental primitives first articulated in the early 2000s, meaning they share similar descriptive and applicable parameters.

Consequently, the Semantic-Procedural Gap doesn't just obscure the past; it creates a blind spot for the future. If a security engineer uses these hallucinated parameters to configure a Rate-Limiting IDS rule or a Honeypot threshold, the resulting defense would be operationally useless despite the 'fluent' summary appearing correct. This proves that reliable parameter extraction is not just a linguistic challenge, but a critical requirement for automated security reasoning.

VI. Discussion

Our findings show that document grounding improves extraction accuracy but does not eliminate the semantic–procedural gap. Models often summarize attack intent correctly while failing to preserve the quantitative relationships needed for defensive configuration. This leads to five lessons.

1) Lesson 1: Narrative Fluency Is Not Operational Precision: High accuracy on attack vectors and vendors can create a halo effect: users may trust quantitative outputs because the surrounding explanation sounds correct. Packet-count variance shows that this trust is unsafe.

2) Lesson 2: Historical Primitives Remain Operationally Relevant: The preliminary Temporal Erosion trend is important because DNS attacks exhibit longitudinal continuity. Weakness on older papers can therefore affect analysis of modern attacks that reuse the same primitives.

3) Lesson 3: Grounding Is Necessary but Insufficient: Providing the paper improves recall, but models still confuse packet counts, probability bounds, queries, and forged responses. Reliable use in security workflows requires validation against source text and protocol constraints.

4) Lesson 4: Qualitative Patterns of Technical Failure: Manual inspection showed three recurring failures: substituting generic DNS values for paper-specific counts, confusing related quantities such as attempts and responses, and collapsing formulas or conditional parameters into unsupported single numbers. These patterns suggest that errors reflect weak procedural tracking rather than random noise alone.

5) Lesson 5: Silent Failure Is Riskier Than Refusal: Fluent hallucinations are more hazardous than refusals because they can produce syntactically valid but operationally hollow IDS or firewall configurations. For example, using a generic value instead of the specific "200 per query" parameter for Kaminsky-style attacks can yield an ineffective detection threshold.

6) Applied Implications: LLMs are useful for documentation support and attack summarization, but not for autonomous defensive configuration. Operational use should combine LLM outputs with structured extraction, deterministic checks, domain constraints, and human verification.

7) Limitations: This study has four main limitations. First, public web interfaces introduce hidden prompts, changing configurations, and document-processing variability. Second, citation-only results may reflect memorization rather than reasoning. Third, zero-shot prompts may understate performance achievable with iterative prompting. Fourth, the corpus and annotation process are intentionally focused on DNS cache poisoning, which improves depth but limits generality.

VII. Conclusion

We evaluated five leading LLMs on their ability to recover operational parameters from DNS cache-poisoning literature. The results show a consistent semantic–procedural gap: models often describe attacks correctly, but struggle to recover precise packet counts and other quantitative details needed for defensive configuration. Document grounding significantly improves accuracy, yet does not eliminate stochastic extraction errors. We therefore conclude that grounding is necessary but insufficient for high-assurance security workflows. LLM outputs should be treated as candidate interpretations requiring source-level and human verification. Future work will extend the evaluation to DNS tunneling and denial-of-service attacks and examine controlled API-based pipelines with structured extraction and validation.

Paper	Year	#Pkts
Schuba et al. [30]	1993	1
V. Sacramento [31]	2002	2^{16}
Klein, Amit [32]	2007	>100
Kaminsky, Dan [8]	2008	$200 \cdot q$
Herzberg et al. [9] (1)	2012	2^{16}
Herzberg et al. [9] (2)	2012	2^{16}
Herzberg et al. [33]	2013	1
Herzberg et al. [34]	2013	$\sim 2^{11}$
Herzberg et al. [35]	2013	2^{16}
Zheng et al. [36]	2020	1
Man et al. [37]	2020	2^{16}
Dai et al. [38]	2021	64
Klein et al. [39]	2021	2^{16}
Jeitner et al. [40] (1)	2022	2^{16}
Jeitner et al. [40] (2)	2022	2^{16}
Li et al. [10]	2023	2^{16}
Heftring et al. [41]	2023	2^{16}
Li et al. [11]	2024	2^{16}

TABLE III: **Corpus overview** Operational parameters as reported in POPS [12], along with the number of packets required for a successful execution reported by the original papers ($\#Pkts$). In the case of Kaminsky's attack, q denotes the number of forged packets sent per forged subdomain query.

APPENDIX A
ACKNOWLEDGMENTS

This research has been funded by the Israeli Ministry of Innovation, Science, and Technology.

APPENDIX B
PAPER CORPUS

In this appendix, we provide the paper corpus, derived from [12].

REFERENCES

[1] Y. Li, C. Huang, S. Deng, M. L. Lock, T. Cao, N. Oo, H. W. Lim, and B. Hooi, "{KnowPhish}: Large language models meet multimodal knowledge graphs for enhancing {Reference-Based} phishing detection," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 793–810.

[2] F. Ji, K. Lee, H. Koo, W. You, E. Choo, H. Kim, and D. Kim, "Evaluating the effectiveness and robustness of visual similarity-based phishing detection models," in *34th USENIX Security Symposium (USENIX Security 25)*, 2025, pp. 3201–3220.

[3] Y. Nong, H. Yang, L. Cheng, H. Hu, and H. Cai, "{APPATCH}: Automated adaptive prompting large language models for {Real-World} software vulnerability patching," in *34th USENIX Security Symposium (USENIX Security 25)*, 2025, pp. 4481–4500.

[4] J. Lin and D. Mohaisen, "From large to mammoth: A comparative evaluation of large language models in vulnerability detection," in *Proceedings of the 2025 Network and Distributed System Security Symposium (NDSS)*, 2025.

[5] G. Deng, Y. Liu, V. Mayoral-Vilches, P. Liu, Y. Li, Y. Xu, T. Zhang, Y. Liu, M. Pinzger, and S. Rass, "{PentestGPT}: Evaluating and harnessing large language models for automated penetration testing," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 847–864.

[6] L. Kurlandski, H. Berger, Y. Pan, and M. Wright, "Beyond raw bytes: Towards large malware language models," 2026.

[7] H. Xu, S. Wang, N. Li, K. Wang, Y. Zhao, K. Chen, T. Yu, Y. Liu, and H. Wang, "Large language models for cyber security: A systematic literature review," 2025. [Online]. Available: https://arxiv.org/abs/2405.04760

[8] D. Kaminsky, "Black ops 2008: It's the end of the cache as we know it," *Black Hat USA*, vol. 2, 2008.

[9] A. Herzberg and H. Shulman, "Security of patched DNS," in *Computer Security–ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings 17*. Springer, 2012, pp. 271–288.

[10] X. Li, C. Lu, B. Liu, Q. Zhang, Z. Li, H. Duan, and Q. Li, "The maginot line: Attacking the boundary of {DNS} caching protection," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 3153–3170.

[11] X. Li, W. Xu, B. Liu, M. Zhang, Z. Li, J. Zhang, D. Chang, X. Zheng, C. Wang, J. Chen *et al.*, "Tudoor attack: Systematically exploring and exploiting logic vulnerabilities in DNS response pre-processing with malformed packets," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2023, pp. 46–46.

[12] Y. Afek, H. Berger, and A. Bremler-Barr, "Pops: From history to mitigation of dns cache poisoning attacks," in *Proceedings of the 34th USENIX Security Symposium*. Seattle, WA, USA: USENIX Association, 2025, pp. 3537–3550. [Online]. Available: https://www.usenix.org/system/files/usenixsecurity25-afek.pdf

[13] H. Berger, A. Z. Dvir, and M. Geva, "A wrinkle in time: a case study in DNS poisoning," *International Journal of Information Security*, vol. 20, pp. 313–329, 2021.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM computing surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[16] H. Berger, H. King, and O. David, "Dreaming with chatgpt: Unraveling the challenges of llms dream generation," in *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, 2024, pp. 140–147.

[17] L. Cao, "Learn to refuse: Making large language models more controllable and reliable through knowledge scope limitation and refusal mechanism," *arXiv preprint arXiv:2311.01041*, 2023.

[18] S. Pasch, "Llm content moderation and user satisfaction: Evidence from response refusals in chatbot arena," *arXiv preprint arXiv:2501.03266*, 2025.

[19] Y. Zou, M. An, and D. Wang, "Password guessing using large language models," in *34th USENIX Security Symposium (USENIX Security 25)*, 2025, pp. 7799–7818.

[20] R. Meng, M. Mirchev, M. Böhme, and A. Roychoudhury, "Large language model guided protocol fuzzing," in *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*, vol. 2024, 2024.

[21] X. Ma, L. Luo, and Q. Zeng, "From one thousand pages of specification to unveiling hidden bugs: Large language model assisted fuzzing of matter {IoT} devices," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 4783–4800.

[22] F. Alharbi, J. Chang, Y. Zhou, F. Qian, Z. Qian, and N. Abu-Ghazaleh, "Collaborative client-side DNS cache poisoning attack," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1153–1161.

[23] T. Ma, C. Xu, Z. Zhou, X. Kuang, L. Zhong, and L. A. Grieco, "Intelligent-driven adapting defense against the client-side DNS cache poisoning in the cloud," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[24] J. Park, R. Jang, M. Mohaisen, and D. Mohaisen, "A large-scale behavioral analysis of the open dns resolvers on the internet," *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, pp. 76–89, 2021.

[25] R. Yazdani, R. van Rijswijk-Deij, M. Jonker, and A. Sperotto, "A matter of degree: characterizing the amplification power of open dns resolvers," in *International Conference on Passive and Active Network Measurement*. Springer, 2022, pp. 293–318.

[26] Y. Wan and K.-W. Chang, "White men lead, black women help: Uncovering gender, racial, and intersectional bias in language agency," *arXiv preprint arXiv:2404.10508*, 2024.

[27] Y. Wan, G. Pu, J. Sun, A. Garimella, K.-W. Chang, and N. Peng, "" kelly is a warm person, joseph is a role model": Gender biases in llm-generated reference letters," *arXiv preprint arXiv:2310.09219*, 2023.

[28] C. ChenghaoZhu, N. Chen, Y. Gao, Y. Zhang, P. Tiwari, and B. Wang, "Is your llm outdated? a deep look at temporal generalization," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025, pp. 7433–7457.

[29] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
[30] C. Schuba and E. H. Spafford, "Addressing weaknesses in the domain name system protocol," *Master's thesis, Purdue University*, 1993.
[31] V. Sacramento, "Vulnerability in the sending requests control of bind versions 4 and 8 allows DNS spoofing," 2002.
[32] A. Klein, "Bind 9 DNS cache poisoning," *Report, Trusteer, Ltd*, vol. 3, 2007.
[33] A. Herzberg and H. Shulman, "Fragmentation considered poisonous, or: One-domain-to-rule-them-all. org," in *2013 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2013, pp. 224–232.
[34] A. Herzberg and H. Shulman, "Vulnerable delegation of DNS resolution," in *Computer Security–ESORICS 2013: 18th European Symposium on Research in Computer Security, Egham, UK, September 9-13, 2013. Proceedings 18*. Springer, 2013, pp. 219–236.
[35] A. Herzberg and H. Shulman, "Socket overloading for fun and cache-poisoning," in *Proceedings of the 29th Annual Computer Security Applications Conference*, 2013, pp. 189–198.
[36] X. Zheng, C. Lu, J. Peng, Q. Yang, D. Zhou, B. Liu, K. Man, S. Hao, H. Duan, and Z. Qian, "Poison over troubled forwarders: A cache poisoning attack targeting {DNS} forwarding devices," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 577–593.
[37] K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang, and H. Duan, "DNS cache poisoning attack reloaded: Revolutions with side channels," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1337–1350.
[38] T. Dai, H. Shulman, and M. Waidner, "DNS-over-TCP considered vulnerable," in *Proceedings of the Applied Networking Research Workshop*, 2021, pp. 76–81.
[39] A. Klein, "Cross layer attacks and how to use them (for DNS cache poisoning, device tracking and more)," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 1179–1196.
[40] P. Jeitner, H. Shulman, L. Teichmann, and M. Waidner, "{XDRI} attacks-and-how to enhance resilience of residential routers," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 4473–4490.
[41] E. Heftrig, H. Shulman, and M. Waidner, "Poster: Off-path DNSSEC downgrade attacks," in *Proceedings of the ACM SIGCOMM 2023 Conference*, 2023, pp. 1120–1122.