

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ORCA SECURITY LTD,
Petitioner

v.

WIZ, INC.,
Patent Owner

Inter Partes Review Case No. IPR2025-01086
U.S. Patent No. 12,001,549

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 12,001,549**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	SUMMARY OF THE '549 PATENT	2
A.	DESCRIPTION OF THE '549 PATENT	2
B.	SUMMARY OF THE PROSECUTION HISTORY	3
C.	LEVEL OF SKILL OF A PERSON HAVING ORDINARY SKILL IN THE ART	3
III.	REQUIREMENTS UNDER 37 C.F.R. § 42.104	4
A.	GROUND FOR STANDING UNDER 37 C.F.R. § 42.104(A)	4
B.	IDENTIFICATION OF CHALLENGE UNDER 37 C.F.R. § 42.104(B)	4
C.	CLAIM CONSTRUCTION UNDER 37 C.F.R. § 42.104(B)(3)	5
IV.	SHOWING OF ANALOGOUS, PRIOR ART	6
A.	U.S. PATENT NO. 11,303,666 ("PETERS") (EX. 1004)	6
B.	U.S. PATENT PUBLICATION NO. 2024/0403428 ("LAL PATENT PUBLICATION") (EX. 1005) AND U.S. PROVISIONAL PATENT APPLICATION NO. 63/472,227 ("LAL") (EX. 1006)	10
C.	A POSITA WOULD HAVE BEEN MOTIVATED TO COMBINE PETERS AND LAL	12
V.	GROUND 1: CLAIMS 1-5 AND 11-16 ARE OBVIOUS OVER PETERS AND LAL	18
A.	CLAIM 1	18
1.	<i>Claim 1(Pre): "A method for providing cybersecurity incident response, comprising:"</i>	18
2.	<i>Claim 1(a): "receiving an incident input based on a cybersecurity event"</i>	18
3.	<i>Claim 1(b): "generating a prompt for a large language model (LLM) based on the received incident input"</i>	20
4.	<i>Claim 1(c): "configuring the LLM to generate an output based on the generated prompt"</i>	23
5.	<i>Claim 1(d): "mapping the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence response"</i>	25
6.	<i>Claim 1(e): "generating a query based on the received incident input and the mapped scenario"</i>	28

7.	<i>Claim 1(f): “executing the query on a security database, the security database including a representation of a computing environment”</i>	30
8.	<i>Claim 1(g): “initiating a mitigation action based on a result of the executed query.”</i>	33
B.	CLAIM 2	35
1.	<i>Claim 2: “The method of claim 1, wherein the incident input includes any one of: a query, a statement, and a combination thereof.”</i>	35
C.	CLAIM 3	37
1.	<i>Claim 3: “The method of claim 1, wherein the LLM is trained on any one of: a data schema utilized in representing the computing environment, incident data classified to a scenario, the plurality of scenarios, and a combination thereof.”</i>	37
D.	CLAIM 4	39
1.	<i>Claim 4: “The method of claim 3, wherein generating the query further comprises: generating a second prompt for the LLM which when executed by the LLM outputs the query, wherein the second prompt is generated based on any one of: the received incident input, the data schema, the plurality of scenarios, and a combination thereof.”</i>	39
E.	CLAIM 5	43
1.	<i>Claim 5: “The method of claim 3, further comprising: training the LLM further on a plurality of database queries, each database query executable on the security database.”</i>	43
F.	CLAIM 11	44
1.	<i>Claim 11(Pre): “A non-transitory computer-readable medium storing a set of instructions for providing cybersecurity incident response, the set of instructions comprising: one or more instructions that, when executed by one or more processors of a device, cause the device to:”</i>	44
2.	<i>Claim 11(a): “receive an incident input based on a cybersecurity event;”</i>	46
3.	<i>Claim 11(b): “generate a prompt for a large language model (LLM) based on the received incident input;”</i>	46
4.	<i>Claim 11(c): “configure the LLM to generate an output based on a cybersecurity event;”</i>	46

5.	<i>Claim 11(d): “map the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence response;”</i>	46
6.	<i>Claim 11(e): “generate a query based on the received incident input and the mapped scenario;”</i>	46
7.	<i>Claim 11(f): “execute the query on a security database, the security database including a representation of a computing environment; and”</i>	46
8.	<i>Claim 11(g): “initiate a mitigation action based on a result of the executed query.”</i>	47
G.	CLAIM 12	47
1.	<i>Claim 12(Pre): “A system for providing cybersecurity incident response comprising:”</i>	47
2.	<i>Claim 12(a): “a processing circuitry;”</i>	47
3.	<i>Claim 12(b): “a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:”</i>	48
4.	<i>Claim 12(c): “receive an incident input based on a cybersecurity event;”</i>	49
5.	<i>Claim 12(d): “generate a prompt for a large language model (LLM) based on the received incident input;”</i>	49
6.	<i>Claim 12(e): “configure the LLM to generate an output based on a cybersecurity event;”</i>	49
7.	<i>Claim 12(f): “map the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence response;”</i>	49
8.	<i>Claim 12(g): “generate a query based on the received incident input and the mapped scenario;”</i>	49
9.	<i>Claim 12(h): “execute the query on a security database, the security database including a representation of a computing environment; and”</i>	49
10.	<i>Claim 12(i): “initiate a mitigation action based on a result of the executed query.”</i>	49
H.	CLAIM 13	50
1.	<i>Claim 13: “The system of claim 12, wherein the incident input includes any one of: a query, a statement, and a combination thereof.”</i>	50
I.	CLAIM 14	50

1.	<i>Claim 14: “The system of claim 12, wherein the LLM is trained on any one of: a data schema utilized in representing the computing environment, incident data classified to a scenario, the plurality of scenarios, and a combination thereof.”</i>	50
J.	CLAIM 15	50
1.	<i>Claim 15: “The system of claim 14, wherein the memory contains further instructions that, when executed by the processing circuitry for generating the query, further configure the system to: generate a second prompt for the LLM which when executed by the LLM outputs the query, wherein the second prompt is generated based on any one of: the received incident input, the data schema, the plurality of scenarios, and a combination thereof.”</i>	50
K.	CLAIM 16	50
1.	<i>Claim 16: “The system of claim 14, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to: train the LLM further on a plurality of database queries, each database query executable on the security database.”</i>	50
VI.	CONCLUSION	51
VII.	MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)	52
A.	REAL PARTY-IN-INTEREST	52
B.	RELATED MATTERS	52
C.	LEAD AND BACK-UP COUNSEL	52
D.	37 C.F.R. § 42.8(B)(4) – SERVICE INFORMATION	53

TABLE OF AUTHORITIES

Cases:

<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007)	14, 15
<i>Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.</i> , 868 F.3d 1013 (Fed. Cir. 2017)	6
<i>Orca Security Ltd. v. Wiz, Inc.</i> , 1:23-cv-00758-JLH-SFR (D. Del.)	5

Statutes:

35 U.S.C. § 102(a)(1).....	6
35 U.S.C. § 102(a)(2)	11
35 U.S.C. § 103.....	1, 4
35 U.S.C. § 311.....	1

Regulations:

37 C.F.R. § 42.6(e).....	60
37 C.F.R. § 42.8(b)(1).....	51
37 C.F.R. § 42.8(b)(3).....	52
37 C.F.R. § 42.8(b)(4).....	52, 53
37 C.F.R. § 42.24(d).....	59
37 C.F.R. § 42.63(e).....	60
37 C.F.R. § 42.8(b)(1).....	52
37 C.F.R. § 42.104(a).....	4
37 C.F.R. § 42.104(b).....	60

37 C.F.R. § 42.104(b)(3).....	5
37 C.F.R. § 42.105.....	60
37 C.F.R. § 42.105(b)	60

I. INTRODUCTION

Pursuant to 35 U.S.C. § 311, Petitioner Orca Security Ltd. (“Petitioner”) requests *Inter Partes* Review (“IPR”) of Claims 1-5 and 11-16 (the “Challenged Claims”) of U.S. Patent No. 12,001,549 (“the ’549 Patent”) and seeks cancellation of the Challenged Claims under 35 U.S.C. § 103. Petitioner shows that the cybersecurity concepts claimed in the ’549 Patent were well known by the earliest priority date, and further shows that a person of ordinary skill in the art (“POSITA”) would have been motivated to combine these well-known teachings to arrive at the claimed invention.

The ’549 Patent claims a method, apparatus, and system for employing a large language model (“LLM”) in combination with a security database to provide “cybersecurity incident response[.]” Ex. 1001, ’549 Patent, *Abstract*. This alleged advance was not new. Presented here are two prior art references that render obvious the alleged invention of the ’549 Patent. Specifically, the prior art presented shows that the use of basic LLMs and security databases for cybersecurity threat response and mitigation was not novel or new, at even the earliest possible priority date of the ’549 Patent. As such, Petitioner respectfully requests that the Board institute *Inter Partes* Review for the ’549 Patent.

II. SUMMARY OF THE '549 PATENT

A. Description of the '549 Patent

The '549 Patent describes “a method and system for providing cybersecurity incident response.” Ex. 1001, '549 Patent, 7:23-24. More specifically, the '549 Patent describes a method and system for using a Large Language Model (LLM) to process cybersecurity incidents and initiate corresponding threat mitigation actions. *See id.*, 14:50-53.

The claimed system first receives a cybersecurity “incident input,” which the '549 Patent discloses includes “a natural language query, an unstructured query, an alert, a combination thereof, and the like.” *Id.*, 7:40-44. After the incident input is provided to the LLM, it is mapped, based on the output of an LLM, to a corresponding cybersecurity scenario or “risk,” such as “a workload with an exposed database [or] a workload connected to a public network[.]” *Id.*, 7:57-60. After the incident input is mapped to a particular scenario, the LLM then generates a database query for a security database before executing said query and ultimately initiating a corresponding mitigation action. *Id.*, 16:20, 16:41-43, 17:15-21; *see also id.*, Figs. 5, 6.

The LLM of the '549 Patent is trained on various data, including data schema representing the computing environment, incident data relating to a scenario, or database queries. *Id.*, cl. 3, 5. The LLM is further provided with prompts generated

via predefined templates that can be modified by various inputs or results. *See id.*, 13:20-21, 15:61-62, 17:35-52.

B. Summary of the Prosecution History

A first office action identified allowable subject matter but rejected all originally presented claims 1-21 on the grounds of nonstatutory double patenting over certain claims of Application No. 18/466,832. Ex. 1002, File History, 131. After Patent Owner filed a terminal disclaimer (*id.*, 118–20), the Examiner allowed the application, finding that the prior art did not teach a limitation “wherein each scenario is associated with an incidence response; generating a query based on the received incident input and the mapped scenario; executing the query on a security database.” *Id.*, 16.

C. Level of Skill of a Person Having Ordinary Skill in the Art

A POSITA at the time of the '549 Patent would have had a bachelor's degree in computer engineering, computer science, electrical engineering, or a related field, with about two to three years of experience in the fields of cybersecurity and artificial intelligence. Additional education or experience might substitute for the above requirements. *See* Ex. 1003, Zadok Decl., ¶¶ 31-36.

III. REQUIREMENTS UNDER 37 C.F.R. § 42.104

A. Grounds for Standing Under 37 C.F.R. § 42.104(a)

Petitioner certifies that the '549 Patent is available for IPR and that Petitioner is not barred or estopped from requesting an IPR on the grounds identified herein.

B. Identification of Challenge Under 37 C.F.R. § 42.104(b)

Petitioner requests that the Board institute IPR of the Challenged Claims under AIA 35 U.S.C. § 103. The '549 Patent issued from an application filed on January 31, 2024, which claims priority as a continuation-in-part of an application filed on August 28, 2023. For purposes of this Petition, Petitioner does not challenge whether the claims are entitled to the August 28, 2023, priority date. The relief requested by Petitioner is that claims 1-5 and 11-16 be canceled based on the grounds below:

Proposed Grounds of Unpatentability ¹	Exhibits
<u>Ground 1:</u> Claims 1-5 and 11-16 are obvious under 35 U.S.C. § 103 over <i>Peters</i> in view of <i>Lal</i> .	1004, 1005, 1006

¹ All obviousness grounds include the knowledge of a POSITA.

C. Claim Construction Under 37 C.F.R. § 42.104(b)(3)

Petitioner notes that the parties have proffered the following constructions for two terms of the '549 Patent in the related—and stayed²—district court litigation.

Term/Phrase	Petitioner's Proposed Construction	Patent Owner's Proposed Construction
“wherein each scenario is associated with an incidence response”	Indefinite	“incidence” should read “incident”. Otherwise, no construction is necessary.
“based on”	“dependent on”; the limitations of claims 1, 11 and 12 must be performed in order	No construction is necessary. Alternatively, “influenced by”; the limitations of a method claim are not required to be performed in order.

Ex. 1010, Amended Joint Claim Construction Chart, 15. The district court’s *Markman* hearing “is CONTINUED” and has not been rescheduled in view of the instituted Wiz IPRs and related stay. *Orca Security Ltd. v. Wiz, Inc.*, 1:23-cv-00758-JLH-SFR, Dkt. 223, Oral Order (Dec. 17, 2024).

Because the proposed ground below renders the Challenged Claims obvious under each party's proposed construction of “based on,” and under Patent Owner’s

² The district court litigation was stayed “through final written decision in each of the Wiz IPRs.” Ex. 1008, 2. The “Wiz IPRs” refer to the instituted IPRs filed by Patent Owner against Petitioner’s patents.

Proposed Construction of “wherein each scenario is associated with an incidence response,” Petitioner does not contend that any claim construction is necessary. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (construction is required “only to the extent necessary to resolve the controversy”).³

IV. SHOWING OF ANALOGOUS, PRIOR ART

A. U.S. Patent No. 11,303,666 (“Peters”) (Ex. 1004)

U.S. Patent No. 11,303,666, titled “Systems and Methods for Intelligent Cyber Security Threat Detection and Mitigation Through an Extensible Automated Investigations and Threat Mitigation Platform,” issued on April 12, 2022, from a provisional application filed on October 14, 2020. The non-provisional application was filed on September 29, 2021. Peters therefore qualifies as prior art to the ’549 Patent under 35 U.S.C. § 102(a)(1).

Peters is analogous prior art to the ’549 Patent because Peters is in the same field of endeavor as the ’549 Patent and is reasonably pertinent to a problem with which the inventors of the ’549 Patent were involved. Ex. 1003, Zadok Decl., ¶¶ 72-74. Like the ’549 Patent, Peters is directed to cybersecurity, specifically to threat

³ Petitioner does not waive any argument in litigation that claim terms in the ’549 Patent are indefinite or additional terms need construction.

detection and mitigation. *Compare* Ex. 1001, '549 Patent, 1:16-18 (“The present disclosure relates generally to cybersecurity incident response and specifically to initiating mitigation actions in response to detected cybersecurity threats”) *with* Ex. 1004, Peters, 1:17-20 (“a new and useful cyber threat detection and mitigation system and methods in the cybersecurity field.”). Additionally, like the '549 Patent, Peters addresses problems related to guiding remediation and mitigation solutions. *Compare* Ex. 1001, '549 Patent, 1:51-56 *with* Ex. 1004, Peters, 1:45-47. Peters was not cited or considered during the prosecution of the '549 Patent. *See generally* Ex. 1002.

Peters describes a “system and method for handling a cybersecurity event.” Ex. 1004, Peters, *Abstract*. Peters’ invention seeks to “create improved systems and methods for intelligently scaling detection capabilities of a security operations service while maintaining quality and improving its technical capabilities to efficiently respond to an increasingly large volume of security risks to computing and organizational computing assets.” Ex. 1004, Peters, 1:48-54. To address this problem, Peters employs a multi-step process for receiving, validating, classifying, investigating, and mitigating cybersecurity alerts as outlined in Figure 2, below:

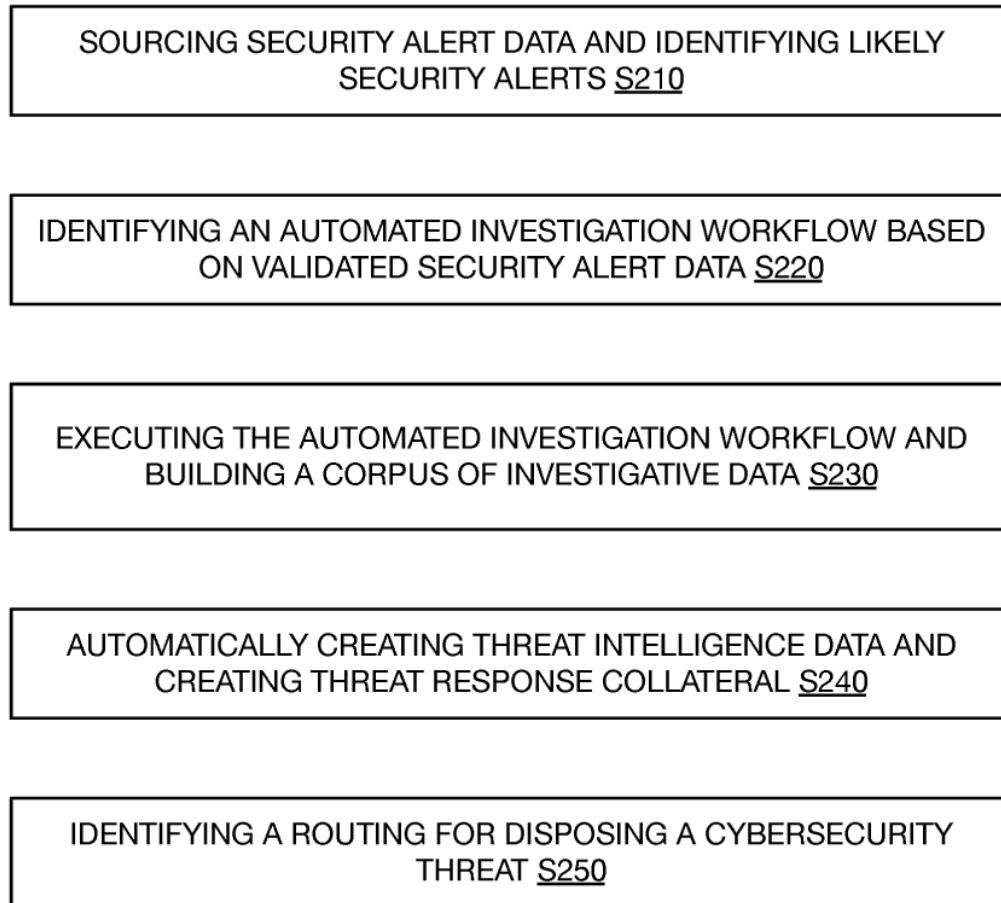


FIGURE 2

Id., Fig. 2.

Peters first describes receiving and normalizing a security alert from a “plurality of distinct sources of cyber security alert data.” *Id.*, 5:50; *see id.*, 5:65-6:3, 8:21-28. After receiving the security alert, which, preferably relates to “event data or activity data occurring with or involving one or more computing resources, computer network resources, data resources, cloud-based resources, any computer accessible digital entity or device, and/or the like,” Peters’ system classifies the alert

as a “distinct cybersecurity threat type or category.” *Id.*, 8:42-46, 10:36-41, 11:20-26. In one implementation, Peters relies on machine learning models for this classification. *Id.*, 10:14-17; *see also id.*, 11:15-33, 12:41-47. Specifically, Peters describes an “ensemble of machine learning classifiers” where each machine learning classifier is trained for a particular cybersecurity threat:

For instance, an ensemble may include a first machine learning classifier that predicts a likelihood that a validated security alert may be a “suspicious login” or not, a second machine learning classifier that predicts a likelihood that the validated security alert may be an “authorized access” to a secured computing asset, and a third machine learning classifier that predicts a likelihood that the validated security alert may be a “violation of computer security policy” by an employee or the like.

Id., 11:15-26.

The classification of the security alert then “inform[s] a selection and/or automated initialization of an automated investigation workflow.” *Id.*, 12:45-47. An investigation workflow identifies investigation instructions or rules “specifically designed for a given validated security alert[.]” *Id.*, 13:17-19. These instructions, when executed, “inform or identify sources of investigation data, configure and execute one or more API calls, perform an analysis or and extraction of specific investigation data, [and] deliver[] and/or present[] investigation results.” *Id.*, 12:53-56; *see also id.*, 14:28-33, 14:49-61. The resulting “security investigation data from any suitable source including, but not limited to, an Active Directory and the like,” (*Id.*, 14:46-48), is then compiled “into a corpus of investigation data,” (*Id.*, 14:66-

67) from which “cybersecurity threat intelligence data” (*Id.*, 15:29-32) is derived by “identifying and extracting informative and/or probative pieces of data from the corpus of investigation data” (*Id.*, 15:34-36; *see also id.*, 15:53-57).

Finally, “a threat mitigation route or a threat disposal route based on the cybersecurity intelligence data” is identified. *Id.*, 18:5-7. These mitigation or disposal routes can include, but are not limited to, “escalating the validated security alert to a confirmed state of malicious attack or incident, ... informing a subscriber regarding a threat risk state of the validated security alert, and... deescalating the validated security alert to a determined state of non-malicious thereby informing a termination state of a validated security alert.” *Id.*, 18:13-20. Peters’ system can also “selectively prescribe” a threat mitigation or handling route or immediately triage the alert. *Id.*, 18:39-44.

B. U.S. Patent Publication No. 2024/0403428 (“Lal Patent Publication”) (Ex. 1005) and U.S. Provisional Patent Application No. 63/472,227 (“Lal”) (Ex. 1006)⁴

U.S. Patent Publication No. 2024/0403428, titled “System and Method for Utilizing Large Language Models and Natural Language Processing Technologies

⁴ For purposes of this Petition, Petitioner cites primarily to the Lal Provisional Application (“Lal”), though all disclosures are supported by the Lal Patent Publication.

to Pre-Process and Analyze Data to Improve Detection of Cyber Threats” published on December 5, 2024, and claims priority to U.S. Provisional Patent Application No. 63/472,227, titled “A Cybersecurity System,” filed on June 9, 2023, and which is incorporated by reference. Lal therefore qualifies as prior art to the ’549 Patent under 35 U.S.C. § 102(a)(2).

Lal is analogous prior art to the ’549 Patent because Lal is in the same field of endeavor as the ’549 Patent and is reasonably pertinent to a problem with which the inventors of the ’549 Patent were involved. Ex. 1003, Zadok Decl., ¶¶ 76-78. Like the ’549 Patent, Lal is directed to cybersecurity and the use of artificial intelligence therefor. *Compare* Ex. 1001, ’549 Patent, *Title*, 7:23-29, Claims 1-21 *with* Ex. 1006, Lal, [0002], [0004]. Additionally, like the ’549 Patent, Lal addresses problems related to transparency and ease of use of a cybersecurity system. *Compare* Ex. 1001, ’549 Patent, 1:51-54 *with* Ex. 1006, Lal, [0018], [0044], [0046]. Neither Lal’s Publication or Provisional Application was cited or considered during the prosecution of the ’549 Patent.

Lal describes “[m]ethods, systems, and apparatus... for an Artificial Intelligence based cyber security system.” Ex. 1006, Lal, [0004]. Lal explains that “[t]he advent of large language models (LLMs) has opened up new possibilities in bolstering cyber security practices.” *Id.*, [0019]. Such new possibilities achievable by Lal’s LLM include “provid[ing] valuable cyber analyst answers,” “translat[ing]

user questions into API queries within an AI-based cyber security platform,” and “generat[ing] summaries and recommendations for users of cyber security appliances,” among many others. *Id.*, [0018]. To do so, Lal’s LLM is trained on a wide breadth of general and cybersecurity-specific information, like API specifications, cybersecurity documentation, common cybersecurity breaches, along with possible incident responses. *See, e.g., id.*, [0017], [0034], [0039], [0044].

C. A POSITA Would Have Been Motivated to Combine Peters and Lal

Prior to the filing of the ’549 Patent, a POSITA would have recognized that Peters provided a detailed system for detecting and responding to cybersecurity alerts. As LLMs grew in popularity and accuracy (for example, ChatGPT was released to the public on November 30, 2022, and was rapidly and widely adopted in the field of cybersecurity), a POSITA working with Peters’ cybersecurity system would have sought to incorporate such an LLM. *See* Ex. 1003, Zadok Decl., ¶¶ 80-103. In particular, a POSITA would have been motivated to combine Peters’ cybersecurity detection and remediation system with Lal’s cybersecurity LLM for the reasons set forth below.

First, Lal provides an express motivation to incorporate its system with Peters’ because it describes the advantages of employing its cybersecurity LLM within broader cybersecurity mitigation systems. For example, Lal explains that the effectiveness of its LLM “lies in its ability to access and integrate diverse data

sources” and to “tap into external data stores, such as threat intelligence platforms and vulnerability databases, to enrich its understanding of the threat landscape.” Ex. 1006, Lal, [0134]; *see also id.*, [0133] (“[t]he artificial intelligence-based cyber security analyst tool can handle data collection and extraction from multiple sources, reducing the manual effort required to gather relevant information for analysis.”). Because Peters describes querying from external sources, such as “an Active Directory” (Ex. 1004, Peters, 14:44-48), a POSITA working with Peters’ system would have been motivated to implement Lal’s LLM to promote its efficiency in responding to cybersecurity alerts. Ex. 1003, Zadok Decl., ¶¶ 81-82.

Second, Lal teaches that its LLM can “augment threat analysis and investigations to handle threat investigations at machine speed and scale.” Ex. 1006, Lal, [0126]; *see also id.*, [0049] (“By prioritizing breaches, organizations can allocate their resources more effectively...”). Considering Peters’ desire “for intelligently scaling detection capabilities of a security operations service” (Ex. 1004, Peters, 1:49-50), a POSITA would have been motivated to implement Lal’s augmented analysis with Peters’ system to permit handling more cybersecurity alerts and threats in less time. Ex. 1003, Zadok Decl., ¶ 83.

Third, Lal teaches that, by implementing its LLM, “organizations can enhance the transparency of AI-based cybersecurity platforms.” Ex. 1006, Lal, [0044]. For example, “[t]he LLM can interpret user queries, translate them into API

queries, and then explain how the AI system arrives at its decisions.” *Id.* Doing so “helps build trust and confidence in the AI system’s capabilities, as users can gain insights into the decision-making process and validate the system’s outputs.” *Id.* Peters similarly describes an analyst or administrator interacting with the disclosed cybersecurity system. *See* Ex. 1004, Peters, 7:47-67. As such, a POSITA would have been motivated to give the analyst operating in parallel with Peters’ system greater insight into the decisions being made by the system. Ex. 1003, Zadok Decl., ¶¶ 84-86. Each of the above express teachings from Lal would have encouraged a POSITA to modify Peters’ existing cybersecurity response system with Lal’s cybersecurity-focused LLM. *Id.*, ¶¶ 81-86; *see also KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 415–21 (2007).

Fourth, Peters already contemplates using artificial intelligence, namely machine learning classifiers, to process security alerts. Ex. 1004, Peters, 11:3-6. In fact, Peters lists a gamut of machine learning types that can be implemented by its system. *Id.*, 6:18-7:3 (“... a kernel method,... a clustering method,... an associated rule learning algorithm..., an artificial neural network model,... a deep learning algorithm...”). Considering Peters’ express disclosure of using machine learning and the boom of LLMs after Peters’ filing date, a POSITA would have looked to incorporate an LLM into Peters’ system, similar to Peters’ pre-existing use of machine learning classifiers. Ex. 1003, Zadok Decl., ¶¶ 87-88. In particular, a

POSITA would have sought out an LLM trained for cybersecurity purposes, like that described in *Lal. Id.*, ¶ 89. Because Lal’s LLM is trained on a vast array of cybersecurity-specific information, a POSITA would have found Lal’s LLM an obvious choice for a cybersecurity system like Peters’. *Id.*, ¶ 90.

Fifth, introducing Lal’s LLM to Peters’ system would have consolidated Peters’ ensemble of machine learning classifiers and improved Peters’ overall performance. *Id.*, ¶ 91. Specifically, Peters notes that its invention seeks to “efficiently respond to an increasingly large volume of security risks to computing and organization computing assets.” Ex. 1004, Peters, 1:52-54. Replacing Peters’ ensemble of individual machine learning classifiers, each specific to only one type of cybersecurity threat, with Lal’s single, already-trained LLM would have further promoted the efficiency of Peters’ classification process and improved its ability to respond quickly to cybersecurity alerts and threats. Ex. 1003, Zadok Decl., ¶¶ 91-92. Such a combination would have constituted a simple substitution of one known element (e.g., Peters’ machine learning) for another (e.g., Lal’s LLM) to obtain predictable results (e.g., more efficient cybersecurity response). *Id.*, ¶ 92; *see also KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 415–21 (2007).

Sixth, a POSITA would have been motivated to integrate Lal’s LLM into Peters’ method for configuring its API queries. *Id.*, ¶ 93. Peters describes storing many source-specific API configuration parameters to produce an investigative

query for execution on a particular data source. Ex. 1004, Peters, 14:55-61. That is, each data source has a different set of rules for making a successful query. *Id.*, ¶ 94. Instead of maintaining these rules and updating them as systems change, a POSITA would have been motivated to incorporate an LLM, particularly Lal's LLM already trained on various API specifications to generate API queries executable on various data sources. *Id.*, ¶ 94. Again, in light of Peters' efficiency and scalability aims, substituting many configuration parameters with an LLM would allow Peters' system to detect and respond to more cybersecurity alerts with greater speed and accuracy. *Id.*, ¶ 95.

Seventh, a POSITA would have been motivated to incorporate Lal into Peters' system because of its highly contextualized cybersecurity-focused template-style prompts. *Id.*, ¶ 96. For example, Lal discloses that, to provide greater context to its LLM, it relies on prompts with placeholders for data related to the system. *See e.g.*, Ex. 1006, Lal, [0031]–[0033]. Implementing such template prompts would allow Peters' system to classify alerts and generate queries specific to the cybersecurity alert that caused the initiation of Peters' threat response method. Ex. 1003, Zadok Decl., ¶¶ 97-98. Lal's ability to incorporate active data into its LLM's output would have been recognized as a critical feature for a POSITA seeking to incorporate an LLM into Peters' system. *Id.*, ¶¶ 97-98. Such a combination would have constituted a simple substitution of one known element (i.e., LLMs and

associated template prompts) for another (i.e., Peters' machine learning classifiers and query configuration parameters) to obtain predictable results (i.e., quicker and more accurate threat classifications and query generations). *Id.*, ¶ 98.

Eighth, considering Peters' lack of disclosure regarding storage of its instructions, a POSITA would have been motivated to incorporate Lal's non-transitory storage mediums to store the software required to implement Peters' cybersecurity system. *Id.*, ¶ 99.

Ninth, a POSITA would have had a reasonable expectation of success in combining Peters and Lal because both inventions focus on widely understood and recognized cybersecurity systems and methods. *Id.*, ¶¶ 100-01; *see e.g.*, Ex. 1004, Peters, Fig. 2. A POSITA would have understood that such processes could have been implemented using a combination of software and hardware without undue experimentation. Ex. 1003, Zadok Decl., ¶ 101.

Tenth, combining Peters and Lal would have combined known methods to yield predictable results. For example, Lal describes that a benefit of its LLM is the ability to automate human decision-making tasks. *See e.g.*, Ex. 1006, Lal, [0133]. Incorporating Lal's LLM to Peters' cybersecurity system would predictably result in the system becoming more automated and with less constant reliance on supervision by highly trained IT personnel. Ex. 1003, Zadok Decl., ¶ 102. This combination would have allowed for less-trained personnel to rely on the

understanding of the LLM for processing and responding to cybersecurity alerts and threats. *Id.*, ¶ 102.

V. GROUND 1: CLAIMS 1-5 AND 11-16 ARE OBVIOUS OVER PETERS AND LAL

Peters in view of Lal discloses each and every element of claims 1-5 and 11-16. Ex. 1003, Zadok Decl., ¶¶ 104-226. As such, Peters in view of Lal renders obvious those claims to a POSITA in view of the knowledge of such a person in this art. *Id.*

A. Claim 1

1. Claim 1(Pre): “A method for providing cybersecurity incident response, comprising:”

To the extent the preamble is limiting, Peters discloses *a method for providing cybersecurity incident response*⁵ within the meaning of the ’549 Patent. Specifically, Peters discloses a “cyber threat detection and mitigation system and methods in the cybersecurity field.” Ex. 1004, Peters, 1:18-20; *see also id.*, Title, Abstract, cl. 1; Ex. 1003, Zadok Decl., ¶ 104.

2. Claim 1(a): “receiving an incident input based on a cybersecurity event”

Peters discloses *receiving an incident input based on a cybersecurity event* (e.g., receiving Peters’ security alert). The ’549 Patent discloses that an *incident*

⁵ Petitioner denotes claim language by italics.

input can be “a natural language query, an unstructured query, an alert, a combination thereof, and the like[.]” Ex. 1001, ’549 Patent, 7:41-42. Elsewhere, the ’549 Patent states that the “incident input” may include “a query, a statement, and a combination thereof.” *Id.*, cl. 2, cl. 13, 3:33-34, 4:43-44, 5:58-59. Still elsewhere, the ’549 Patent states that “in an embodiment, the incident input 510 includes an alert, a natura (*sic*) language query, a combination thereof, and the like.” *Id.*, 14:23-25. Thus, the ’549 Patent broadly characterizes incident inputs into various types of (unstructured) queries and statements/alerts. As an example of an alert, the ’549 Patent gives the statement ““EC2 virtual machine with ID of i-012abcd34efghi56 infected with malware with SHA1 hash of 3395856ce81f2b7382dee72602f798b642f14141’.” *Id.*, 15:49-52. A POSITA would thus have understood that such an alert is simply a non-query or a statement. Ex. 1003, Zadok Decl., ¶ 106.

In the same way, Peters collects “event data and raw security alert data from one or more distinct sources of data.” Ex. 1004, Peters, 8:13-25; *see also id.*, 8:52-9:2. Peters further discloses that event or security alert data can include “one or more of vendor alert data, continuous telemetry data, subscriber data, investigative data, other data, and/or the like.” *Id.*, 8:25-28. Preferably, the security alert data “relate[s] to event data or activity data occurring with or involving one or more computing resources, computer network resources, data resources, cloud-based resources, any

computer accessible digital entity or device, and/or the like[.]” *Id.*, 8:42-46. Peters further discloses that security alert data is processed by a security alert engine and routed to a downstream cybersecurity threat handling processes as a “validated” security alert. *Id.*, 9:35-40. These validated security alerts are *based on a cybersecurity event* because Peters teaches that its “validated security alerts” are processed to determine an identification of the cybersecurity threat type. *Id.*, 9:10-54; *see also* 8:35-38 (disclosing that “validated security alert” and “cybersecurity threat” can be used interchangeably). Thus, Peters discloses *receiving an incident input based on a cybersecurity event* within the meaning of the ’549 Patent.

3. Claim 1(b): “generating a prompt for a large language model (LLM) based on the received incident input”

Peters in view of Lal discloses this limitation. As described for the previous limitation, Peters teaches receiving an *incident input* (e.g., Peters’ security alert). *See e.g.*, Ex. 1004, Peters, 8:29-37.

Lal teaches cybersecurity-minded large language models (“LLMs”) with many possible uses, including mitigating ongoing cyberattacks, answering client queries, and making API queries, among others. Ex. 1006, Lal, [0004]. To do so, Lal uses prompts to provide greater context to guide the LLM’s decision-making and outputs related to the cybersecurity system. *Id.*, [0031]. Generally, prompts are “set[s] of instructions provided to an LLM that program[] the LLM by customizing it and/or enhancing or refining its capabilities.” Ex. 1011, White, 1; *see also* Ex.

1003, Zadok Decl., ¶¶ 112-115. Examples of such prompts provided to the LLM(s)

in Lal include:

The cyber security system has seen a login from a rare IP in Russia as well as regular events from Cambridge given the following possible actions <SET OF ACTIONS> which would be the best one to take. Explain why you made that decision.

Ex. 1006, Lal, [0033].

Given that this Month we have seen the following autonomous response actions: [block IP 1, block IP 2, block user 3, block user 4] produce me some security recommendations.

Id., [0064].

Summarize John Boy's office365 SaaS events for the last week and decide if his current permissions set which is [X Y Z] is appropriate or whether they can be reduced.

Id., [0066].

Prompts are commonplace in LLM configuration and training and are often prepared as templates with placeholders for dynamic data or information, like those described in Lal. Ex. 1003, Zadok Decl., ¶¶ 112-116; *see also* Ex. 1012, Arvidsson, 10 (“Applying templates is another technique that is established to be a viable strategy when prompting... [T]hey provide a structure that is predefined which could reduce the risk of leaving out vital information [and] are reusable which could save time and effort[.]”). The '549 Patent itself describes using template-style prompts: “[i]n certain embodiments, the prompt is generated based on a template such as ‘how did RESOURCE_ID become infected with MALWARE_ID’.” Ex. 1001, '549

Patent, 17:38-52 (further explaining that “MALWARE_ID is a data field which is modified based on the incident input.”).

Lal provides example prompts specific to incoming cybersecurity alert analysis, and a POSITA would have understood that such prompts are tailored to each specific task. Ex. 1006, Lal, [0033], [0064], [0066]; Ex. 1003, Zadok Decl., ¶ 116. Moreover, a POSITA would have recognized that, to generate a custom prompt for cyber threat analysis, the placeholders in the templates described in Lal, for example, “a/b/c,” “[X Y Z],” or “<SET OF ACTIONS>,” would be replaced with information relevant to the specific task and associated security alert. *Id.*, ¶¶ 119. For example, when discussing prompts, Lal explains explicitly that the “LLM would be trained on the cybersecurity API and **be aware of where information can be found**, what kind of data each endpoint returns, and the data the cybersecurity AI analysis considers relevant for similar activity.” Ex. 1006, Lal, [0054]. Thus, a POSITA would have understood that Lal’s system would know where to find the placeholder information to generate a template prompt based on an inbound cybersecurity alert. Ex. 1003, Zadok Decl., ¶ 117.

In the proposed Peters-Lal combination, the prompts for Lal’s LLM would be tailored with data specific to Peters’ inbound cybersecurity alert. *Id.*, ¶ 119. Specifically, Lal’s system would generate a prompt for the LLM by inputting Peters’ security alert into one or more of Lal’s pre-existing prompt templates. *Id.*, ¶ 119.

Because Peters already teaches using a “security alert machine learning system 114” to classify inbound security alerts, a POSITA would have been motivated to modify Peters with the LLM as taught by Lal. Ex. 1004, Peters, 6:10-14; *see also id.*, 6:18-65; Ex. 1003, Zadok Decl., ¶ 120. In particular, a POSITA would have recognized that an LLM, especially one with significant cybersecurity-specific training data as taught by Lal, is a sophisticated machine learning tool that would be well-suited for Peters’ cybersecurity alert and mitigation method and system. Ex. 1003, Zadok Decl., ¶ 120. And for the reasons explained in Section IV.C., a POSITA would have found it obvious and been motivated to modify Peters’ system to incorporate an LLM and associated template-prompts as taught by Lal. *Id.*, ¶¶ 80-103. Thus, Peters in view of Lal discloses *generating a prompt for a large language model (LLM)* (e.g., Lal’s template-style prompts for use with its one or more LLMs) *based on the received incident input* (e.g., Peters’, received security alert) within the meaning of the ’549 Patent.

4. Claim 1(c): “configuring the LLM to generate an output based on the generated prompt”

Lal discloses this limitation as taught by the ’549 Patent. Specifically, the ’549 Patent explains that “**the prompt**, when processed by the LLM, **configures** the LLM to output an explanation for the result.” Ex. 1001, ’549 Patent, 17:41-43; *see also id.*, 14:49-51 (“**the prompt**, when executed by the LLM, **configures** the LLM to select a scenario from the plurality of scenarios”) (emphasis added).

Likewise, Lal discloses prompts tailored to certain cybersecurity events. *See, e.g., id.*, [0033], [0070], [0064], [0065], [0066]; *see also* limitation 1(b). Each prompt, by providing context to the LLM, directs (i.e., “configures”) the LLM to produce an output specific to the information included in the prompt. Ex. 1003, Zadok Decl., ¶125. For example, the following prompt from Lal: “The cybersecurity system has seen a login from a rare IP in Russia as well as regular events from Cambridge given the following possible actions <SET OF ACTIONS> which would be the best one to take. Explain why you made that decision[.]” would configure the LLM to output a best action and an explanation of how it reached that conclusion. Ex. 1006, Lal, [0033]. After ingesting another template prompt tailored to Peters’ cybersecurity alert, as explained above, a POSITA would have understood that Lal’s highly trained LLM would produce an output similarly tailored to the generated prompt and cybersecurity alert data. Ex. 1003, Zadok Decl., ¶ 125; *see also* Ex. 1006, Lal, [0044] (explaining that users can “validate the system’s outputs”).

Thus, Peters in view of Lal discloses *configuring the LLM* (e.g., Lal’s LLM) *to generate an output based on the generated prompt* (e.g., Lal’s template prompt generated with information related to Peters’ inbound security alert) within the meaning of the ’549 Patent.

5. ***Claim 1(d): “mapping the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence⁶ response”***

Peters in view of Lal discloses this limitation. The ’549 Patent explains that “each scenario corresponds to a cybersecurity situation... the cybersecurity situation includ[ing] a cybersecurity risk, such as a workload with an exposed database [and/or] a workload connected to a public network[.]” Ex. 1001, ’549 Patent, 7:50-60.

Peters describes that its method “may function to compute and/or predict a likely cybersecurity threat type for a validated security alert.” Ex. 1004, Peters, 10:34-36. After receiving and validating a security alert, Peters implements an “ensemble of machine learning classifiers that may function to predict at least one of a plurality of distinct cybersecurity threat types for a validated security alert.” *Id.*, 11:4-6; *see also id.*, 10:13-21 (“the one or more machine learning models computing against a given feature dataset of a subject security alert may function to output a classification label for the subject security alert[.]”). Based on the features of a validated security alert, each machine learning classifier within the ensemble is trained to predict whether a particular cybersecurity threat type exists. *Id.*, 11:9-12.

⁶ For the sake of this Petition only, Petitioner accedes, *arguendo*, in Patent Owner’s contention that “incidence” should be read as “incident” in this claim.

The machine learning classifier with the highest probability or confidence value “may be selected or identified as the cybersecurity threat type for a target validated security alert.” *Id.*, 11:30-34. This security alert is then classified by the machine learning ensemble as a particular cybersecurity threat type, for example, a “suspicious login,” “authorized access,” or “violation of computer security policy.” *Id.*, 11:19-26. A POSITA would have understood that such a threat type classification *map[s] the received incident input into one of a plurality of scenarios*. Ex. 1003, Zadok Decl., ¶¶ 129-130.

Because Peters describes using machine learning to classify cybersecurity alerts into particular cybersecurity threat types (*id.*, 6:10-14, 6:18-65), a POSITA would have been motivated to introduce Lal’s cybersecurity-specific LLM into Peters’ method to streamline and improve the classification of cybersecurity alerts. Ex. 1003, Zadok Decl., ¶ 130. Alone, Peters discloses multiple distinct machine learning classifiers, each responsible for identifying a specific cybersecurity threat type. Ex. 1004, Peters, 11:9-26. A POSITA would have recognized that replacing Peters’ multiple machine learning classifiers with Lal’s single LLM that can, for example, classify “events into buckets,” “output a prediction about the entity,” or “determine whether [a] pattern is indicative of behavior of a malicious actor-human, program or other threat” would result in a single, cybersecurity-focused LLM to be trained and monitored instead of Peters’ swath of threat-specific machine learning

classifiers. Ex. 1006, Lal, [0082], [0285] [0276]; *see also id.*, [0198]; Ex. 1003, Zadok Decl., ¶ 130. Further, for the reasons discussed in Section IV.C., a POSITA would have found it obvious and been motivated to modify Peters' system to incorporate one or more LLMs as taught by Lal. Ex. 1003, Zadok Decl., ¶¶ 80-103.

Finally, the '549 Patent explains that “[i]n some embodiments, the incident response includes an alert, a query, a natural language query, a database query, a combination thereof, and the like.” Ex. 1001, '549 Patent, 15:45-48. Similarly, Peters describes that each “class or distinct type of validated security alert may be mapped or electronically linked to a distinct automated investigation workflow.” Ex. 1004, Peters, 12:23-25. For example, “a phishing security threat may trigger a phishing automated investigation workflow” or “a suspicious login threat[] may trigger a suspicious login automated investigation workflow.” *Id.*, 12:34-40. These investigation workflows are cybersecurity threat type-specific and “configured for automatically performing one or more investigative tasks for handling” detected cybersecurity alerts. *Id.*, 12:29-32. Specifically, an investigation workflow initiates one or more “investigation subroutines” or “actions” to identify rules and instructions for handling a particular cybersecurity threat type. *Id.*, 12:51.

Based on the cybersecurity threat type classification, an investigation subroutine can “inform or identify sources of investigation data, configure and execute one or more API calls, [and] perform an analysis of an extraction of specific

investigation data[.]” *Id.*, 12:53-55. Like the *incident responses* identified by the ’549 Patent, Peters also explains that an investigation workflow can include a “set or corpus of investigative queries” that define questions relevant to a particular security alert. *Id.*, 13:12-19. For example, “if a cybersecurity threat type includes a suspicious login, one or more of the investigation criteria or investigative queries for this cybersecurity threat type may include a request for a source IP address of the suspicious login.” *Id.*, 13:31-35. For these reasons, *each scenario*, or cybersecurity threat type, *is associated with an incidence response*.

Thus, Peters in view of Lal discloses *mapping the received incident input* (e.g., Peters’ received security alert) *into a scenario* (e.g., Peters’ cybersecurity threat type classification) *of a plurality of scenarios* (e.g., Peters’ plurality of distinct cybersecurity threat types) *based on the output of the LLM* (e.g., Lal’s LLM output), *wherein each scenario is associated with an incidence response* (e.g., Peters’ investigation workflow).

6. Claim 1(e): “generating a query based on the received incident input and the mapped scenario”

Peters discloses this limitation. Once the validated security alert has been mapped to a corresponding investigation workflow, as described above, the investigation workflow compiles a “corpus of investigation data,” from which “threat intelligence data” is extracted to determine how best to mitigate a cybersecurity alert. Ex. 1004, Peters, 2:11-20; *see also id.*, 15:53-57, 15:34-36. To

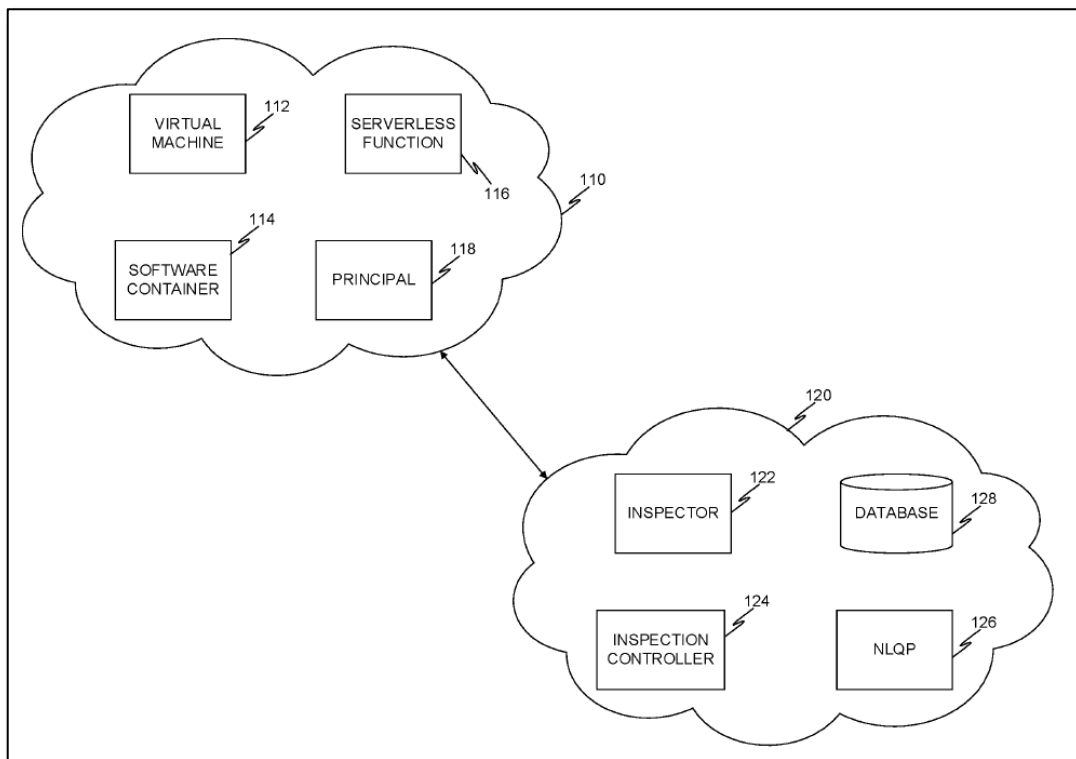
source this investigation data, Peters describes that S230 “may function to **generate one or more investigation queries** to the one or more sources requesting additional security alert information and/or details relating to the validated security alert” based on the investigative instructions and rules associated with the particular investigation workflow, cybersecurity threat, and alert. *Id.*, 14:15-18 (emphasis added); *see also id.*, 14:23-28, 14:49-52.

For example, “if a cybersecurity threat type includes a suspicious login, one or more of the investigation scripts or rules may define API call parameters for **configuring an API call for sourcing data** relating to an agent identifier and/or IP address associated with the suspected suspicious login.” *Id.*, 14:28-33 (emphasis added). Moreover, each source of security investigation data is “associated with or otherwise mapped to a set of configuration parameters or requirements for creating a successful API request to the source.” *Id.*, 14:53-56. Additionally, a POSITA would have recognized that such API calls are a *query* in the form of a structured request for information. *See* Ex. 1003, Zadok Decl., ¶¶ 137-38.

Thus, Peters discloses *generating a query* (e.g., Peters’ investigative queries) *based on the received incident input* (e.g., Peters’ received security alert) *and the mapped scenario* (e.g., Peters’ cybersecurity threat type) within the meaning of the ’549 Patent.

7. ***Claim 1(f): “executing the query on a security database, the security database including a representation of a computing environment”***

Peters discloses *executing the query on a security database, the security database including a representation of a computing environment*. According to the '549 Patent, a security database can be columnar, graphical, structured, or unstructured, and a computing environment can be “a cloud computing environment, a networked environment, an on-premises environment, a combination thereof, and the like.” Ex. 1001, '549 Patent, 9:60-62, 8:42-45. The '549 Patent depicts both a database 128 and a computing environment 110:



Id., Figure 1

In addition to generating investigative queries, as explained above, Peters also describes automatically executing those queries on various sources of security investigation data to build the corpus of investigative data. *See e.g.*, Ex. 1004, Peters, 12:54, 14:60-61. Namely, the “security investigations engine[,]” which is responsible for collecting security investigation data, “may be programmatically... linked or in communication with a plurality of distinct sources of security data.” *Id.*, 14:35-37. These sources can include “security APIs of third-party security services or vendors and/or native security APIs or native services of a subscriber to the system or service implementing the method 200.” *Id.*, 14:40-44. Peters also notes that the security investigations engine can “collect security investigation data from any suitable source including... an Active Directory and the like.” *Id.*, 14:45-48.

Peters’ sources of security data are *security databases including a representation of a computing environment* as claimed by the ’549 Patent. Ex. 1003, Zadok Decl., ¶¶ 143, 145-49. Specifically, Peters discloses an Active Directory as an example of an accessible security data source. *Id.*, 14:45-48. A POSITA would have known that “Active Directory” is a Microsoft service and technology that stores information about the objects on the network, such as servers, network users, and computer accounts. Ex. 1003, Zadok Decl., ¶ 146; *see also* Ex. 1013, 1. Specifically, Active Directory includes: 1) a schema outlining the “objects and attributes contained in the directory,” 2) a “global catalog that contains information about

every object in the directory,” and 3) a “query and index mechanism” that allows users to locate objects within the network. *Id.* Active Directory was well-known before the time of the ’549 Patent; for example, U.S. Patent Pub. No. 2018/004870 to Ohayon et al. (“Ohayon”), similarly describes Active Directory as “a directory database” that includes data such as “information regarding computing devices (e.g., computers and servers) that reside in the network, users of the network (e.g., authorized identities and valid credentials), directory sites, applications/services that make use of the network, network attributes/topology, organization hierarchy, etc.” Ex. 1014, Ohayon, [0068].

Additionally, Microsoft’s Active Directory manages network security features like user permissions and passwords. *See* Ex. 1020, *Passwords Technical Overview*; Ex. 1021, *Active Directory Security Groups*; *see also* Ex. 1003, Zadok Decl., ¶¶ 148-49. A POSITA would have also recognized that a source with the ability to return a specific answer to a “request for a source IP address of the suspicious login[,]” would have included access to network information regarding the *computing environment*. Ex. 1004, Peters, 13:34-35. For at least these reasons, a POSITA would have understood Peters’ disclosure of an Active Directory to be a *security database including a representation of a computing environment*. Ex. 1003, Zadok Decl., ¶ 150.

Thus, because Peters discloses executing generated API calls on sources including Active Directory, Peters discloses *executing the query* (e.g., Peters' executing API calls) *on a security database, the security database including a representation of a computing environment* (e.g., Peters' Active Directory) within the meaning of the '549 Patent.

8. Claim 1(g): “initiating a mitigation action based on a result of the executed query.”

Peters discloses this limitation as claimed by the '549 Patent. Specifically, the '549 Patent explains that a mitigation action can include “generating a notification, generating an alert,... generating a risk score, updating a risk score, [or] initiating a remediation action.” Ex. 1001, '549 Patent, 17:16-21.

Peters teaches that S250 “may function to identify[] a **threat mitigation route** or a threat disposal route based on the cybersecurity intelligence data.” Ex. 1004, Peters, 18:4-7 (emphasis added). As described above, Peters' method generates, executes, and collects results from investigative queries related to a security alert and the resulting threat type classification and associated investigation workflow. *See id.*, 14:15-18, 14:23-28, 14:45-52, 12:54; *see also* Ex. 1003, Zadok Decl., ¶ 153. Such investigative “data, information, and/or other results obtained through an inquiry via one or more API calls or the like,” comprise the corpus of investigation data. *Id.*, 12:63-65.

Next, to identify a route for mitigating the cybersecurity threat, Peters describes extracting threat intelligence data from the corpus of investigation data specific to the initiated investigation workflow. *Id.*, 15:61-67. “For example, for a cybersecurity threat type of ‘suspicious_login’, S240 may function to extract an IP address value from a corpus of investigation data for satisfying an investigative requirement for source location data for the cybersecurity threat type of ‘suspicious_login’.” *Id.*, 15:53-57. Using the derived threat intelligence data, Peters’ S250 can “**selectively prescribe** one of a plurality of distinct threat mitigation and/or threat handling routes... or **immediately execut[e]** a disposal or triage of the validated security alert.” *Id.*, 18:39-44 (emphases added). The mitigation and handling routes include but are not limited to “escalating the validated security alert to a confirmed state of malicious attack or incident, ... informing a subscriber regarding a threat risk state of the validated security alert, and... deescalating the validated security alert to a determined state of non-malicious thereby informing a termination state.” *Id.*, 18:14-20.

Thus, Peters discloses *initiating a mitigation action* (e.g., Peters’ executing a threat mitigation route) *based on a result of the executed query* (e.g., Peters’ cybersecurity intelligence data derived from results of one or more investigative or API queries) within the meaning of the ’549 Patent.

B. Claim 2

1. ***Claim 2: “The method of claim 1, wherein the incident input includes any one of: a query, a statement, and a combination thereof.”***

Peters discloses this claim. As discussed above, the '549 Patent broadly divides the disclosed incident input components into different types of queries and non-query statements. *See* limitation 1(a), *supra*. The only example of a non-query incident input is an alert. Ex. 1001, '549 Patent, 15:49-52 (explaining that “EC2 virtual machine with ID of i-012abcd34efghi56 infected with malware with SHA1 hash of 3395856ce81f2b7382dee72602f798b642f14141” is an alert); *see also supra* limitation 1(a). Accordingly, a POSITA would recognize that an alert is a type of *statement*. Ex. 1003, Zadok Decl., ¶ 162.

Peters similarly discloses that *incident inputs* (i.e., security alerts in the terminology of Peters) can include statements. For example, Peters describes that “[a] validated security alert or a likely security alert, as interchangeably referred to herein, preferably relates to a **service-generated indication or notification** of a probable or a likely event that may include a cybersecurity attack, a suspicious cyber event, a breach of cybersecurity policy and/or the like.” Ex. 1004, Peters, 9:5-10 (emphasis added). A POSITA would have understood that such a notification relating to a cybersecurity event is a statement. Ex. 1003, Zadok Decl., ¶¶ 163-64. Contemporaneous literature confirms that alerts were provided in the form of

statements. *Id.*, ¶ 164. For example, U.S. Patent No. 8,146,146 to Coviello et al. (“Coviello”) titled, “Method and apparatus for integrated network security alert information retrieval,” shows an example of a “network security alert” in Figure 3 below:

FIG. 3

301						
302	303	304	305	306	307	308
DATE/TIME	ALERT ID	EVENT	SOURCE IP	SOURCE PORT	DEST. IP	DEST. PORT
2004/07/08 18:48:44	A12345678	SYSTEM SCANNING ACROSS IP ADDRESSES ON SAME PORT	135.210.96.46	137	201.1.98.18	137

ALERTS (1)

Ex. 1015, Coviello, Figure 3 (annotated). Specifically, the alert includes an Event Field 304, which includes the statement “System Scanning Across IP Addresses on Same Port” to describe the security alert. *Id.* Thus, Peters discloses or renders obvious *the method of claim 1, wherein the incident input* (e.g., Peters’ security alert) *includes... a statement as claimed by the ’549 Patent.*

C. Claim 3

1. ***Claim 3: “The method of claim 1, wherein the LLM is trained on any one of: a data schema utilized in representing the computing environment, incident data classified to a scenario, the plurality of scenarios, and a combination thereof.”***

Peters in view of Lal discloses Claim 3.

Lal’s LLM is trained on a wide variety of cybersecurity-specific information. Ex. 1003, Zadok Decl., ¶¶ 168-173. Namely, in addition to “text-based data... from the open internet such as webpages and sources such as scientific research, books, forums or social media posts[,]” Lal’s LLM is also explicitly trained for cybersecurity troubleshooting, “cyber threat landscapes[,] and current active cyber threats.” Ex. 1006, Lal, [0004]; *see e.g., id.*, [0017], [0039] (“trained on both the **entire cyber security documentation corpus**”), [0082] (“trained on existing module structures and **event classification**”), [0100] (“trained on certification information and responses”) (emphases added). Ex. 1003, Zadok Decl., ¶ 168.

First, Peters in view of Lal discloses that the LLM is trained on *a data schema utilized in representing the computing environment*. In the proposed Peters-Lal combination, investigative or API queries would be outputted based on the prompt provided to Lal’s LLM and would be executable on different sources of security data, for example, Active Directory. Ex. 1003, Zadok Decl., ¶ 171; *see also Claim 4, infra* (detailing Lal’s query generation). Because Lal explains that “[t]he LLM would be trained on the cyber security API and be aware of where information can

be found [and] what kind of data each endpoint returns,” a POSITA would have understood that Lal’s LLM would be trained on Active Directory’s schema (*see* EX1013, *Microsoft Active Directory Domain Services Overview*, 1). Ex. 1003, Zadok Decl., ¶¶ 169-71.

Second, Lal’s LLM can be “provided with [i.e., trained] a list of breaches and their triggers. . . [from which] the LLM can consider the severity of the breach, the impact on critical systems or data, and the potential risk to the organization.” Ex. 1006, Lal, [0049]; Ex. 1003, Zadok Dec. ¶¶ 168, 172-73 (explaining that providing information to an LLM is a form of training). A POSITA would have understood that Lal’s “list of breaches and their triggers” is a *plurality of scenarios* where “each scenario corresponds to a cybersecurity situation... [like] a workload having malware [or] a workload generating network traffic with a prohibited domain,” as described in the ’549 Patent. Ex. 1001, ’549 Patent, 7:47-55; Ex. 1003, Zadok Decl., ¶ 172. For the reasons further discussed in Section IV.C., a POSITA would have found it obvious and been motivated to modify Peters’ system to incorporate LLMs as taught by Lal. *Id.*, ¶ 171.

Thus, Peters in view of Lal discloses *the method of claim 1, wherein the LLM* (e.g., Lal’s LLM) *is trained on any one of: a data schema utilized in representing the computing environment* (e.g., Peters’ Active Directory schema), *incident data classified to a scenario, the plurality of scenarios* (e.g., Lal’s list of breaches and

their triggers or Peters' cybersecurity threat types), *and a combination thereof* as claimed.

D. Claim 4

1. ***Claim 4: “The method of claim 3, wherein generating the query further comprises: generating a second prompt for the LLM which when executed by the LLM outputs the query, wherein the second prompt is generated based on any one of: the received incident input, the data schema, the plurality of scenarios, and a combination thereof.”***

Peters in view of Lal discloses Claim 4. Namely, Lal teaches a cybersecurity-specific LLM that is configured to “assist in querying detailed log data” and “can handle data extraction from multiple sources[.]” Ex. 1006, Lal, [0017], [0133]; *see also id.*, [0059] (“The LLM understands the intent behind the query and translates it into the appropriate search syntax[.]”). To perform this data extraction from multiple data sources, Lal’s LLM is trained on an API specification “so that the LLM can understand how to make API requests and **translate questions into API queries[.]**” *Id.*, [0004] (emphasis added).

As explained for limitation 1(b), *supra*, Lal relies on template prompts to direct the output of, or “configure,” its one or more LLMs. *See* Ex. 1003, Zadok Decl., ¶¶ 113-19. Lal goes on to explain that the LLMs can specifically “be prompted to query.” *Id.*, [0098]. “For instance, an LLM can be prompted to query OSINT (Open-Source Intelligence) tools or a customer’s threat intelligence tools by connecting to their APIs.” *Id.* Lal’s first generated prompt, described for limitation

1(b), *supra*, is associated with the task of classifying Peters' inbound security alert into a particular cybersecurity threat type. For the querying task described above, however, a POSITA would have recognized that Lal's LLM must receive another, or *second prompt which when executed by the LLM outputs the query*. Ex. 1003, Zadok Decl., ¶ 181-83. Because prompts are "instructions," LLMs often rely on multiple rounds of prompting to refine their results. *Id.*; *see also id.*, ¶¶ 59-61. By doing so, the LLM can tailor its output to for greater accuracy or desirability to the user or system. *Id.* Furthermore, a POSITA would have understood that LLMs are "conversational" in nature and iteratively build off of previous outputs and prompts. *Id.*; *see also* EX1011, White, 13 ("The model is therefore constantly being fed the previous outputs and the prompt.").

Accordingly, Peters in view of Lal teaches *the second prompt is generated based on . . . the received incident input*. As described in limitation 1(b), *supra*, Lal describes using template prompts that provide context for its LLM prompts. In the proposed Peters-Lal combination, the prompts for Lal's LLM would be tailored with data specific to Peters' inbound cybersecurity alert. *Id.*, ¶ 185. Specifically, Lal's system would generate a first prompt for the LLM by inputting Peters' alert data into one or more of its pre-existing prompt templates. *Id.* Then, depending on the classification of the security alert and resulting investigation workflow, Lal's system would generate a *second prompt*, this time updating the template data fields with

information for both the security alert and for outputting a query for a particular data source. *Id.*

Lal also teaches *the second prompt is generated based on . . . the data schema*. Notably, Lal’s LLM is “trained on the cyber security API and [is] aware of where information can be found, what kind of data each endpoint returns, and the data the cyber security AI Analyst considers relevant for similar activity.” *Id.*, [0054]; *see also id.*, [0046] (providing an exemplary prompt of “Given our API does find me all devices that communicated over TCP in the last 4 days.”). A POSITA would have understood that Lal’s LLM would need to be prompted with the *data schema*, which “defines the classes of objects and attributes contained in the directory,” in order to output a successful query to a source like Peters’ Active Directory. EX1013, *Microsoft Active Directory Domain Services Overview*, 1; Ex. 1003, Zadok Decl., ¶¶ 186-87.

A POSITA would have been motivated to combine Lal’s prompts for querying with Peters’ configuration of investigative queries. *Id.*, ¶ 185. Specifically, a POSITA would have been motivated to replace Peters’ many data source-specific API configuration parameters with Lal’s trained LLM and template prompts for querying to improve the accuracy of and streamline Peters’ query configuration process. *Id.* In the proposed Peters-Lal combination, information related to Peters’ security alert, the resulting cybersecurity threat type, and the data schema specific to

the target data source would be incorporated into Lal's prompt placeholders (or data fields) so that Lal's LLM can output a query tailored to the security alert for successful execution on Peters' security database, (e.g., Active Directory). *Id.*, ¶¶ 185-86. Introducing information specific to the alert and the data source to the generated prompt allows for that information to be considered by the LLM and its output, making its output more accurate and trustworthy. *Id.*, ¶ 185. Also, for the reasons further discussed in Section IV.C., a POSITA would have found it obvious and been motivated to modify Peters' system to incorporate LLMs as taught by Lal. *Id.*, ¶¶ 80-103.

Thus, Peters in view of Lal discloses *the method of claim 3, wherein generating the query* (e.g., Lal's generating an API query) *further comprises: generating a second prompt for the LLM* (e.g., Lal's template prompt specific to querying) *which when executed by the LLM* (e.g., Lal's one or more LLMs) *outputs the query* (e.g., Lal's API query), *wherein the second prompt is generated based on any one of: the received incident input* (e.g., Peters' received security alert), *the data schema* (e.g., Peters' Active Directory schema), *the plurality of scenarios, and a combination thereof* as claimed.

E. Claim 5

1. ***Claim 5: “The method of claim 3, further comprising: training the LLM further on a plurality of database queries, each database query executable on the security database.”***

Peters in view of Lal discloses this claim. As explained for limitation 1(c), *supra*, Lal's LLM is trained on a wide variety of cybersecurity-specific information, including information specific to querying. For example, Lal's LLM can translate “the intent behind the [user's] query. . . into the appropriate search syntax.” Ex. 1006, Lal, [0059]. To do so, Lal's LLM is “trained on the search syntax for the cybersecurity Advanced Search, allowing users to query for detailed log data in a friendly human-speech way.” *Id.*, [0060]. Additionally, Lal discloses training the “LLM with an API specification” so the “LLM can **understand how to make API requests and translate questions into API queries.**” *Id.*, [0045] (emphasis added). As explained above, a POSITA would have understood an API query to be a database query because Lal's API is connected to a database and provides responsive information to API calls. Ex. 1003, Zadok Decl., ¶¶ 191-92; *see also* limitation 1(e).

Further, a POSITA would have understood that API specifications include example queries. Ex. 1003, Zadok Decl., ¶¶ 193-95. For example, APIs such as the REST API and the JSON API have specifications that include example API queries (i.e., database queries). Ex. 1017; 1018; *see also* Ex. 1003, Zadok Decl., ¶¶ 193, 194 (citing Exs. 1017 and 1018).

It would have been obvious to combine Peters' security data source (e.g., Active Directory) with Lal's LLMs trained on specific syntaxes with examples of database queries specific to Active Directory in order to output API queries taught by Lal. Ex. 1003, Zadok Decl., ¶ 196; *see also* Limitation 1(f). A POSITA would have understood that an LLM trained on an API specification including example queries and search syntax would allow for outputting of valid queries that are executable on the security data source. *Id.*

Thus, Peters in view of Lal discloses *training the LLM* (e.g., Lal's LLM) *further on a plurality of database queries* (e.g., Lal's API specifications including queries), *each database query executable on the security database* (e.g., Peters' Active Directory) as claimed.

F. Claim 11

- 1. Claim 11(Pre): “A non-transitory computer-readable medium storing a set of instructions for providing cybersecurity incident response, the set of instructions comprising: one or more instructions that, when executed by one or more processors of a device, cause the device to:”**

To the extent the preamble is limiting, Peters in view of Lal teaches this limitation. Peters' cybersecurity system is implemented by “one or more computing servers, a distributed network of computers (e.g., a cloud network), computer processors, and the like[.]” Ex. 1004, Peters, 6:14-17 (emphasis added). A POSITA would have understood that the system of Peters, whether implemented on

computing servers or on a cloud network, would have included a non-transitory computer-readable medium storing a set of instructions for providing the method of Peters' cybersecurity incident response system. Ex. 1003, Zadok Decl., ¶¶ 198-200.

Lal further describes that its software is performed by “one or more processing components,” and discloses that “any portions of the method, apparatus and system implemented as software can be stored in **one or more non-transitory storage devices in an executable format to be executed by one or more processors.**” Ex. 1006, Lal, [0319]. A POSITA would have understood that, for Peters' claimed method and system to be executed by its disclosed computer processors, the associated instructions must be stored on non-transitory storage devices, such as those expressly taught by Lal. Ex. 1003, Zadok Decl., ¶ 199. Accordingly, to the extent not disclosed in Peters, a POSITA would have been motivated to incorporate Lal's non-transitory storage media to store the software required to implement the cybersecurity system of the Peters-Lal system. *Id.*, ¶ 200. And for the reasons further discussed in Section IV.C., a POSITA would have found it obvious and been motivated to modify Peters' system to incorporate one or more LLMs as taught by Lal. Ex. 1003, Zadok Decl., ¶¶ 80-103.

Thus, Peters in view of Lal discloses *a non-transitory computer-readable medium* (e.g., Lal's non-transitory storage devices) *storing a set of instructions for providing cybersecurity incident response* (e.g., Peters' and Lal's methods for

cybersecurity mitigation), *the set of instructions comprising: one or more instructions that, when executed by one or more processors of a device (e.g., Peters' computer processors), cause the device (e.g., Lal's cyber security appliance) to perform the subsequent steps as claimed.*

2. ***Claim 11(a): "receive an incident input based on a cybersecurity event;"***

See Limitation 1(a).

3. ***Claim 11(b): "generate a prompt for a large language model (LLM) based on the received incident input;"***

See Limitation 1(b).

4. ***Claim 11(c): "configure the LLM to generate an output based on a cybersecurity event;"***

See Limitation 1(c).

5. ***Claim 11(d): "map the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence response;"***

See Limitation 1(d).

6. ***Claim 11(e): "generate a query based on the received incident input and the mapped scenario;"***

See Limitation 1(e).

7. ***Claim 11(f): "execute the query on a security database, the security database including a representation of a computing environment; and"***

See Limitation 1(f).

8. *Claim 11(g): “initiate a mitigation action based on a result of the executed query.”*

See Limitation 1(g).

G. **Claim 12**

1. *Claim 12(Pre): “A system for providing cybersecurity incident response comprising:”*

To the extent the preamble is limiting, Peters discloses this limitation. As explained for the preceding claims, Peters teaches “a cybersecurity threat detection and threat mitigation system” that “includes a security alert engine 110, an automated security investigations engine 120, and a security threat mitigation user interface 130.” Ex. 1004, Peters, 5:36-42.

2. *Claim 12(a): “a processing circuitry;”*

Peters discloses this limitation. Specifically, Peters discloses that its cybersecurity response system can include “computing servers, a distributed network of computers (e.g., a cloud network), **computer processors**, and the like[.]” Ex. 1004, Peters, 6:14-17 (emphasis added). A POSITA would have recognized that computer processors constitute *a processing circuitry*. Ex. 1003, Zadok Decl., ¶ 211.

Lal further describes that its software is performed by “one or more **processing components**,” and discloses that “any portions of the method, apparatus and system implemented as software can be stored in one or more non-transitory storage devices in an executable format to be executed by **one or more processors**.” Ex. 1006, Lal, [0319]. Accordingly, to the extent not disclosed in Peters, a POSITA

would have been motivated to incorporate Lal's processing components to implement the cybersecurity system of the Peters-Lal system. Ex. 1003, Zadok Decl., ¶ 211. And for the reasons further discussed in Section IV.C., a POSITA would have found it obvious and been motivated to modify Peters' system to incorporate one or more LLMs as taught by Lal. Ex. 1003, Zadok Decl., ¶¶ 80-103.

3. *Claim 12(b): "a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:"*

Peters in view of Lal teaches a *memory containing instructions*. As described above in Claim 11(pre), Peters in view of Lal teaches or renders obvious a memory containing instructions for executing by the processing circuitry. Further, Lal's storage medium can be, "for example, a physical computer readable medium such as a semiconductor memory or solid-state memory, magnetic tape, a removable computer diskette, a random-access memory (RAM), [or] a read-only memory (ROM)[.]" Ex. 1006, Lal. A POSITA would have understood that, for Peters' claimed method and system to be executed by a processor, the associated instructions must be stored in memory. Ex. 1003, Zadok Decl., ¶ 214. Accordingly, a POSITA would have been motivated to incorporate Lal's memory to store the software required to implement Peters' cybersecurity system. *Id.* For the reasons further discussed in Section IV.C., a POSITA would have found it obvious and been

motivated to modify Peters' system to incorporate one or more LLMs as taught by Lal. Ex. 1003, Zadok Decl., ¶¶ 80-103.

4. ***Claim 12(c): “receive an incident input based on a cybersecurity event;”***

See Limitation 1(a).

5. ***Claim 12(d): “generate a prompt for a large language model (LLM) based on the received incident input;”***

See Limitation 1(b).

6. ***Claim 12(e): “configure the LLM to generate an output based on a cybersecurity event;”***

See Limitation 1(c).

7. ***Claim 12(f): “map the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence response;”***

See Limitation 1(d).

8. ***Claim 12(g): “generate a query based on the received incident input and the mapped scenario;”***

See Limitation 1(e).

9. ***Claim 12(h): “execute the query on a security database, the security database including a representation of a computing environment; and”***

See Limitation 1(f).

10. ***Claim 12(i): “initiate a mitigation action based on a result of the executed query.”***

See Limitation 1(g).

H. Claim 13

1. *Claim 13: “The system of claim 12, wherein the incident input includes any one of: a query, a statement, and a combination thereof.”*

See Claim 2.

I. Claim 14

1. *Claim 14: “The system of claim 12, wherein the LLM is trained on any one of: a data schema utilized in representing the computing environment, incident data classified to a scenario, the plurality of scenarios, and a combination thereof.”*

See Claim 3.

J. Claim 15

1. *Claim 15: “The system of claim 14, wherein the memory contains further instructions that, when executed by the processing circuitry for generating the query, further configure the system to: generate a second prompt for the LLM which when executed by the LLM outputs the query, wherein the second prompt is generated based on any one of: the received incident input, the data schema, the plurality of scenarios, and a combination thereof.”*

See Claim 4.

K. Claim 16

1. *Claim 16: “The system of claim 14, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to: train the LLM further on a plurality of database queries, each database query executable on the security database.”*

See Claim 5.

VI. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests *inter partes* review of the Challenged Claims.

Respectfully submitted,

ERISE IP, P.A.

BY: /s/ Robin A. Snader
Robin A. Snader, Reg. No. 66,085
robin.snader@eriseip.com
7015 College Boulevard, Suite 700
Overland Park, Kansas 66211
(913) 777-5600 Telephone
(913) 777-5601 Facsimile

COUNSEL FOR PETITIONER

VII. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)

A. Real Party-In-Interest

Petitioner is the only real party-in-interest. 37 C.F.R. § 42.8(b)(1).

B. Related Matters

The '549 Patent is presently the subject of a stayed patent infringement lawsuit filed against Petitioner in the United States District Court for the District of Delaware. *Orca Security Ltd. v. Wiz, Inc.*, Case No. 23-758-JLH-SRF (D. Del.). The '549 Patent is not currently and has not been subject to any other district court litigation, *inter partes* review, reexamination, or other post-grant proceeding.

C. Lead and Back-Up Counsel

Petitioners provide the following designation and service information for lead and back-up counsel. 37 C.F.R. § 42.8(b)(3) and (b)(4).

Lead Counsel	Back-Up Counsel
Robin A. Snader (Reg. No. 66,085) robin.snader@eriseip.com PTAB@eriseip.com <u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601	Eric A. Buresh (Reg. No. 50,394) eric.buresh@eriseip.com Kathleen Fitterling (Reg. No. 62,950) kathleen.fitterling@eriseip.com Chris Schmidt (Reg. No. 63,982) chris.schmidt@eriseip.com <u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601

D. 37 C.F.R. § 42.8(b)(4) – Service Information

Please address all correspondence to the lead and back-up counsel as shown above. Petitioner consents to electronic service by e-mail at the e-mail addresses provided above.

CLAIM LISTING APPENDIX
U.S. Patent No. 12,001,549 for Claims 1-5 and 11-16

Claim Designation	Claim Language
Claim 1(P)	A method for providing cybersecurity incident response, comprising:
Claim 1(a)	receiving an incident input based on a cybersecurity event;
Claim 1(b)	generating a prompt for a large language model (LLM) based on the received incident input;
Claim 1(c)	configuring the LLM to generate an output based on the generated prompt;
Claim 1(d)	mapping the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence response;
Claim 1(e)	generating a query based on the received incident input and the mapped scenario;
Claim 1(f)	executing the query on a security database, the security database including a representation of a computing environment; and
Claim 1(g)	initiating a mitigation action based on a result of the executed query.
Claim 2	The method of claim 1, wherein the incident input includes any one of: a query, a statement, and a combination thereof.
Claim 3	The method of claim 1, wherein the LLM is trained on any one of: a data schema utilized in representing the computing environment, incident data classified to a scenario, the plurality of scenarios, and a combination thereof.
Claim 4	The method of claim 3, wherein generating the query further comprises: generating a second prompt for the LLM which when executed by the LLM outputs the query, wherein the second prompt is generated based on any one of: the received incident input, the data schema, the plurality of scenarios, and a combination thereof.
Claim 5	The method of claim 3, further comprising: training the LLM further on a plurality of database queries, each database query executable on the security database.

Claim Designation	Claim Language
Claim 11(P)	A non-transitory computer-readable medium storing a set of instructions for providing cybersecurity incident response, the set of instructions comprising:
Claim 11(a)	one or more instructions that, when executed by one or more processors of a device, cause the device to:
Claim 11(b)	receive an incident input based on a cybersecurity event;
Claim 11(c)	generate a prompt for a large language model (LLM) based on the received incident input;
Claim 11(d)	configure the LLM to generate an output based on the generated prompt;
Claim 11(e)	map the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence response;
Claim 11(f)	generate a query based on the received incident input and the mapped scenario;
Claim 11(g)	execute the query on a security database, the security database including a representation of a computing environment; and
Claim 11(h)	initiate a mitigation action based on a result of the executed query.
Claim 12(P)	A system for providing cybersecurity incident response comprising:
Claim 12(a)	a processing circuitry;
Claim 12(b)	a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:
Claim 12(c)	receive an incident input based on a cybersecurity event;
Claim 12(d)	generate a prompt for a large language model (LLM) based on the received incident input;
Claim 12(e)	configure the LLM to generate an output based on the generated prompt;
Claim 12(f)	map the received incident input into a scenario of a plurality of scenarios based on the output of the LLM, wherein each scenario is associated with an incidence response;
Claim 12(g)	generate a query based on the received incident input and the mapped scenario;

Claim Designation	Claim Language
Claim 12(h)	execute the query on a security database, the security database including a representation of a computing environment; and
Claim 12(i)	initiate a mitigation action based on a result of the executed query.
Claim 13	The system of claim 12, wherein the incident input includes any one of: a query, a statement, and a combination thereof.
Claim 14	The system of claim 12, wherein the LLM is trained on any one of: a data schema utilized in representing the computing environment, incident data classified to a scenario, the plurality of scenarios, and a combination thereof.
Claim 15	The system of claim 14, wherein the memory contains further instructions that, when executed by the processing circuitry for generating the query, further configure the system to: generate a second prompt for the LLM which when executed by the LLM outputs the query, wherein the second prompt is generated based on any one of: the received incident input, the data schema, the plurality of scenarios, and a combination thereof.
Claim 16	The system of claim 14, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to: train the LLM further on a plurality of database queries, each database query executable on the security database.

APPENDIX OF EXHIBITS

Exhibit	Description
1001	U. S. Patent No. 12,001,549
1002	File History of U.S. Patent No. 12,001,549—Application No. 18/428,794
1003	Declaration of Erez Zadok, Ph. D. (this document)
1004	U. S. Patent No. 11,303,666 to Peters et al. (“Peters”)
1005	U. S. Patent Application Publication No. 2024/0403428 (“Lal Patent Application Publication”)
1006	U. S. Provisional Patent Application No. 63/472,227 (“Lal”)
1007	Complaint from <i>Orca Security, Ltd. v. Wiz, Inc.</i> , No. 23-758-JHL-SRF (D. Del.)
1008	Order Issuing Stay from <i>Orca Security, Ltd. v. Wiz, Inc.</i> , No. 23-758-JHL-SRF (D. Del.)
1009	Curriculum Vitae of Erez Zadok, Ph.D.
1010	Joint Claim Construction from <i>Orca Security, Ltd. v. Wiz, Inc.</i> , No. 23-758-JHL-SRF (D. Del.)
1011	Jules White et al., <i>A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT</i>
1012	Simon Arvidsson and Johan Axell, <i>Prompt Engineering Guidelines for LLMs in Requirements Engineering</i>
1013	Microsoft, <i>Microsoft Active Directory Domain Services Overview</i> , Web Archive (August 16, 2022)
1014	U. S. Patent Pub. No. 2018/004870 to Ohayon et al.
1015	U. S. Patent No. 8,146,146 to Coviello et al.
1016	November 30, 2022, OpenAI Announcement Regarding ChatGPT’s Public Release
1017	RESTAPI Specification, Web Archive (February 16, 2023)
1018	JSON Specification, Web Archive (May 24, 2023)
1019	Merriam Webster, Dictionary Definition of “Network”
1020	Microsoft, <i>Passwords Technical Overview</i>
1021	Microsoft, <i>Active Directory Security Groups</i> , Web Archive (May 21, 2023)
1022	Toloka, <i>Empowering AI Through LLM Training</i>
1023	Isabelle Nguyen, <i>What Is a Large Language Model (LLM)?</i>
1024	Dorian Drost, <i>Different Ways of Training LLMs</i>
1025	Angie Lee, <i>No Hang Ups with Hangul: KT Trains Smart Speakers, Customer Call Centers with NVIDIA AI</i>

Exhibit	Description
1026	Greyling, <i>12 Prompt Engineering Techniques</i>
1027	Kojima, <i>Large Language Models are Zero-Shot Reasoners</i>
1028	Mengsay Loen, <i>Mastering Prompt Engineering for Effective LLM Output: Tips, Techniques, and Warning</i>
1028	Loem, <i>Mastering Prompt Engineering for Effective LLM Output: Tips, Techniques, and Warning</i>

CERTIFICATION OF WORD COUNT

The undersigned certifies pursuant to 37 C.F.R. §42.24 that the foregoing Petition for *Inter Partes* Review, excluding any table of contents, mandatory notices under 37 C.F.R. §42.8, certificates of service or word count, or appendix of exhibits or claim listings, contains 10,713 words according to the word-processing program used to prepare this document (Microsoft Word).

Dated: June 4, 2025

ERISE IP, P.A.

BY: /s/ Robin A. Snader
Robin A. Snader, Reg. No. 66,085
robin.snader@eriseip.com
7015 College Blvd., Suite 700
Overland Park, KS 66211
P: (913) 777-5600
F: (913) 777-5601

COUNSEL FOR PETITIONER

**CERTIFICATE OF SERVICE ON PATENT OWNER
UNDER 37 C.F.R. § 42.105**

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b), the undersigned certifies that on June 4, 2025, a complete and entire copy of this Petition for *Inter Partes* Review and Exhibits were provided via Federal Express to the Patent Owner by serving the correspondence address of record for the '549 Patent:

M&B IP Analysts, LLC
150 Morristown Road
Suite 205
Bernardsville, NJ 07924-2626

Further, a courtesy copy of this Petition for *Inter Partes* Review was sent via email to Patent Owner's litigation counsel:

Frederick L Cottrell, III (cottrell@rlf.com)
Kelly E. Farnan (farnan@rlf.com)
Christine Dealy Haynes (haynes@rlf.com)
Alex Miller (alex.miller@wsgr.com)
Callie Davidson (ccdavidson@wsgr.com)
Jordan R. Jaffe (jjaffe@wsgr.com)
Lisa D. Zang (lzang@wsgr.com)
Praatika Prasad (pprasad@wsgr.com)

Respectfully submitted,

ERISE IP, P.A.

BY: /s/ Robin A. Snader
Robin A. Snader, Reg. No. 66,085
robin.snader@eriseip.com
7015 College Blvd., Suite 700
Overland Park, KS 66211
P: (913) 777-5600

IPR2025-01086

U.S. Patent No. 12,001,549

F: (913) 777-5601

COUNSEL FOR PETITIONER