

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TREND MICRO, INC.
Petitioner

v.

TAASERA LICENSING LLC

Case No. IPR-2023-00800
Patent No. 8,990,948

**PETITION FOR INTER PARTES REVIEW
OF U.S. PATENT NO. 8,990,948**

Filed on behalf of Trend Micro, Inc.

By: Robert M. Hansen

Matthew J. Anderson

Scott B. Amankwatia

The Marbury Law Group, PLLC

11800 Sunrise Valley Drive

15th Floor

Reston, VA 20191

Tel: (703) 391-2900

Fax: (703) 391-2901

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Grounds for Standing (37 C.F.R. § 42.104(a)).....	1
III.	Identification of Challenge (37 C.F.R. § 42.104(b))	1
A.	Citation of Prior Art	1
B.	Statutory Ground for Challenge (37 C.F.R. § 42.104(b))	2
IV.	The '948 patent	3
A.	Claim of Priority of the '948 patent	3
B.	Overview of the '948 patent.....	3
C.	Summary of the Prosecution History	8
D.	Level of Ordinary Skill in the Art	11
E.	Claim Construction.....	11
V.	HOW THE CLAIMS ARE UNPATENTABLE	12
A.	Overview of Asserted Prior Art	13
i.	Mukherjee.....	13
ii.	Cooper	18
B.	Motivation to Combine Mukherjee and Cooper	21
C.	Comparison of Mukherjee and Cooper to the Challenged Claims	26

1. Claim 1	26
1[a]. A method of providing real-time operational integrity of an application on a native computing environment, the method comprising:	26
1[b].monitoring, by a plurality of sensory inputs, one or more of network dialogs of the application, system operations initiated by the application, a runtime configuration of the application, resource utilization by the application, and integrity of the application;.....	31
1[c].generating real-time behavior based events for determining the real-time operational integrity of the application executing on the native computing environment which includes a network analyzer, an integrity processor, an event correlation matrix, a risk correlation matrix, and a trust supervisor;.....	35
1[d]. correlating, by the event and risk correlation matrix, threat classifications based on the temporal sequence of the generated real-time behavior based events; and.....	46
1[e]. displaying, in a plurality of runtime dashboards of an administrative console of the computing environment, real-time status indications for operational integrity of the application.	50

2. Claim 2	59
------------------	----

**The method of claim 1, wherein the real-time operations of applications
are monitored continuously.....59**

3. Claim 3	60
------------------	----

**The method of claim 1, wherein the event and behavior correlation
engine receives one or more of: 60**

4. Claim 4	63
------------------	----

**The method of claim 3, wherein the scans are performed as per a
schedule, and wherein the schedule is one or more of hourly, daily,
weekly, bi-weekly, or monthly, by a third party collaboration service to
score the severity of an application based on the vulnerability,
compliance, and patch status to affirm positive presence of a threat. 63**

5. Claim 5	67
------------------	----

**The method of claim 1, wherein the plurality of runtime dashboards are
a plurality of application runtime reputation score dashboards that
comprise one or more of: 67
a global view dashboard; 67
a system configuration dashboard;.....67
a resource utilization dashboard; 67
an application integrity dashboard; and 67**

	a network activity dashboard.....	67
VI.	NO BASIS FOR DISCRETIONARY DENIAL	71
VII.	CONCLUSION.....	73
VIII.	Mandatory Notices (37 C.F.R. § 42.8(a)(1))	73

TABLE OF AUTHORITIES

Cases

<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH,</i> IPR2019-01469, Paper 6 (Feb. 13, 2020).....	73
<i>Cellco Partnership v. Huawei Device Co.,</i> IPR2020-01117, Paper 10 (Feb. 3, 2021).....	74
<i>Graham v. John Deere Co.,</i> 383 U.S. 1 (1966)	13
<i>KSR Int’l Co. v. Teleflex, Inc.,</i> 550 U.S. 398 (2007)	13
<i>Phillips v. AWH Corp.,</i> 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	12
<i>Titanium Metals Corp. of Am. v. Banner,</i> 778 F.2d 775, 782 (Fed. Cir. 1985)	36, 64, 65, 69, 70
<i>Wellman, Inc. v. Eastman Chem. Co.,</i> 642 F.3d 1355 (Fed. Cir. 2011)	12

Statutes

35 U.S.C. § 102(b)(pre-AIA)	2
35 U.S.C. § 103(a)(pre-AIA)	3
35 U.S.C. § 325(d)	72
35 U.S.C. §102(a)(1)(pre-AIA)	9

Rules

37 C.F.R. § 42.10(a).....	74
37 C.F.R. § 42.10(b)	74
37 C.F.R. § 42.8(a)(1)	73
37 C.F.R. § 42.8(b)(2).....	74
37 C.F.R. § 42.8(b)(3).....	74
37 C.F.R. §42.100(b)	11
37 C.F.R. § 42.104(a).....	1

EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 8,990,948 to Kumar <i>et al.</i>
1002	Prosecution history of the '948 Patent
1003	Expert Declaration of Seth James Nielson, Ph.D. under 37 C.F.R. § 1.68.
1004	<i>Curriculum Vitae</i> of Seth James Nielson, Ph.D.
1005	Biswanath Mukherjee, <i>Network Intrusion Detection</i>, IEEE Network, (June 1994).
1006	U.S. Patent No. 7,451,488 to Cooper <i>et al.</i>
1007	William R. Cheswick, <i>Firewalls and Internet Security Second Edition, Repelling the Wily Hacker 2</i>, (2003).
1008	Ross Anderson, <i>Network Attack and Defense</i>, Security Engineering: A Guide to Building Dependable Distributed Systems 2, 633-678, (2008).
1009	Andrew S. Tanenbaum, <i>Modern Operating Systems 3</i>, Pearson Prentice Hall, (2008).
1010	U.S. Patent Publication No. 2008/0175266 to Alperovitch <i>et al.</i>
1011	U.S. Patent Publication No. 2011/0145711 to Njemanze <i>et al.</i>
1012	U.S. Patent No. 5,604,862 to Midgely <i>et al.</i>

I. Introduction

Trend Micro Inc. (“Petitioner”) petitions for *inter partes* review of claims 1-5 of U.S. Patent No. 8,990,948 to Kumar *et al.*, entitled “SYSTEMS AND METHODS FOR ORCHESTRATING RUNTIME OPERATIONAL INTEGRITY.” (“the ’948 patent,” EX1001), which according to USPTO records is assigned to Taasera Licensing LLC (“Taasera”).

In fashion, often what’s old is new; however, this is never true for patents. In the case of the ’948 patent, the relevant prior art was old enough that it did not come up during prosecution. As will be shown, the basic concepts and details of the ’948 patent claims were public information long before its earliest priority date. Many years later, Patent Owner Taasera sought to patent essentially the same technology.

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

Petitioner certifies that the ’948 patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

III. Identification of Challenge (37 C.F.R. § 102(b))

A. Citation of Prior Art

Petitioner cites the following prior art references:

“Network Intrusion Detection” by Biswanath Mukherjee, L. Todd Heberlein, and Karl n. Levitt (“Mukherjee,” EX1005) is a research paper that published in 1994. Mukherjee is prior art under pre-AIA § 102(b) because it was published and publicly available in May/June of 1994, more than 1 year before the earliest possible priority date of the ’948 patent. Mukherjee was not disclosed to or considered during prosecution of the ’948 patent.

U.S. Patent No. 7,451,488 to COOPER *et al.* (“Cooper,” EX1006) is prior art under pre-AIA § 102(b) because it was published on January 13, 2005, more than 1 year before the earliest possible priority date of the ’948 patent. Cooper was not disclosed to or considered during prosecution of the ’948 patent.

A. Statutory Ground for Challenge (37 C.F.R. § 42.104(b))

Petitioner requests review of ’948 patent claims 1-5 (the “Challenged Claims”) on the following grounds:

GROUND 1: Claims 1-5 are unpatentable under pre-AIA 35 U.S.C. § 103(a) as obvious in view of Mukherjee (EX.1005) and Cooper (EX.1006).

The '948 patent

A. Claim of Priority of the '948 patent

The '948 patent issued on March 24, 2015, from U.S. Patent Application No. 13/559,707, filed July 27, 2012, which claimed priority to Provisional Application No. 61/641,007, filed May 1, 2012. EX1001, cover page. Solely for the purposes of this Petition, the earliest priority date for the '948 patent is taken to be May 1, 2012 (the “priority date”), so the '948 patent should be examined under pre-AIA statutes.

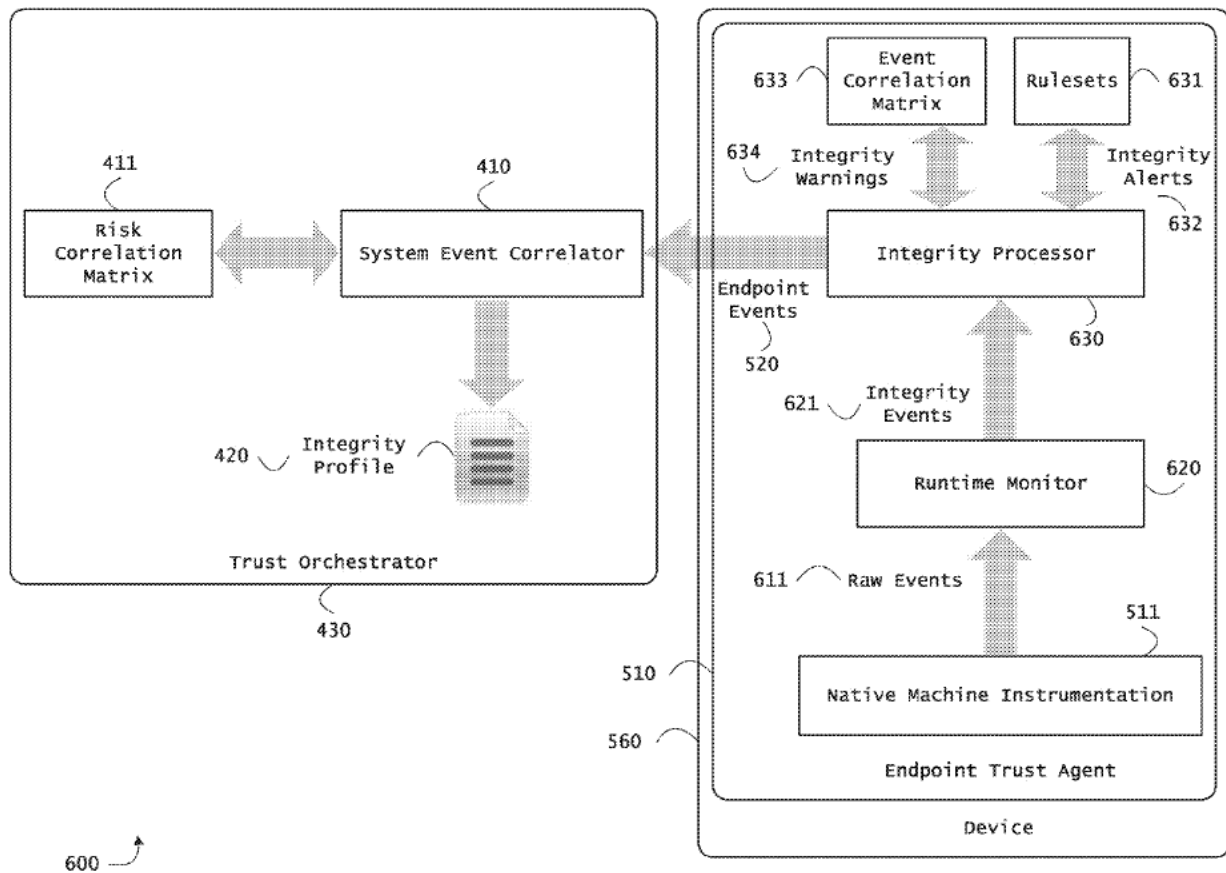
B. Overview of the '948 patent

The '948 patent is in the field of computer security. EX1001, 1:15-19. As background, the patent recognizes that “[t]he proliferation of applications (business, social networking and gaming software) in Enterprise and cloud computing ecosystems has significantly increased the attack surface and window of exposure,” and “[t]he proliferation of unmanaged devices such as mobile computing devices (i.e., tablets and smartphones) and Bring Your Own Device (BYOD) policies in Enterprise network environments has increased the risk of advanced coordinated threats.” *Id.*, 2:15-24.

To address this problem, the '948 patent discloses systems and method that provide or determine the operational integrity of a computing device and the applications running on the same. EX1001, 7:65-8:2.

An exemplary system or environment for the patent is shown in FIG. 6.

FIG. 6



EX1001, FIG. 6.

As shown in FIG. 6, the computing environment includes a device 560. The device 560 “may be a desktop, server, laptop, a mobile computing device such as a smartphone or tablet, or other types of mobile computing platforms that comprise (a) an operating system (OS); (b) a user operating the device to access a service or another device 560 over a network; and (c) application packages, applications or application components that may be downloaded and installed either with or without user intervention for later execution.” EX1001, 13:16-25. The device 560

includes native machine instrumentation, which can represent any event subscription, callback, notification mechanism provided by the supported operating system (OS) on the device 560. *Id.*, 13:43-47.

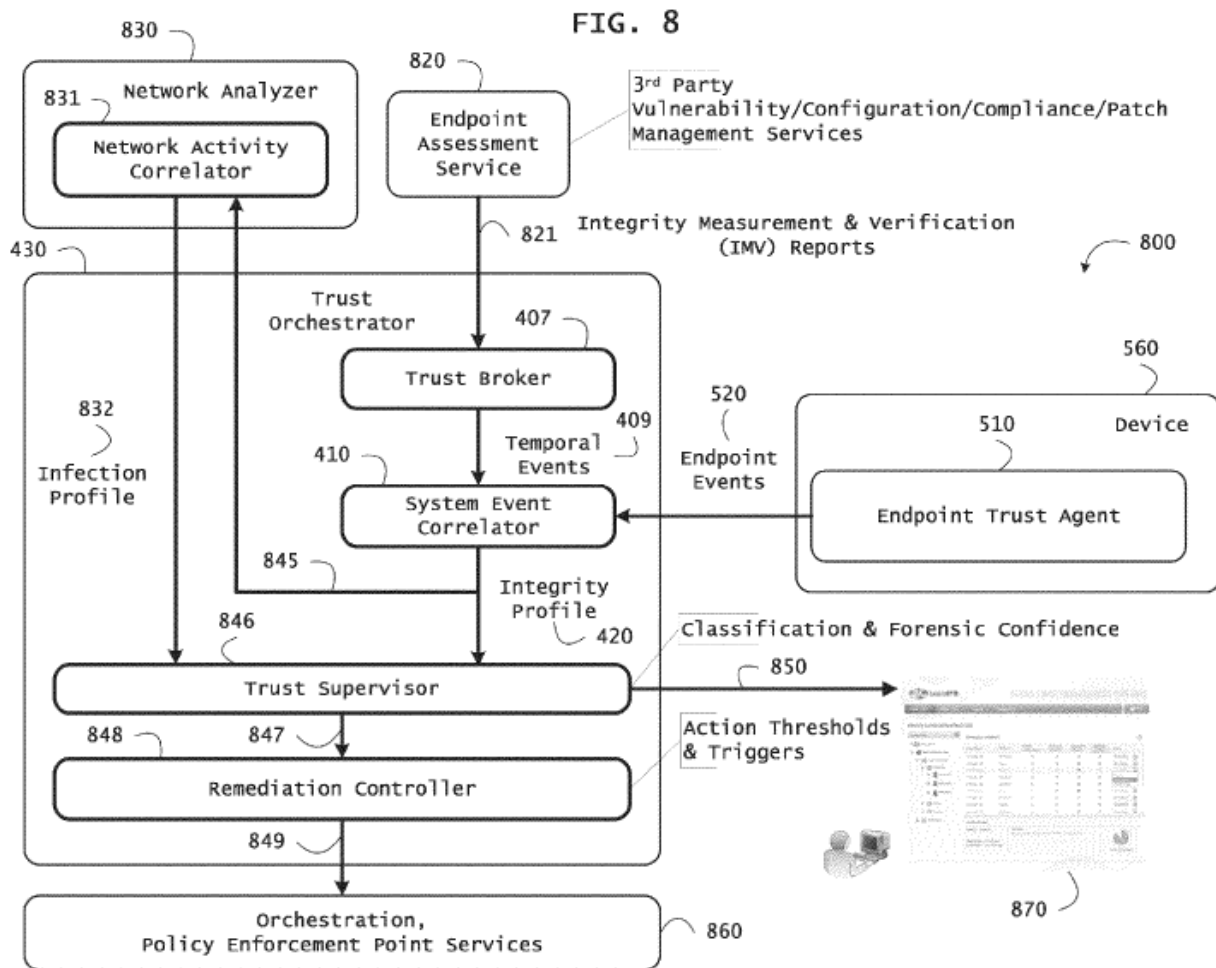
For determining the operational integrity of an application executing on the computing device, various operations, activity, or events associated with the application can be monitored and assessed. *Id.*, 9:39-48, FIGS. 1, 2, 3A, 5, 6. These operations, activities, or events—which may be sensed or detected by one or more sensors—include a runtime configuration of the application, resource utilization by the application, integrity of the application, network activities or dialogs of the application, and system and platform resources consumed by applications. *Id.*, 8:12-16, 9:3-28, FIGS. 1, 2, 5, 6.

As the various operations, activity, and events associated with the application occur, information for the events is generated and can be used for determining the operational integrity of the application. *Id.*, 10:39-44;13:1-15.

The computing environment of FIG. 6 also includes a runtime monitor 620, an integrity processor 630, rulesets 631, an event correlation matrix 633, a system event correlator 410, and a risk correlation matrix 411. *Id.* The runtime monitor 620 is configured to register for and receive a plurality of raw events 611 from native machine instrumentation 511 on the device 560.” *Id.*, 15:12-19. An integrity processor 630, which resides on the host or computer being monitored, receives

and processes information that may relate to the integrity of host computer, and generate warnings or alerts for the same. *Id.*, 15:15-19, 17:17-27, 30:45-52, FIG. 6. The integrity processor 630 may apply rulesets 631 to generate integrity alerts 632 and generate or populate an event correlation matrix 633. *Id.* The event correlation matrix 633 maps integrity warnings 634 to endpoint events 520. *Id.* The risk correlation matrix 411 correlates various events that may occur in the system to risks. *Id.*, 14:56-15:2.

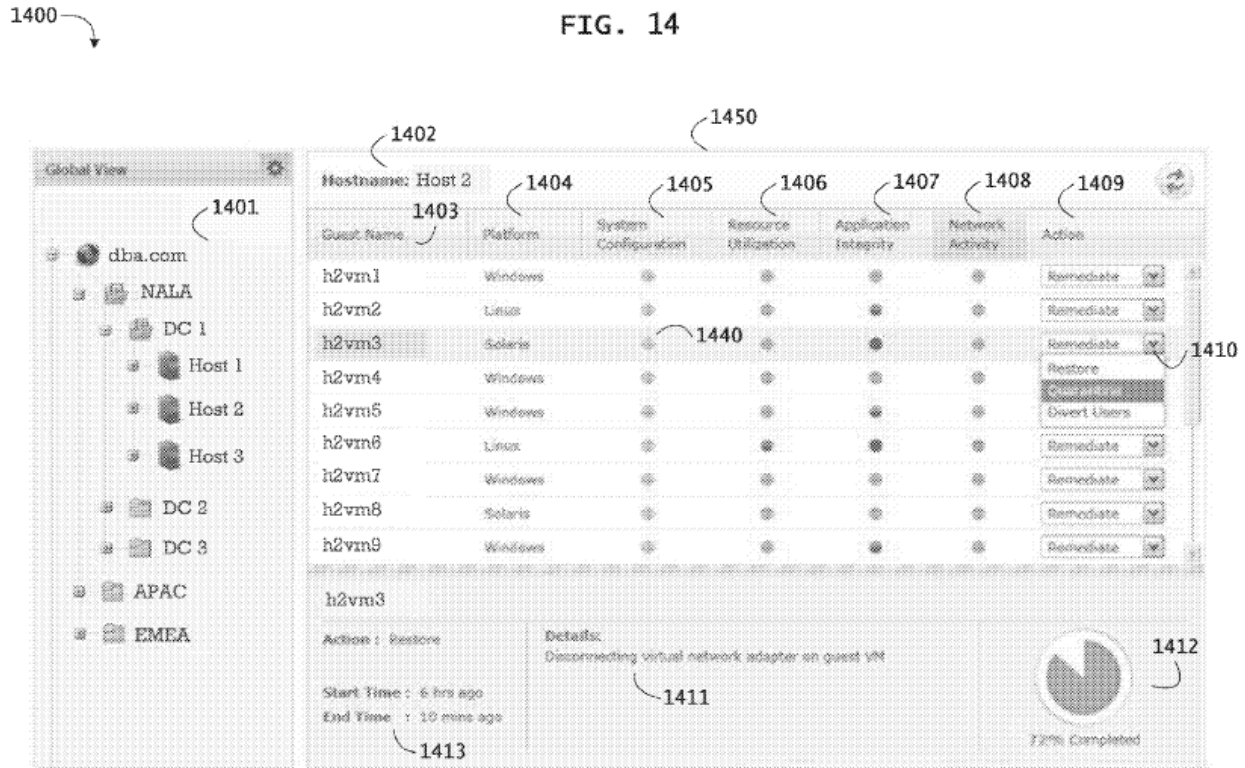
A further exemplary computing system or environment is shown in FIG. 8.



EX1001, FIG. 8.

The system or computing environment includes a network analyzer 830 and a trust supervisor 846. *Id.* The network analyzer 830 monitors all network traffic and activities of the device and provides information or forensic evidence of malicious network dialogs indicative of an infected victim or device. *Id.*, 20:40-47. The trust supervisor 846 receives information for events relating to the monitored device and network, and processes the same to identify threats or vulnerability that compromise or affect the trust that may be assigned. *Id.*, 22:31-37, FIGS. 6, 8, 10; EX1003, 81. In some embodiments, the trust supervisor correlates events associated to a device 560 for the classification and identification of threats based on the forensic confidence of leading indicators. EX1001, 20:44-53.

The status of the application's operational integrity, system configuration, resource, application integrity, and network activity is displayed in a graphical user interface (GUI) or dashboard for the administrator to view. *Id.*, 7:8-12, FIGS 14-18. An example GUI or dashboard is shown in FIG. 14:



EX1001, FIG. 14.

C. Summary of the Prosecution History

Prosecution began with a Restriction Office Action issued on April 11, 2014, to which Taasera responded by electing claims 5-9 for examination. EX1004, pp.196-202.

A Non-final Office Action on July 17, 2014 rejected claims 5-9 under 35 U.S.C. §102(a)(1)(pre-AIA) and §103(a)(pre-AIA) as unpatentable as obvious in view of three patent references.¹

¹ The Office Action cited U.S. Pub. Nos. 2008/0175266 to Alperovitch et al. and 2011/0145711 to Njemanze et al., and U.S. Pat. No. 5,604,862 to Midgely et al.

After an Examiner interview, Taasera amended the claims into the form that issued. Specifically, Taasera amended independent claim 5 as follows:

5. (Currently Amended) A method of providing real-time ~~runtime~~ operational integrity of an application on a native computing environment ~~and a system using a computing platform comprising a network trust agent, an endpoint trust agent, an event and behavior correlation engine, and a trust orchestration server~~, the method comprising:

monitoring, by a plurality of sensory inputs, one or more of network dialogs of the application, system operations initiated by the application, a runtime configuration of the application ~~activity, system configuration for a system the application is executing on~~, resource utilization by the application, and integrity of the application;

generating real-time behavior based events for determining the real-time operational integrity of the application executing on the native computing environment which includes a network analyzer, an integrity processor an event correlation matrix, a risk correlation matrix, and a trust supervisor;

correlating, by the event and risk correlation matrix, threat classifications ~~behavior correlation engine~~, risk based on the temporal sequence of the generated real- time behavior based events ~~monitoring~~; and

displaying, in a plurality of runtime dashboards of an administrative console of the computing environment ~~platform~~, real

~~time~~ real-time status indications for operational integrity of the ~~system~~
~~and~~ application.

Id., pp. 261-262.

Taasera argued that the cited prior art failed to disclose the limitations of “generating real-time behavior-based events for determining the real-time operational integrity of the application executing on the native computing environment which includes a network analyzer, an integrity processor, an event correlation matrix, a risk correlation matrix, and a trust supervisor.” *Id.*, p.268. Taasera also argued that the cited prior art failed to disclose “displaying, in a plurality of runtime dashboards of an administrative console of the computing environment, real-time status indications for operational integrity of the application.”

Thereafter, a Notice of Allowance issued on December 24, 2014, which included a statement of allowable subject matter. Specifically, the Examiner’s stated that none of the cited prior arts disclose the combination of limitations: **“monitoring, by a plurality of sensory inputs, one or more of network dialogs of the application, system operations initiated by the application, a runtime configuration of the application resource utilization by the application, and integrity of the application; generating real-time behavior based events for determining the real-time operational integrity of the application executing on the native computing environment which includes a network analyzer, an**

integrity processor, an event correlation matrix, a risk correlation matrix, and a trust supervisor; correlating, by the event and risks correlation matrix, threat classifications based on the temporal sequence of the generated real-time behavior based events.” *Id.*, pp. 289-290 (emphasis in the original). The patent issued on March 24, 2015.

D. Level of Ordinary Skill in the Art

As established by the Declaration of Dr. Seth Nielson dated April 6, 2023 (EX1003), a person having ordinary skill in the art (PHOSITA) at the time of the earliest claimed priority date for the ’948 patent would have had a bachelor’s degree in computer science, or a related field, or an equivalent technical degree or equivalent work experience, and an additional one to two years of education or experience in the field of computer security topics. More education can supplement practical experience and vice versa. EX1003, ¶¶88.

E. Claim Construction

In this proceeding, claims are interpreted under the same standard applied by Article III courts (i.e., the *Phillips* standard). 37 C.F.R. §42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

So far in parallel litigation neither Taasera nor any defendants have proposed a claim construction. For purposes of this proceeding only, Petitioner proposes that all claim terms should be given their plain and ordinary meanings and no

specific claim constructions are necessary to apply the asserted prior art to the Challenged Claims.² *See Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011) (“[C]laim terms need only be construed to the extent necessary to resolve the controversy.”).

B. HOW THE CLAIMS ARE UNPATENTABLE

The following discussion and the accompanying evidence demonstrate that there is a reasonable likelihood that Petitioner will prevail with respect to the Challenged Claims.

Petitioner evaluates the scope and content of the prior art, any differences between the art and the claim, and the knowledge of person of ordinary skill in the art in accordance with *Graham v. John Deere Co.*, 383 U.S. 1 (1966) and *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398 (2007). Petitioner is unaware of any secondary considerations relevant to the ’948 patent and a more than adequate written rationale is provided to support the combination of Mukherjee and Cooper under *KSR*. 550 U.S. at 418.

² Petitioner reserves the right in other proceedings to construe the Challenged Claims and pursue indefiniteness theories.

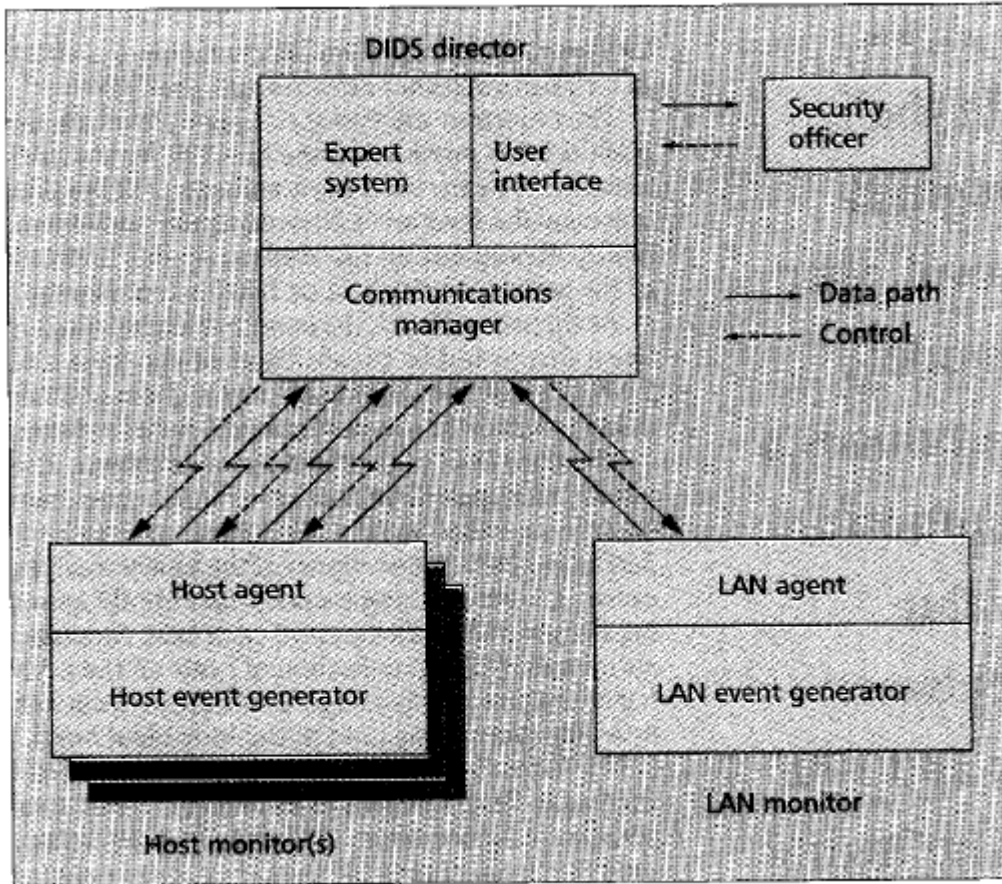
A. Overview of Asserted Prior Art

Petitioner submits that claims 1-5 of the '948 patent are invalid under as unpatentable under 35 U.S.C. § 103(a) in view of the combination of Mukherjee and Cooper.

i. Mukherjee

Mukherjee is directed towards the state-of-the-art of Intrusion Detection Systems in 1994. EX1003, ¶¶90; *see also id.*, ¶¶62-75 (citing EX1007-EX1012). In Mukherjee, “several host-based and network-based IDSs are surveyed, and the characteristics of the corresponding systems are identified.” EX1005, p.1. Mukherjee explains, “Intrusion detection systems (IDSs) are based on the beliefs that an intruder’s behavior will be noticeably different from that of a legitimate user and that many unauthorized actions are detectable. Typically, IDSs employ statistical anomaly and rule-based misuse models in order to detect intrusions.” *Id.* Typically, an IDS alerts a system security officer (SSO) when it detects an attack. *Id.* In Mukherjee, “several host-based and network-based IDSs are surveyed, and the characteristics of the corresponding systems are identified.” *Id.*

One of the IDS systems described in Mukherjee is a Detection and Alerting System (DIDS) that is part of a larger network security monitor (NSM) system. EX1005, pp.8,11, Figs. 1,2.



EX1005, Fig. 2.

Regarding DIDS, Mukherjee discloses:

The DIDS architecture combines distributed monitoring and data reduction with centralized data analysis. This approach is unique among current IDSs. The components of DIDS are the DIDS director, a single host monitor per host, and a single LAN monitor for each broadcast LAN segment in the monitored network. DIDS can potentially handle hosts without monitors since the LAN monitor can report on the network activities of such hosts. The host and LAN monitors are primarily responsible for the collection of evidence of

unauthorized or suspicious activity, while the DIDS director is primarily responsible for its evaluation.

EX1005, p.11.

In the DIDS architecture, each host in the monitored domain is equipped with a host monitor. The host and LAN monitors report to the DIDS director any "interesting" events that may possibly lead to intrusive activity. EX1005, p.12. The director employs an expert system (rule-based analysis system) to detect possible attacks. EX1005, pp.12-13. The architecture performs information aggregation so that even if the activity of a network-wide user may not be suspicious on a single host, the aggregate behavior may be suspect or unpermitted. EX1005, p.13. This architecture also performs "accountability" by tying users with their actions. *Id.*

The DIDS architecture is designed for a generalized distributed environment that is heterogeneous. EX1005, p.8,11. The network nodes can be host or servers from different vendors. *Id.* Activity and event reports are sent independently and asynchronously from the host and LAN monitors to the DIDS director through a communications infrastructure. EX1005, p.11, Fig.-1. The high-level communication protocols between the components of the DIDS system are based on International Organization for Standardization (ISO) standards, allowing for inclusion of other systems. EX1005, p.11. The architecture also provides for

bidirectional communication between the DIDS director and any monitor in the configuration. *Id.*

The host monitor consists of a host event generator (HEG) and a host agent. EX1005, p.12, Fig.-2. The HEG collects and analyzes audit records from the host's operating system and applications (e.g., rsh). EX1005, pp.11-12. The audit records are scanned for notable events, which are transactions that are of interest independent of any other records. *Id.* These include, among others, failed events, user authentications, changes to the security state of the system, and any network access such as rlogin and rsh. EX1005, p.12. These notable events are then sent to the director for further analysis. The HEG also tracks user sessions and reports anomalous behavior aggregated over time through user and group profiles, which is supported by the integration of the HAYSTACK intrusion detection algorithm into DIDS. EX1005, pp.12-15. The host agent handles all communications between the host monitor and the DIDS director. EX1005, p.12.

The LAN monitor consists of a LAN event generator (LEG) and a LAN agent. The LEG may operate as a subset of the NSM. EX1005, p.12. The LAN monitor's main responsibility is to observe all of the traffic on its segment of the LAN in order to monitor host-to-host connections, services used, and volume of traffic. *Id.* The LAN monitor reports on network activities such as rlogin and telnet

connections, the use of security-related services, and changes in network traffic patterns. EX1005, pp.12-13.

The DIDS director consists of three major components that are all located on the same dedicated workstation - a communications manager, an expert system, and a user interface. The communications manager is responsible for the transfer of data between the director and each of the host and the LAN monitors. The expert system uses rule-based logic for evaluating and reporting on the security state of the monitored system. The DIDS director receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a whole. The DIDS director's user interface provides the operator with interactive access to the entire system.

The expert system of the DIDS director in Mukherjee includes an intrusion detection model (IDM) that is used to analyze network data and other data to efficiently detect intrusions. EX1005, pp.12-13. This efficiency, in part, is derived from the multiple layers of abstraction that process network information before passing it onto a higher level. EX1003, ¶¶92-95. In this manner, raw audit records are converted to source agnostic record, which are converted to source-based (subject) records, which are converted to contextualized events, which are converted to threat-based combinations of events, which are converted to risk assessments per threat as a single value as the final, sixth layer/level. EX1005,

¶92. The processing within the IDM of Mukherjee is fundamental to threat analysis.

ii. Cooper

Cooper discloses methods for “scanning for vulnerabilities on a continuous basis and interpreting the resulting traffic in the context of policy.” EX1006, 1:21-25. Cooper discloses a network security system for vulnerability assessment using a policy-based monitor system that includes a vulnerability assessment client and a vulnerability assessment server. Together, the vulnerability assessment client and the vulnerability assessment server perform scans of objects and applications on the network to identify vulnerabilities. *Id.*, claim 1.

The vulnerability assessment client requests vulnerability scans, processes returned results, and stores relevant data. The vulnerability assessment client sends a request to the vulnerability assessment server for a vulnerability scan and receives the results of the scan from the server. *Id.*

The vulnerability assessment server receives the client request for vulnerability scans and performs the scans, returning the results of the scan to the vulnerability assessment client. *Id.* The results of the scan include information such as the IP address of the target host, the service being exercised, the type of vulnerability, and the security level of the vulnerability. *Id.* If an associated vulnerability state exists, vulnerability scan results are mapped into a vulnerability

network event. An enterprise-level user interface and an analyzing module makes the vulnerability network event accessible to a user. Cooper includes figures disclosing a plurality web pages of the user interface. EX1006, Figs.5-8.

The system disclosed in Cooper also includes a module for removing all vulnerability events related to a host that is unreachable for a specified period of time. *Id.* When a policy file is compiled, a compiler computes a complete set of IP addresses to be scanned and outputs this set of IP addresses to a file as input for the vulnerability assessment client. The system further includes a mapping function that maps vulnerability scans to vulnerability networks. *Id.*

Cooper explains the basics of vulnerability scans:

Vulnerability scanning technology is exemplified by commercial products such as the various scanners by Internet Security Systems, Inc. (ISS), the Cisco Scanner by Cisco Systems, Inc., and the Nessus Network Security Scanner. This technology examines the network configuration of hosts on the network by “probing” or “scanning” them using network traffic that is crafted to elicit a response. The response is interpreted and used to determine the configuration of the host that is scanned. Using various probing techniques, the technology is capable of:

scanning a range of Internet addresses to determine which addresses likely represent working hosts and which do not;

scanning a range of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) port numbers within a given host to determine which applications are likely active on a given working host; and

probing an application on a given host to determine whether it is susceptible to previously known attacks, known as “vulnerabilities.”

EX1006, 1:55-64.

Cooper discloses improvements to vulnerability scanning technology that solve some identified problems. In an improvement, Cooper teaches that “[u]sing the network security policy's information, the output of the scanner is filtered, so that redundant information is removed about applications that are sanctioned by policy.” EX1006, 2:56-63.

Cooper discloses that “the policy monitoring system is provided, through said network security policy, with a description of the intended action of the vulnerability scanner. This permits the monitoring system to report the scanner's operation as a normal usage pattern of the network, eliminating the confusion that can result when a vulnerability scan is monitored.” EX1006, 2:64-3:2.

Cooper also discloses that the monitoring performed can be continuous.

In one embodiment of the invention, vulnerability state information of a system can be derived from repeated events, referred to herein as continuous scanning. FIG. 3 is a schematic diagram showing a

workflow of scanned vulnerability states, which are stored in a repository, and emitted as monitored events according to the invention. New vulnerability data is entered into the system and the output is a monitored event. (EX1006, 9:56-64).

B. Motivation to Combine Mukherjee and Cooper

A PHOSITA would have been motivated to combine the teachings of Mukherjee and Cooper for the following reasons.

A PHOSITA would have been motivated to combine Mukherjee and Cooper in part because both systems are directed towards detecting and resolving vulnerabilities in network systems. EX1003, ¶¶108-109; EX1006, 2:45-65; EX1005, pp.8-9. Mukherjee discloses “Intrusion detection systems (IDSs) [that] are based on the beliefs that an intruder’s behavior will be noticeably different from that of a legitimate user and that many unauthorized actions are detectable.” EX1005, p.1. Mukherjee explains that IDS systems may incorporate “rules based on... known system vulnerabilities” for detecting intrusive behavior, and that IDS systems are best used with other tools to enhance their capabilities. EX1005, p.4,12. Similarly, Cooper uses network monitoring technology to analyze vulnerabilities on a networked computer system, including “scanning for vulnerabilities on a continuous basis and interpreting the resulting traffic in the context of policy.” EX1006, 1:21-25; EX1003, ¶109. Cooper explains that

traditional vulnerability scanners, that are simultaneously monitoring the network will likely be detected as an attacker, whereas the vulnerability scanner of Cooper is adapted to work with network monitoring technology, which a PHOSITA would have understood to include an IDS like that of Mukherjee.EX1003, ¶109-112; EX1006, 2:20-30,2:64-3:2.

In particular, a PHOSITA would have been motivated to combine Cooper's vulnerability scans, which are designed to operate effectively with network monitoring technology, into Mukherjee's network scanning IDS's. EX1003, ¶110. Mukherjee explains that:

The DIDS architecture combines distributed monitoring and data reduction with centralized data analysis. This approach is unique among current IDSs. The components of DIDS are the DIDS director, a single host monitor per host, and a single LAN monitor for each broadcast LAN segment in the monitored network. DIDS can potentially handle hosts without monitors since the LAN monitor can report on the network activities of such hosts. The host and LAN monitors are primarily responsible for the collection of evidence of unauthorized or suspicious activity, while the DIDS director is primarily responsible for its evaluation. EX1005, p. 11 (emphasis added).

Mukherjee also explains that IDSs, such as DIDS, are best used with other tools in combination.

It is anticipated that a growing set of tools, including incident-handling tools and network-management tools, will be used in conjunction with the intrusion-detection functions of DIDS. This will give the SSO the ability to actively respond to attacks against the system in real time. Incident-handling tools may consist of possible courses of action to take against an attacker, such as cutting off network access, a directed investigation of a particular user, removal of system access, etc. Network-management tools that are able to perform network mapping would also be useful.

EX1005, p.12 (emphasis added).

The vulnerability scanner of Cooper is designed to work with a monitoring system similar to an IDS and regards this integration as part of the advantages of its teachings. A PHOSITA would have recognized that a “vulnerability scanner” disclosed in Cooper is an example of a “network-management tool” disclosed in Mukherjee. EX1003, ¶¶112-113. Thus, a PHOSITA reviewing Mukherjee and Cooper would have recognized that they disclose being compatible with their respective types of systems as a part of their design. EX1003, ¶¶113-114.

Mukherjee explains that “known system vulnerabilities” (such as those identified by a vulnerability scanner) were already used in IDS systems to identify intrusive behavior. EX1003, ¶¶114-115. A PHOSITA would have been motivated to include vulnerability assessment data from the vulnerability scanner of Cooper in Mukherjee’s DIDS system as described in Mukherjee and Cooper. EX1003, 115-118; EX1005, p.13; EX1006, 7:64-8:6. For example, Mukherjee, in defining a

six-layer analysis system, explains that the fourth layer is for putting “the event in context.” EX1005, p.13. Mukherjee describes that one kind of context is “relative importance of the source of events. *Id.* That is, events related to a particular user, or events from a particular host, maybe more likely to represent an intrusion...”

Id. A PHOSITA would have known that vulnerability information provides context because network traffic of a certain form to a vulnerable host is more likely to be an intrusion than similar network traffic to a host that does not have that specific vulnerability. EX1003, ¶115. A PHOSITA would have been motivated to include the vulnerability information of Cooper in Mukherjee to better identity context for events. *Id.*

Mukherjee also explains that the fifth layer of their intrusion model uses “[e]vents in context [combined] to create threats.” EX1005, p. 38. This layer includes “[s]uspicious acts [that] are events which, while not a violation of policy, are of interest to an IDS.” *Id.* A PHOSITA would have known that network traffic to a vulnerable host is of more interest to the IDS than traffic to a non-vulnerable host. EX1003, ¶115. Therefore, a PHOSITA would have been motivated to include the vulnerability scans of Cooper to improve the identification of threats. *Id.*

In addition to using vulnerability information to better contextualize and process events, a PHOSITA would have been motivated to use Cooper’s

“vulnerability events” in the IDS of Mukherjee as events. *Id.*, ¶116. Mukherjee’s first level of the 6-level model receives “audit records provided by the host OS, by the LAN monitor, or by any third-party auditing package.” EX1005, p.13. Mukherjee identifies system events (i.e., identified by records from the host OS) and network events (i.e., identified by the LAN monitor). Cooper discloses another kind of event: “[t]he scanner reports all application-level vulnerabilities as vulnerability events.” EX1006, at 4:32-40 (emphasis added). Further, Cooper discloses that “[i]n most respects, vulnerability events are handled in a similar fashion to network events.” EX1006, 7:64-8:6. Cooper explains that a “vulnerability may also be mapped into a network event” and explains in detail how that can be accomplished. EX1006, 12:19-13:8.

A PHOSITA would have understood this disclosure of Cooper to mean that a vulnerability event can be translated into a type of network event. EX1003, ¶¶116-117. This conclusion is supported by Cooper’s disclosure that “[v]ulnerability events are viewable through the enterprise manager system user interface. They are indistinguishable from other network events.” EX1006, 15:20-23. Further, Mukherjee discloses using data from “any third-party auditing package.” EX1005, p.13. Therefore, a PHOSITA would have understood that vulnerability events of Cooper could be used in Mukherjee’s first level of analysis either as a new kind of event, or as network events. EX1003, ¶118. A PHOSITA

would have been motivated to include this vulnerability event information to more accurately identify intrusion. EX1003, ¶¶117-119. Specifically, Cooper explains that “the invention provides policy-based vulnerability scanning to minimize false positives.” EX1005, 4:9-16.

Based on the roadmaps in both Mukherjee and Cooper, a PHOSITA would have found it obvious to expand on Mukherjee’s DIDS using Cooper’s vulnerability scanning disclosures, and would have recognized that this combination could be achieved with minimal experimentation with a high likelihood of success. EX1003, ¶120.

C. Comparison of Mukherjee and Cooper to the Challenged Claims

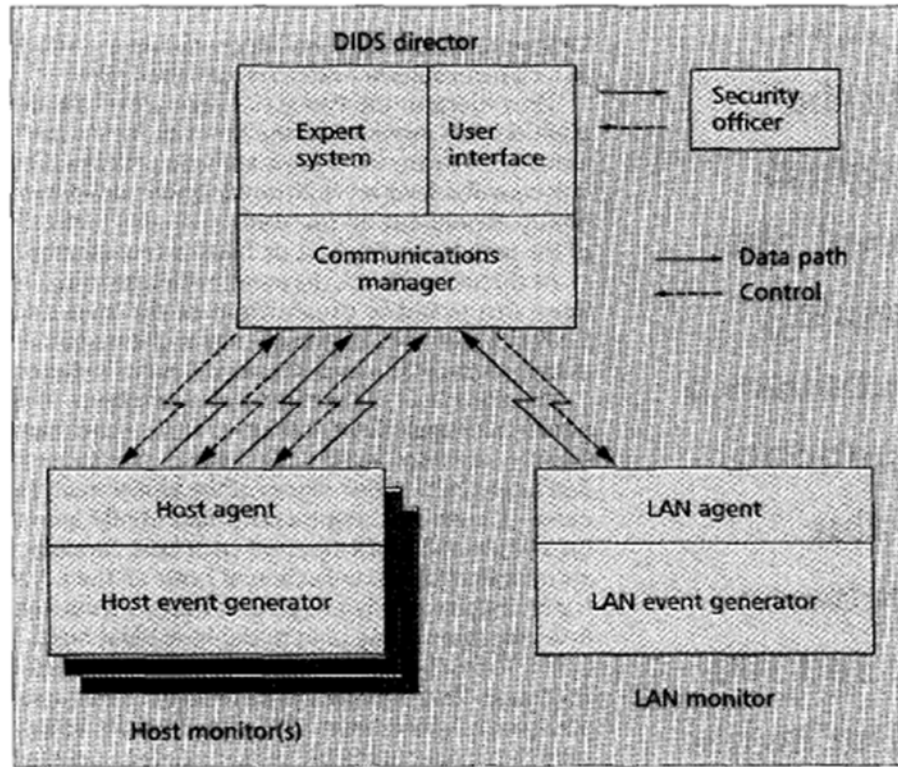
1. Claim 1

1[a].A method of providing real-time operational integrity of an application on a native computing environment, the method comprising:

Mukherjee in view of Cooper teaches or suggests a “method of providing real-time operational integrity of an application on a native computing environment.” EX1003, 125.

Mukherjee discloses rsh, rlogin, and other “running processes” as well as a host’s operating system running on a host computer that is part of a computer network, such as a local area network. EX1005, 12, Figs. 1, 2. A PHOSITA would

have understood that “rlogin” and “rsh” are each an application executing in a “computing environment.” EX1005, 12; EX1003, ¶133. Mukherjee explains that a host monitor in the DIDS architecture “collects and analyzes audit records from the host’s operating system.” EX1005, p. 36.



■ **Figure 2.** *Communications architecture.*

A PHOSITA would have understood events from the host operating system to be “native” rather than non-native events. EX1003, ¶127. This is consistent with the specification of the ’948 patent, which explains that “the native machine instrumentation 511 can represent any event subscription, callback, notification mechanism provided by the supported operating system (OS) on the device 560.” EX1001, 13:43-56 (emphasis added).

Mukherjee teaches or suggests “providing...operational integrity of an application” because it describes the host monitor and an expert system in the DIDS architecture for monitoring and detecting intrusions, threats, and attacks at the host computer. EX1005, pp.11-12. For example, Mukherjee explains that the host monitor includes host event generator (HEG), which. collects and analyzes audit records from the host's operating system and applications (e.g., rsh). *Id.*, pp.11-12, Fig.-2. The audit records are scanned for notable events, which are transactions that are of interest independent of any other records. *Id.* These include, among others, failed events, user authentications, changes to the security state of the system, and any network access such as rlogin and rsh. *Id.*, p.12. These notable events are then sent to the DIDS director for further analysis. The HEG also tracks user sessions and reports anomalous behavior aggregated over time through user and group profiles, which is supported by the integration of the HAYSTACK intrusion detection algorithm into DIDS. *Id.*, pp.12-15. A LAN monitor observes all of the traffic on its segment of the LAN and reports on network activities such as rlogin and telnet connections, the use of security-related services, and changes in network traffic patterns. *Id.*, pp.12-13.

The expert system uses rule-based logic for evaluating and reporting on the security state of the monitored system. The DIDS director receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences

about the security of each individual host, as well as the system as a whole. The DIDS director's user interface provides the operator with interactive access to the entire system. *Id.*, p.12. The expert system of the DIDS director in Mukherjee includes an intrusion detection model (IDM) that is used to analyze network data and other data to efficiently detect intrusions. EX1005, pp.12-13. The expert system “considers the *threats* to the network and the hosts connected to it.” *Id.*, p. 12. “The threats are partitioned by the nature of the abuse and the nature of the target,” which may include active objects that “are essentially running processes.” *Id.*

Mukherjee teaches or suggests that its monitoring and detecting intrusions, threats, and attacks (“providing...operational integrity”) is in “real-time.”

Mukherjee explains that:

The expert system is responsible for evaluating and reporting on the security state of the monitored system. It receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a whole... The director’s user interface allows the SSO interactive access to the entire system. The SSO can watch activities on each host, watch network traffic (by setting “wiretaps”), and request more specific types of information from the monitors.

EX1005, p.12 (emphasis added).

A PHOSITA would have recognized that “watching activities on each host... [and] network traffic” refers to monitoring and provisioning in “real-time,” especially when using words such as “wiretaps.” EX1003, ¶127. As Mukherjee further explains, the use of “incident-handling tools and network-management tools,” such as those disclosed in Cooper, “will give the SSO the ability to actively respond to attacks against the system in real time.” EX1005, p.12 (emphasis added).

As discussed in more detail below, Cooper uses methods and systems similar to those in Mukherjee to perform vulnerability scans of applications to determine a baseline security of an application and using runtime monitoring of the process, network, and system to determine whether applications are compromised. EX1003, ¶130. A PHOSITA would have been motivated to combine Cooper’s analysis of application-specific vulnerabilities with Mukherjee’s analysis of host and network events as a PHOSITA would have been motivated to use Cooper’s per-application vulnerability information to more accurately identify attacks and determine if an application is compromised as described above. EX1003, ¶¶131-133, 140-144.

Thus, Mukherjee in view of Cooper discloses methods of providing real-time operational integrity of an application in native computing environments on a

network (e.g., LAN). EX1003, ¶¶150-153. For these reasons, Mukherjee in view of Cooper teaches or suggests limitation 1[a]. EX1003, ¶153.

1[b]. monitoring, by a plurality of sensory inputs, one or more of network dialogs of the application, system operations initiated by the application, a runtime configuration of the application, resource utilization by the application, and integrity of the application;

Mukherjee in view of Cooper discloses limitation 1[b].

The system traffic monitoring processes disclosed in Mukherjee provide methods that enable the application monitoring disclosed in Cooper. EX1003, ¶¶117-122. Mukherjee discloses “monitoring by a plurality of sensory inputs” by disclosing both a “host monitor” and a “network monitor.” EX1003, ¶¶156-159.

The components of DIDS are the DIDS director, a single host monitor per host, and a single LAN monitor for each broadcast LAN segment in the monitored network.

EX1005, p.11 (emphasis added).

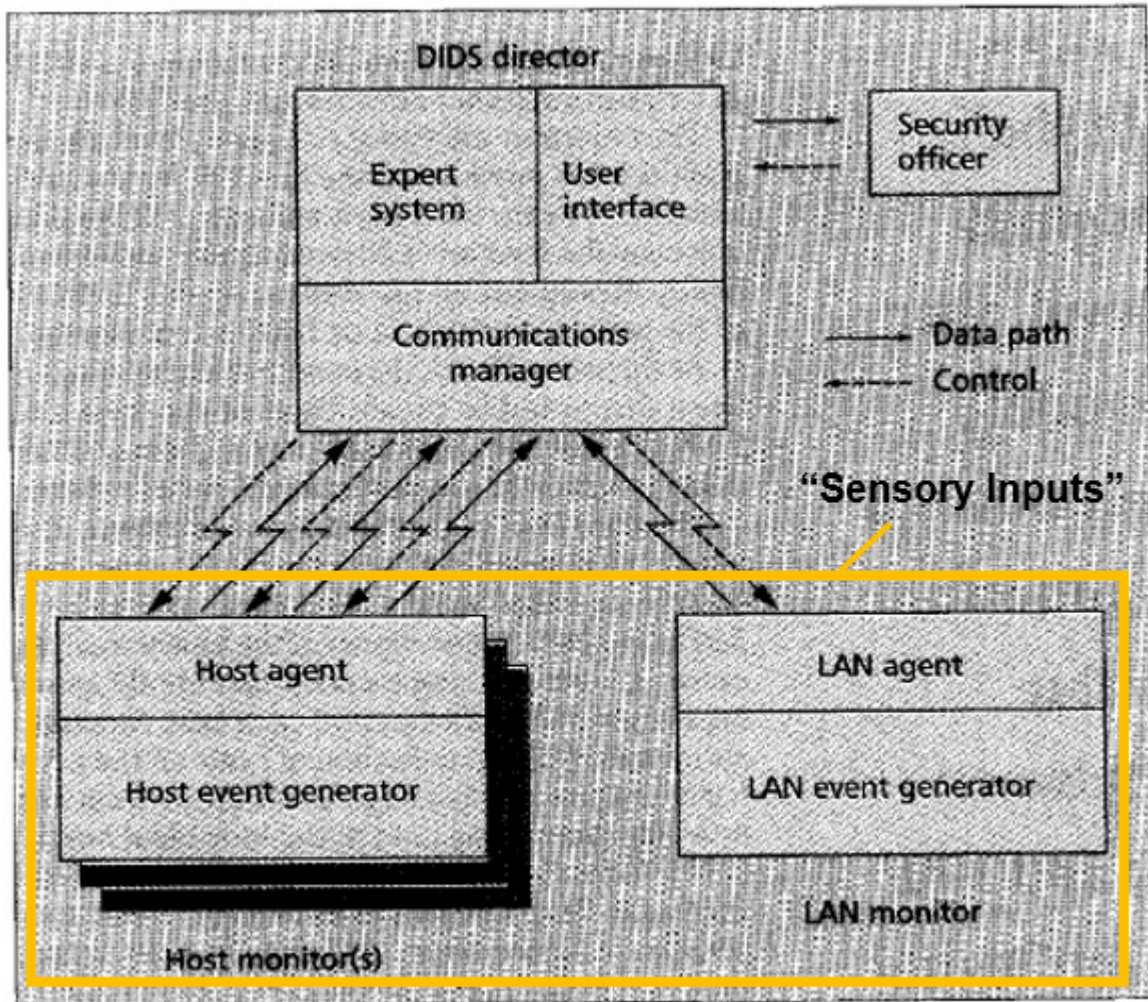
The host monitor consists of a host event generator (HEG) and a host agent. The HEG collects and analyzes audit records from the host’s operating system. The audit records are scanned for notable events.

Id., p.11 (emphasis added).

Like the host monitor, the LAN monitor consists of a LAN event generator (LEG) and a LAN agent. The LEG is currently a subset of the NSM. Its main responsibility is to observe all of the traffic on its segment of the LAN in order to monitor host-to-host connections,

services used, and volume of traffic. The LAN monitor reports on network activities such as rlogin and telnet connections, the use of security-related services, and changes in network traffic patterns.

Id., p.12 (emphasis added; internal citations deleted).



EX1005, Fig. 2 Annotated.

A PHOSITA would have understood the HEG and LAN monitors to be “monitoring by a plurality of sensory inputs” as they each sense one or more operations or activities in the host device and environment, consistent with the ’948 patent’s disclosure. EX1003, ¶157.

Mukherjee describes that the LAN monitor performs monitoring of “traffic on its segment of the LAN in order to monitor host-to-host connections, services used, and volume of traffic (corresponding to “one or more of network dialogs of the application”). EX1005, p. 12.

This excerpt also shows that Mukherjee discloses monitoring network dialogs (e.g., sessions, authentications, network access) associated with various applications (e.g., rsh). EX1003, ¶¶159-161. Mukherjee further explains that “[t]he SSO may ‘tag’ particular security ‘subjects’ and ‘objects’ for special monitoring. This is analogous to setting an alarm to go off when a... program is accessed” and discloses monitoring “active objects [that] are essentially running processes.” EX1003, p.5. A PHOSITA would have recognized that the audit records include application-level (i.e., process-level) audit data. EX1003, ¶164; EX1005, p.13. Thus, Mukherjee teaches monitoring “system operations initiated by the application” such as network traffic and OS audit records actions of software components (e.g., rsh). EX1003, ¶¶164-167.

While the disclosures in Mukherjee teach monitoring network traffic and network dialogs that would be initiated by applications, Cooper discloses using such monitoring methods for monitoring applications. EX1003, ¶141. Specifically, Cooper discloses that a vulnerability scanner connects to an

application on a specific network port and determines from the network traffic whether the application is vulnerable.

The VA Client receives application-level vulnerability results through the TCP socket connection it maintains with the VA Server.

Each vulnerability result indicates the IP address of the target host, the service being exercised, e.g. SSH, the type of vulnerability found, and the severity level of the vulnerability.

EX1006, 12:9-14.

[F]or each vulnerability seen, the vulnerability may also be mapped into a network event as follows: ... The application protocol to which the vulnerability applies, e.g. http, is assigned to the service.... The port where the vulnerability was found is assigned to the destination port.

Id., 12:19-40 (emphasis added).

Thus, the combination of Mukherjee and Cooper discloses monitoring of “network dialogs of the application, ... resource utilization by the application, and integrity of the application. EX1003, ¶¶162-167.

As discussed previously, a PHOSITA would have been motivated to combine the application monitoring of Cooper with the system of Mukherjee because the vulnerability assessment and vulnerability data provided by Cooper would be useful to Mukherjee’s expert system for identifying intrusive behavior. EX1003, ¶114. Further, the network traffic monitoring methods of Mukherjee disclose the systems and methods useful for performing the application-level

vulnerability monitoring of Cooper, thus providing the PHOSITA with an expectation of successfully combining the disclosures. EX1003, ¶162.

Accordingly, Mukherjee and Cooper teach at least two of the alternative limitations listed in 1[b], any one of which is sufficient to render the entire limitation obvious. *See e.g., Titanium Metals Corp. of Am. v. Banner*, 778 F.2d 775,782 (Fed. Cir. 1985) (“It is also an elementary principle of patent law that when...a claim covers several compositions, the claim is "anticipated" if one of them is in the prior art.”) .

For these reasons, Mukherjee or Mukherjee in view of Cooper discloses limitation 1[b]. EX1003, ¶168.

1[c]. generating real-time behavior based events for determining the real-time operational integrity of the application executing on the native computing environment which includes a network analyzer, an integrity processor, an event correlation matrix, a risk correlation matrix, and a trust supervisor;

As discussed above for limitations 1[a] and [1b], Mukherjee in view of Cooper discloses or teaches “real-time operational integrity of the application executing on the native computing environment” because Mukherjee describes monitoring and detecting intrusions, threats, and attacks (“real-time operational integrity”) for running process on the host computer and Cooper extends those methods to determining the integrity or vulnerability of applications executing on the host computer (“application executing on the native computing environment”).

Mukherjee discloses “generating real-time behavior based events for determining” these intrusions, threats, and attacks (corresponding or relating to “real-time operational integrity”). EX1003, ¶171. As Mukherjee explains, “In DIDS, the host and LAN monitors report any ‘interesting’ events, which may possibly lead to intrusive activity, to a centrally-located DIDS director.” EX1005, p.11. These events are detected on hosts where the events “include, among others, failed events, user authentications, changes to the security state of the system, and any network access such as rlogin and rsh.” EX1005, p.12. Events also include “activities such as rlogin and telnet connections, the use of security-related services, and changes in network traffic patterns” *Id.* A PHOSITA would have recognized all of these to be “behavior-based events” because each involves some form of user behavior in the network or access of applications. EX1003, ¶173. Further, because Mukherjee discloses that the SSO is given the ability to “actively respond to attacks against the system in real time,” a PHOSITA would understand the events for determining such attacks would likewise be generated in “real time.” EX1005, p.12 (emphasis added); EX1003, 172.

Cooper further discloses “generating real-time behavior based events for determining the real-time operational integrity of the application executing on the native computing environment.” Cooper discloses using network scanning by a network analyzer (e.g., provided in Mukherjee) to create inventories of devices that

can be attacked, determining which network ports (e.g., TCP and UDP ports) are in use, and probing them over the network to determine application-level vulnerabilities. *Id.*, ¶146.

Vulnerability scanning technology is exemplified by commercial products such as the various scanners by Internet Security Systems, Inc. (ISS), the Cisco Scanner by Cisco Systems, Inc., and the Nessus Network Security Scanner. This technology examines the network configuration of hosts on the network by “probing” or “scanning” them using network traffic that is crafted to elicit a response. The response is interpreted and used to determine the configuration of the host that is scanned. Using various probing techniques, the technology is capable of:

scanning a range of Internet addresses to determine which addresses likely represent working hosts and which do not;

scanning a range of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) port numbers within a given host to determine which applications are likely active on a given working host; and

probing an application on a given host to determine whether it is susceptible to previously known attacks, known as “vulnerabilities.”

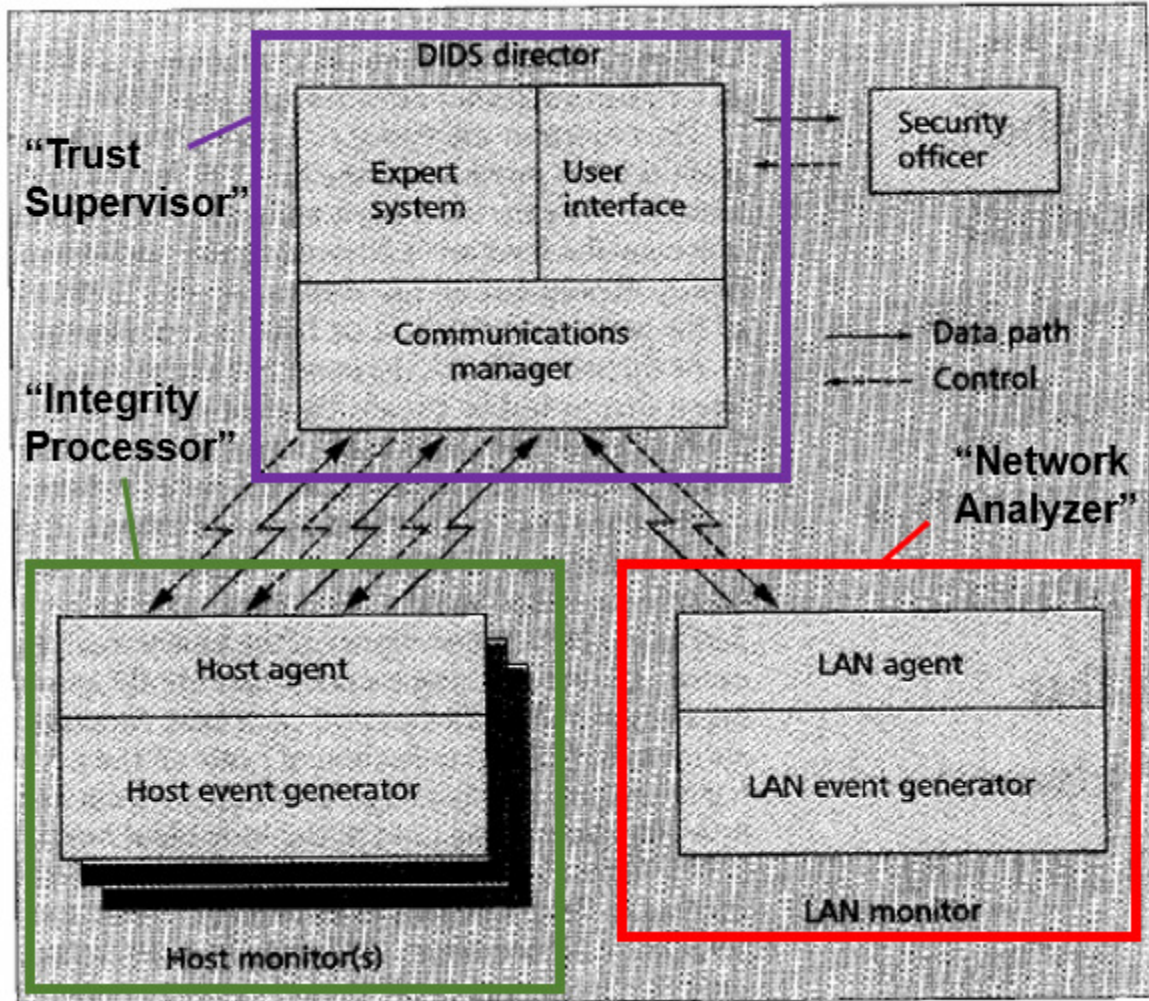
EX1006, 1:55-64 (emphasis added).

As Cooper explains in this citation, the scanning of the vulnerability scanner uses network traffic to elicit a response and vulnerabilities are inferred from the

response. EX1003, ¶147. Cooper further explains that “[t]he scanner reports all application-level vulnerabilities as vulnerability events.” EX1006, 4:32-40.

A PHOSITA would have understood these vulnerability events to be “generat[ed] real-time behavior based events for determining the real-time operational integrity of the application executing in the native computing environment” because the events are generated for the application on the OS by the vulnerability scanner in real-time based on the responses to identify risks to the application that could be exploited. EX1003, ¶147.

Mukherjee discloses a computer network (corresponding to the “native computing environment”) consistent with “a network analyzer, an integrity processor, an event correlation matrix, a risk correlation matrix, and a trust supervisor.” EX1005pp. 12-13; EX1003 ¶¶172-194.



EX1005, Fig. 2 Annotated.

Mukherjee teaches "a network analyzer" in the form of a LAN monitor that monitors networks activity and processes or reports network-based events.

EX1003, ¶179.

In DIDS, the host and LAN monitors report any "interesting" events, which may possibly lead to intrusive activity, to a centrally-located DIDS director.

EX1005, p.11 (emphasis added).

The components of DIDS are the DIDS director, a single host monitor per host, and a single LAN monitor for each broadcast LAN segment in the monitored network.

EX1005, p.11 (emphasis added).

Like the host monitor, the LAN monitor consists of a LAN event generator (LEG) and a LAN agent. The LEG is currently a subset of the NSM [17,27, 29]. Its main responsibility is to observe all of the traffic on its segment of the LAN in order to monitor host-to-host connections services used, and volume of traffic. The LAN monitor reports on network activities such as rlogin and telnet connections, the use of security-related services, and changes in network traffic patterns.

Id., p.12 (emphasis added).

...each layer representing the result of a transformation performed on the data [e.g., from a lower level]. The objects at the first (i.e., lowest) level are the audit records provided by the host OS, by the LAN monitor, or by any third-party auditing package. The objects at this level are both syntactically and semantically dependent on the source. At this level, all of the activity on the host or LAN is represented. At the second level, the event (which has already been discussed in the context of the host and LAN monitor) is both syntactically and semantically independent of the source standard format for events.

Id., p.13 (emphasis added).

A PHOSITA would have recognized the LAN monitor as including a “network analyzer” based on the literal term and descriptions of the operations performed by the LAN Monitor described in Mukherjee. EX1003, ¶180.

Mukherjee's disclosure for its LAN monitor is consistent with the '948 patent's disclosure for "network analyzer." EX1003, ¶180; EX1001, 20:40-47, FIG. 8.

Mukherjee teaches "an integrity processor" in disclosing a host monitor having a host event generator (HEG). The HEG resides on the host computer being monitored. EX1005, Fig. 2. Mukherjee describes that the HEG collects and processes information relating to operations and activity on the host computer, including for suspicious activity ("integrity"), and generates system events for reporting out the same:

In DIDS, the host and LAN monitors report any "interesting" events, which may possibly lead to intrusive activity, to a centrally-located DIDS director.

...

The components of DIDS are the DIDS director, a single host monitor per host, and a single LAN monitor for each broadcast LAN segment in the monitored network.

...

The host monitor consists of a host event generator (HEG) and a host agent. The HEG collects and analyzes audit records from the host's operating system. The audit records are scanned for notable events, which are transactions that are of interest independent of any other records. These include, among other things, failed events, user authentications, changes to the security state of the system, and any

network access such as rlogin and rsh. These notable events are then sent to the director for further analysis.

[The DIDS system] considers the threats to the network and the hosts connected to it. Events in context are combined to create threats. The threats are partitioned by the nature of the abuse and the nature of the target. In other words, what is the intruder doing, and what is he doing it to? Abuses are divided into attacks, misuses, and suspicious acts. Attacks represent abuses in which the state of the machine is changed. That is, the file system or process state is different after the attack than it was prior to the attack.

EX1005, p. 12-13 (emphasis added).

A PHOSITA would have known that a different “process state... after the attack” could indicate applications (such as rlogin and rsh) no longer have operational integrity because an attack has changed them, and changes to the file system because of an attack would indicate that the operating system no longer has operational integrity. EX1003, ¶¶176-177.

A PHOSITA would have recognized Mukherjee’s HEG as including a “integrity processor” based on the literal term and descriptions of the operations performed by the HEG described in Mukherjee. EX1003, ¶185. Mukherjee’s disclosure for its HEG is consistent with the ’948 patent’s disclosure for “integrity processor.” EX1003, 185; EX1001, 15:15-19, 17:17-27, 30:45-52, FIG. 6.

Mukherjee teaches “a trust supervisor” in disclosing an expert system in the DIDS director. EX1003, ¶194. Mukherjee explains that the DIDS director’s expert

system, which itself uses rules from an Intrusion Detection Model (IDM), receives reports from the host and LAN monitors and, from them, makes the inferences about system security to identify threats.

The expert system is responsible for evaluating and reporting on the security state of the monitored system. It receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a whole... The director's user interface allows the SSO interactive access to the entire system. The SSO can watch activities on each host, watch network traffic (by setting "wire taps"), and request more specific types of information from the monitors.

...

DIDS utilizes a rule-based (or production) expert system written in CLIPS. The expert system uses rules derived from a hierarchical Intrusion Detection Model (IDM). The IDM describes the transformation from the distributed raw audit data to high-level hypotheses about intrusions and about overall security of the monitored environment. The IDM consists of six layers, with each layer representing the result of a transformation performed on the data.

...

The fifth layer of the [expert system's] model considers the *threats* to the network and the hosts connected to it. Events in context are combined to create threats. The threats are partitioned by the nature of the abuse and the nature of the target. In other words, what is the

intruder doing, and what is he doing it to? Abuses are divided into *attacks, misuses, and suspicious acts*.

EX1005, pp.12-13 (emphasis added and original).

Mukherjee's description of its expert system is consistent with the descriptions of the '948 patent for the "trust supervisor." EX1003, ¶¶193-194; EX1001, 18:21-31,22:31-37, FIGS. 6, 8, 10. Therefore, a PHOSITA would have understood the Mukherjee's expert model of the DIDS director is a "trust supervisor." *Id.*

Mukherjee also discloses "a risk correlation matrix." The specification of the '948 patent explains that a "risk correlation matrix" can be embodied as "grids that represent an exemplary dynamic model of measurement and identification based on clustering and classification of independent endpoint events (alerts) to generate system warnings that may be mapped to warning categories or classes." EX1001, 12:48-52 (emphasis added). The '948 patent also discloses that "system warnings and threat classification [are] produced by the risk correlation matrix 411." *Id.*, 14:64-65.

Consistent with '948 patent's disclosure for "risk correlation matrix," Mukherjee explains that, in the DIDS system, the fifth layer of the Intrusion Detection Model maps events (within their context produced in level four) to threats. EX1003, ¶190.

The fifth layer of the model considers the threats to the network and the hosts connected to it. Events in context are combined to create threats. The threats are partitioned by the nature of the abuse and the nature of the target. In other words, what is the intruder doing, and what is he doing it to? Abuses are divided into attacks, misuses, and suspicious acts.

EX1005, p.13 (emphasis added).

A PHOSITA would have known that these attack categories represent different types of threats and that the fifth layer of the DIDS expert model includes a “risk correlation matrix” consistent with the ’948 patent’s use of the term.

EX1003, ¶¶189-191.

Mukherjee discloses an “event correlation matrix.” The ’948 patent explains that an “event correlation matrix 633...maps integrity warnings 634 to endpoint events 520.” EX1001, 15:10-19. Consistent with this explanation in the ’948 patent, Mukherjee discloses that the fourth layer of the DIDS director’s expert model “introduces the event in context. There are two kinds of context: temporal and spatial. As an example of temporal context, behavior which is unremarkable during standard working hours may be suspicious during off-hours.” EX1005, p. 13. The model therefore “allows for the application of information about wall-clock time to the events it is considering.” *Id.* “Spatial context implies the relative importance of the source of event. That is, events related to a particular user, or

events from a particular host, may be more likely to represent an intrusion than similar events from a different source.” *Id.*

A PHOSITA would have understood the Mukherjee’s description of the operation of the fourth layer of the expert model as mapping events to security issues is consistent with a “event correlation matrix.” *Id.*

For these reasons, Mukherjee in view of Cooper discloses limitation 1[c].
EX1003, ¶195.

1[d]. correlating, by the event and risk correlation matrix, threat classifications based on the temporal sequence of the generated real-time behavior based events; and

Mukherjee and Cooper each disclose this limitation.

As shown regarding limitation 1[c] above, Mukherjee discloses an “event correlation matrix” and a “risk correlation matrix.” EX1003, ¶198.

Mukherjee also discloses performing “threat classifications based on the temporal sequence of the generated real-time behavior based events.” Mukherjee explains that the event processing of the fourth layer of the IDM (which corresponds to an “event correlation matrix”) uses temporal proximity to correlate events. EX1003, ¶199.

In addition to the consideration of external temporal context, the expert system uses time windows to correlate events occurring in temporal proximity. This notion of temporal proximity implements the heuristic that a call to the UNIX who command followed closely by a login or

logout is more likely to be related to an intrusion than either of those events occurring alone. Such related patterns of behavior are also referred to as attack signatures [23,24].”

EX1005, p.13 (emphasis added).

Mukherjee also teaches that expert layer’s fifth layer relies on the output of fourth layer and thus the events “in context.” EX1003, ¶200.

The fifth layer of the model considers the threats to the network and the hosts connected to it. Events in context are combined to create threats. The threats are partitioned by the nature of the abuse and the nature of the target. In other words, what is the intruder doing, and what is he doing it to? Abuses are divided into attacks, misuses, and suspicious acts.

EX1005, p.13 (emphasis added).

Because the fifth layer depends on the fourth layer, the results of the fifth layer are also based on the temporal proximity discussed above. EX1003, ¶201. Thus, a PHOSITA would have understood this disclosure of Mukherjee to be “correlating by the event and risk correlation matrix threat classifications based on the temporal sequence” of the events. *Id.*

As shown regarding limitations 1[b] and 1[c], Cooper discloses monitoring applications executing on the computer network for vulnerability using methods and systems like those of Mukherjee. Cooper also discloses the limitation of “correlating, by the event and risk correlation matrix, threat classifications based

on the temporal sequence of the generated real-time behavior based events.”

EX1003, ¶¶203-204. Specifically, Cooper disclosed “continuous monitoring” such that a vulnerability could be classified as “new” (when it was seen for the first time), “continuous” (seen multiple times), and other similar identifications.

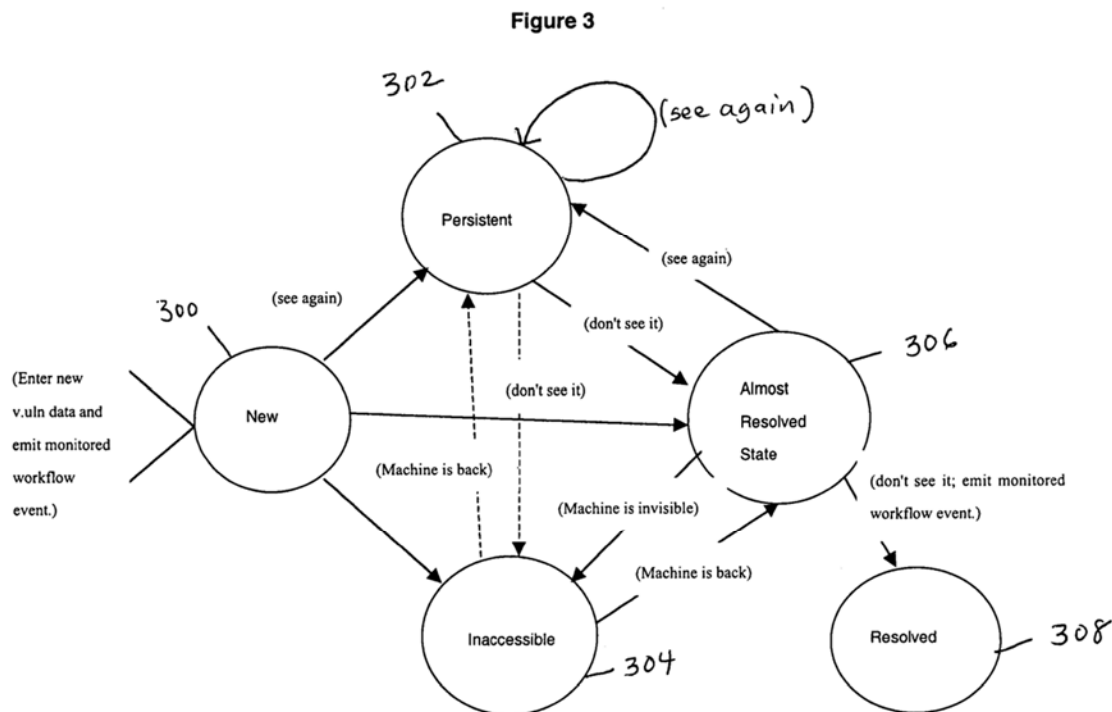
Cooper explains:

Referring to FIG. 3, new vulnerability data enters the system at state new 300 and a monitored event is emitted. If such vulnerability data is seen again, the state becomes persistent 302. If it is not seen again, it becomes almost resolved 306. If the machine on which the vulnerability is detected is no longer visible on the network, the state becomes inaccessible 304 and a monitored event is emitted. From persistent 302, if the vulnerability is detected again, the state remains persistent 302. From almost resolved 306, if a vulnerability is seen again, the state changes to persistent 302. If the vulnerability is not seen after an amount of time, the state becomes resolved 308 and a monitored event is emitted. If the vulnerability is from a machine which ceases to be visible on the network, the state becomes inaccessible 304. From the inaccessible state, if a machine returns from being invisible from the almost resolved state 306, the state returns to the almost resolved state 306. If a machine returns from being invisible from a persistent state 302, the state returns to the persistent state 302.

EX1006, 10:21-39 (emphasis added).

Further, Cooper discloses “continuous scanning” to produce states that vary based on when and how often an event is detected: “When continuous scanning is

used, the vulnerability assessment system provides changes to this state based on new vulnerabilities found or existing vulnerabilities removed,” which corresponds to this claim limitation. EX1006, 3:7-10. Cooper’s Figure 3 is reproduced below.



Thus, a PHOSITA would have found it obvious that the application vulnerability assessments in the combination of Mukherjee and Cooper would be based on the temporal sequence of monitored network events and that it would be obvious to include application vulnerability data from the methods of Cooper in the event correlation matrix and the risk correlation matrix threat classifications of Mukherjee. EX1003, ¶¶205-206.

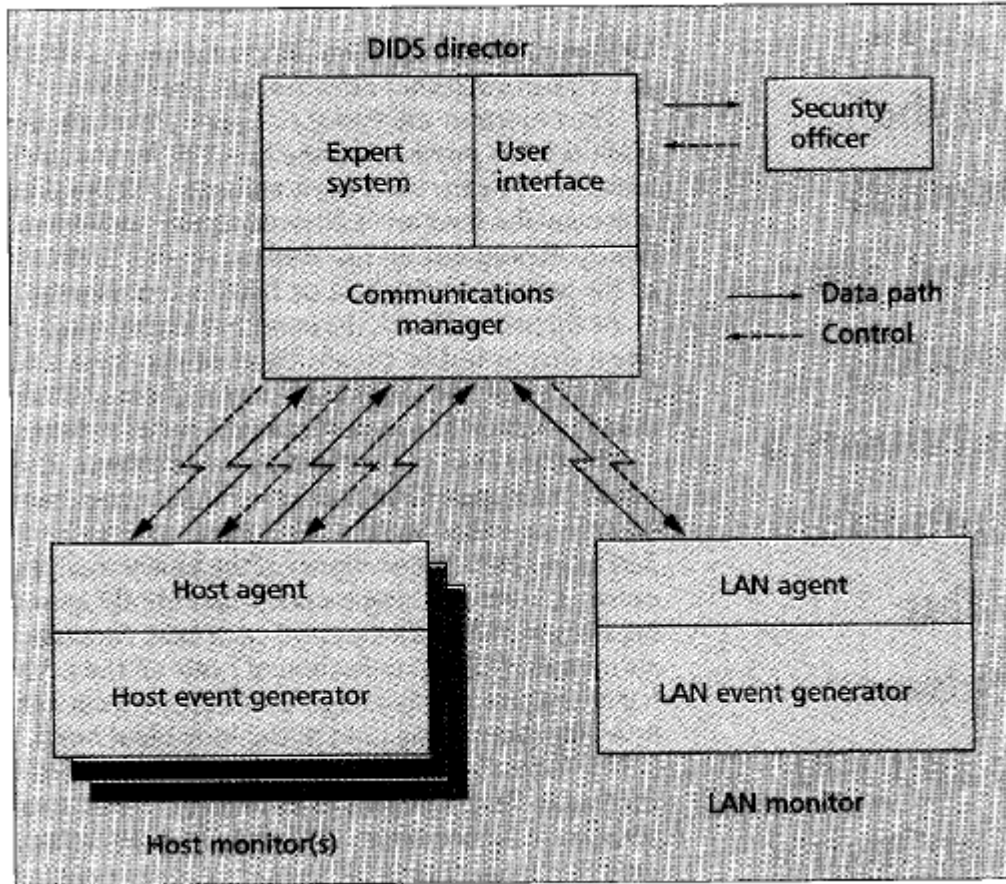
For these reasons, Mukherjee and Cooper each disclose limitation 1[d].

EX1003, ¶207.

1[e]. displaying, in a plurality of runtime dashboards of an administrative console of the computing environment, real-time status indications for operational integrity of the application.

Mukherjee, or alternatively Mukherjee in view of Cooper, discloses or suggests limitation 1[e].

Mukherjee discloses “displaying, in... an administrative console of the computing environment, real-time status indications for operational integrity of the application.” Mukherjee discloses that the DIDS director includes a user interface for permitting the System Security Officer (SSO) “interactive access to the entire system. The SSO can watch activities on each host, watch network traffic (by setting ‘wiretaps’), and request more specific types of information from the monitors” EX1005, p.12; EX1003, ¶210.



EX1005, Fig. 2.

The '948 patent specification explains that “data related to operational integrity, resource utilization, application integrity, and network activity” may be displayed in “graphical user interfaces (GUIs)” or “dashboards.” EX1001, 26:37-46, FIGS. 14-18. Mukherjee discloses displaying data related to network traffic, which a PHOSITA would understand to be “network activity,” and displaying activities on each host, which a PHOSITA would understand to be related to “application integrity.” EX1003, ¶211. A PHOSITA would further understand that Mukherjee’s user interfaces, which allow for “interactive access to the entire

system” would be presented to the SSO at a respective computer terminal (“administrative console of the computing environment”).

Cooper discloses displaying real-time vulnerability information along with associated risks for individual systems, which a PHOSITA would understand to be real-time displays of “operational integrity” of those systems. EX1003, ¶212.

The vulnerability event data can also be integrated into the PDD 114 accessible through the policy-based enterprise manager system user interface, such as Enterprise Manager UI. Each network object page includes a link to vulnerability information about that network object. Unlike the remainder of the PDD 114, such vulnerability information is generated dynamically by querying the vulnerability event database. The vulnerability information is rolled up by host IP address and vulnerability severity, as identified by the outcome component assigned to the events.

EX1006, 8:45-54 (emphasis added).

In the PDD 114, each network object page includes a link to vulnerability information pertaining to the network object. This link invokes a JSP with the appropriate parameters so as to execute a query against the vulnerability events in the database. The JSP constructs an HTML page(s) with vulnerability information sorted by the IP address of the target host. For each target host, a list of vulnerabilities that affect it is displayed. Clicking on each individual vulnerability event provides the vulnerability details information. A PDD 114 accessed through an enterprise manager system provides a view of vulnerability information for an entire policy domain. A PDD 114 accessed through the policy-

based monitor system 100 provides visibility only to the hosts scanned by the VA Server on that monitor.

Id., at 15:47-61, (emphasis added).

Cooper figure 6, reproduced below, shows data can be displayed in a web page from a monitoring system showing vulnerabilities in order of criticality and the temporal state. EX1003, ¶¶217-218.

FIG. 6 is an example view of a Web page from the policy-based monitor system user interface according to the invention. Specifically, a page from the SecurVantage™ Monitor system is shown. The Vulnerabilities tab 600 is selected, whereby a complete list of vulnerabilities is shown in order of the criticality component, severe, important, and informational. It should be appreciated that among other information, each persistent state was seen twice (Count=2). It should also be appreciated that relevant CVE and IAVA data is displayed.

EX1006, 16:36-44).

Figure 6

Securify SecurVantage™ Vulnerabilities - Microsoft Internet Explorer

Address: https://10.6.100.129/sven/varesummary.do?domainID=0&startTime=200403102000&reportingElement=Alamo&id=%7B2561d7be-c7ec-486f-b7c1-0d5e624...

User Name: administrator Logged Into: Monitor System: sample_domain, SVMonitor Policy Domain: sample_domain Page Refreshed: 12:49PM PST 03/11/04

SECURVANTAGE MONITOR

Live Data Traffic Vulnerabilities Manage

Previous: Vulnerability Data By Scanner > VA1

Summary: Alamo Page 1 of 1 or go to page 60

Scanner Name VA1 Last Scan Began 03/11/04 12:46PM PST Last Scan Finished >0% completed Scanner State Scanning

Click table headers to sort. Display: 50 Rows | 100 Rows | All Rows

	Vulnerability Name	Service	Port	State	Count	Component	Scanner Target	CVE	Days Old	Last Seen
1	View OpenSSH < 3.7.1	Ssh	22	Persistent	2	Severe	Alamo	CAN-2003-0602	1	03/11/04 12:46PM PST
2	View format string attack against statd	Udp	32768	Persistent	2	Severe	Alamo	CVE-2000-0666	1	03/11/04 12:46PM PST
3	View Apache < 2.0.43	Tcp	4578	Persistent	2	Important	Alamo	CAN-2002-1156	1	03/11/04 12:46PM PST
4	View Apache < 2.0.45	Tcp	4578	Persistent	2	Important	Alamo	CAN-2003-0132	1	03/11/04 12:46PM PST
5	View Apache < 2.0.46	Tcp	4578	Persistent	2	Important	Alamo	CAN-2003-0243	1	03/11/04 12:46PM PST
6	View Apache < 2.0.47	Tcp	4578	Persistent	2	Important	Alamo	CAN-2003-0192	1	03/11/04 12:46PM PST
7	View Apache < 2.0.48	Tcp	4578	Persistent	2	Important	Alamo	CVE-2002-0061	1	03/11/04 12:46PM PST
8	View Apache2 double slash dir index	Tcp	4578	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
9	View OpenSSH Reverse DNS Lookup bypass	Ssh	22	Persistent	2	Important	Alamo	CAN-2003-0386	1	03/11/04 12:46PM PST
10	View Portable OpenSSH PAM timing attack	Ssh	22	Persistent	2	Important	Alamo	CAN-2003-0190	1	03/11/04 12:46PM PST
11	View Remote host replies to SYN+FIN	Tcp	0	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
12	View SSH protocol version 1 enabled	Ssh	22	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
13	View Unconfigured web server	Tcp	4578	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
14	View fam service	Tcp	32769	Persistent	2	Important	Alamo	CVE-1999-0059	1	03/11/04 12:46PM PST
15	View http TRACE XSS attack	Tcp	4578	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
16	View icmp timestamp request	Icmp	0	Persistent	2	Important	Alamo	CAN-1999-0324	1	03/11/04 12:46PM PST
17	View statd service	Udp	32768	Persistent	2	Important	Alamo	CVE-1999-0018	1	03/11/04 12:46PM PST
18	View Directory Scanner	Tcp	4578	Persistent	2	Informational	Alamo		1	03/11/04 12:46PM PST
19	View HMAP	Tcp	4578	Persistent	2	Informational	Alamo		1	03/11/04 12:46PM PST

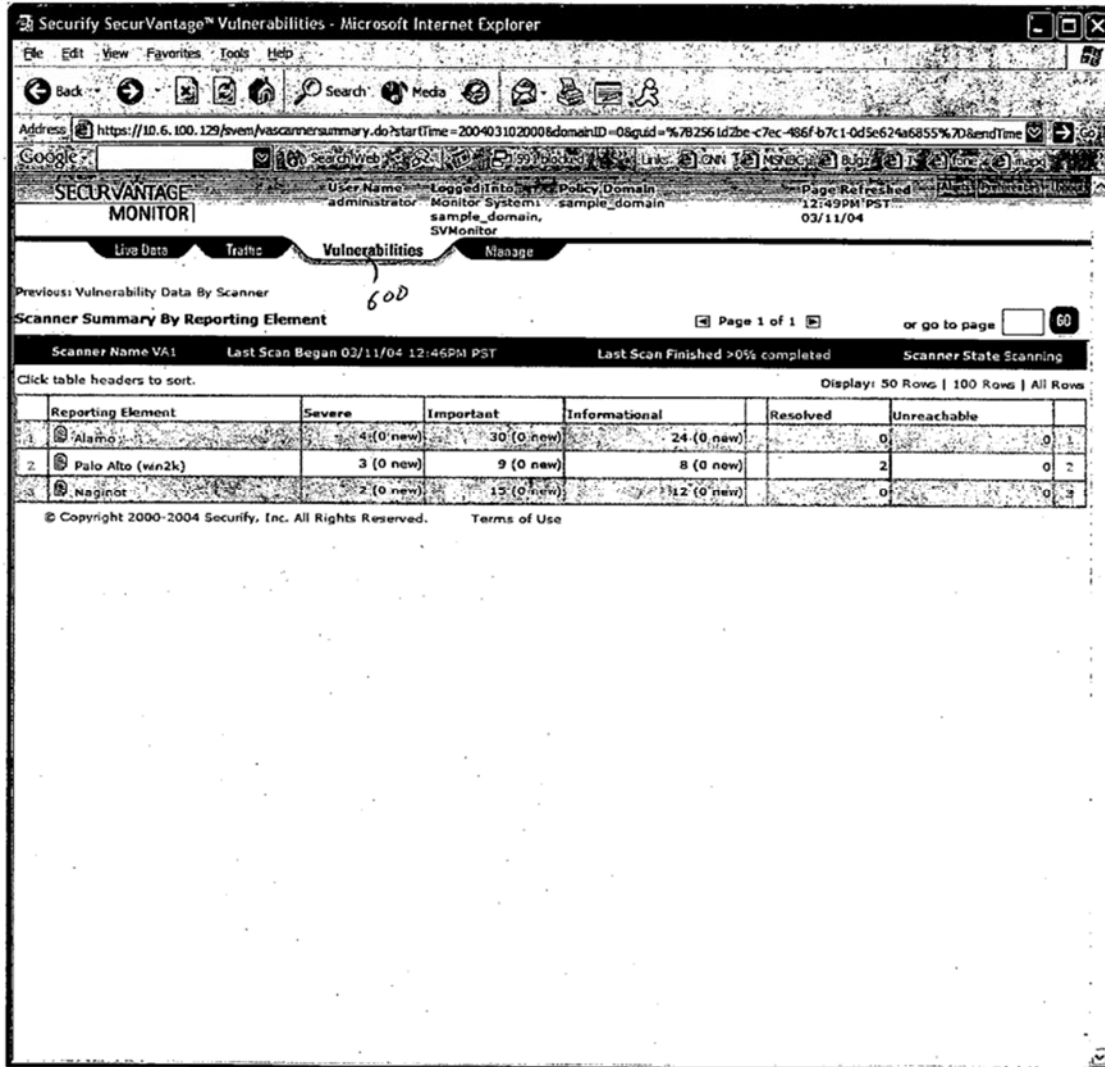
Cooper figure 7, reproduced below, is another web-page based example that shows vulnerabilities along with a summary of the vulnerability scanner's findings. EX1003, ¶¶219-220.

FIG. 7 is an example view of a Web page from the policy-based monitor system user interface according to the invention. The Vulnerabilities tab 600 is selected, whereby the specific view shown is Scanner

Summary by Reporting Element, of which there are three in this example.

EX1006, 16:46-49).

Figure 7

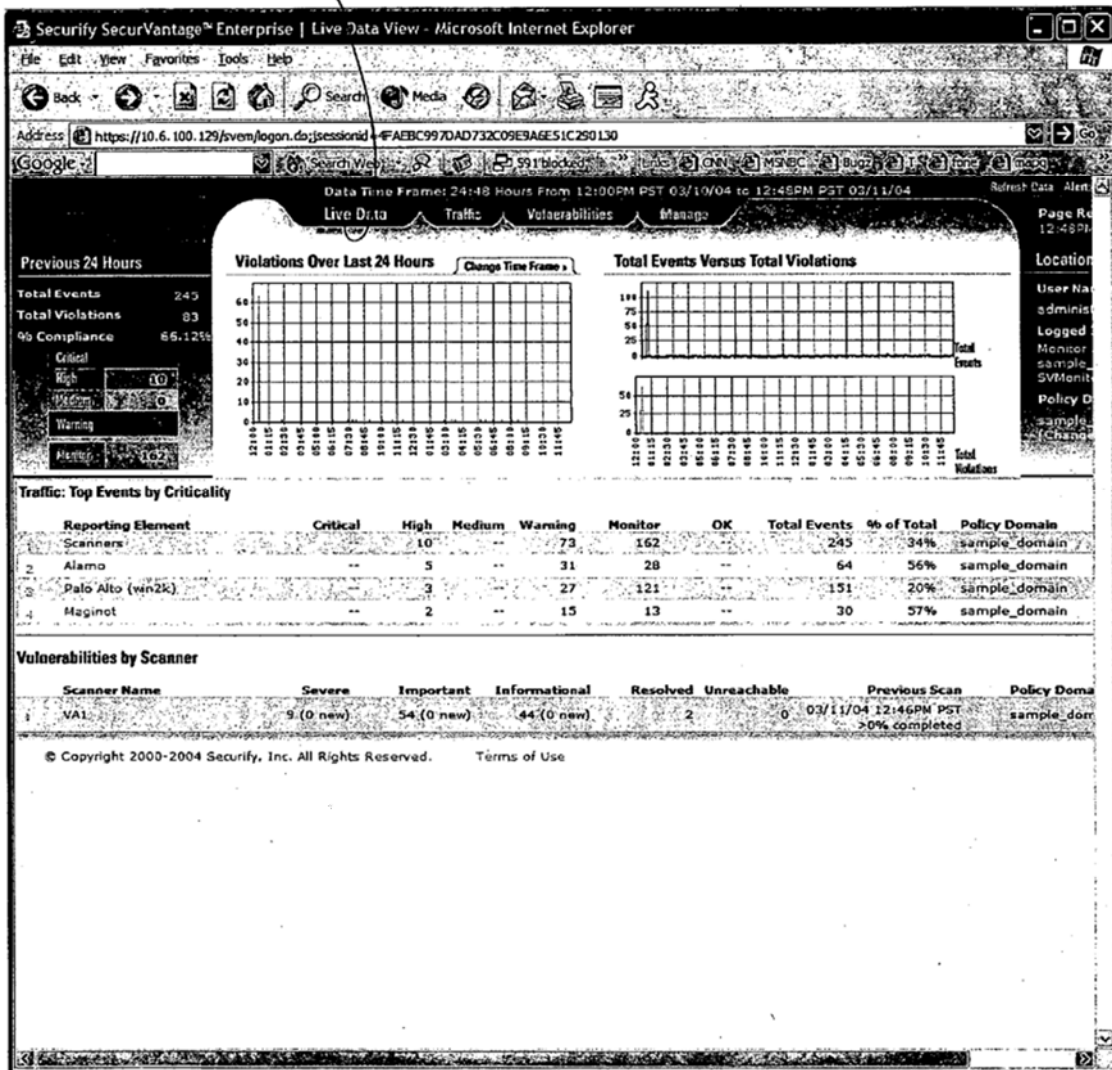


Cooper figure 8, reproduced below, is a webpage that shows data related to events over the past 24 hours identifying the number of events and the number of events that violated the security policy. EX1003, ¶¶221-222.

FIG. 8 is an example view of a Web page from the policy-based monitor system user interface according to the invention. The Live Data tab 800 is selected, whereby data from the previous 24 hours is shown. It should be appreciated that among other information, total events, total violations, and percent compliance is shown. Graphs showing total events versus total violations make is easy for the viewer to detect by visual means the vulnerability of the system at hand. Top events sorted by criticality is displayed in list form which makes it easy for the viewer to glean which reporting elements are impacted the most.

EX1006, 16:50-61.

Figure 8



A PHOSITA would have recognized each of these examples in Cooper as a GUI or “dashboard” of information and the collection of them as a “plurality of dashboards.” EX1003, ¶¶223-224. As shown above and regarding claim 2 below, Mukherjee and Cooper both disclose assessing system integrity and vulnerabilities in real time. Thus, a PHOSITA would have understood the example webpages in

figures 6-8 of Cooper implemented in the system of Mukherjee would be “realtime dashboards.” *Id.*, ¶224.

In addition to the general motivation to combine Mukherjee-Cooper described above, a PHOSITA would have been motivated to take advantage of Cooper’s real-time vulnerability displays in the system of Mukherjee. EX1003, ¶225. As noted above, Mukherjee discloses a user interface (UI) that gives the System Security Officer “interactive access to the entire system.” EX1005, p.12. However, Mukherjee does not provide details on how such a user interface would be constructed. A PHOSITA would have been motivated to look for other references for guidance on creating such an interface and would have recognized that Cooper’s example webpage views to be good examples of UI operations. EX1003, ¶225. A PHOSITA would have required minimal experimentation to apply these dashboards to Mukherjee in order to provide the SSO “interactive access to the entire system.” *Id.* A PHOSITA would have expected the use of these sample views from Cooper to have a high chance of success in being used to display the information of Mukherjee. *Id.*

For these reasons, Mukherjee, or alternatively Mukherjee in view of Cooper, discloses limitation 1[e]. EX1003, ¶226.

Because Mukherjee in view of Cooper discloses every limitation recited in claim 1 as explained above, a PHOSITA would have found claim 1 to be obvious

in view of the combination of Mukherjee in view of Cooper. EX1003, ¶233.

Accordingly, claim 1 should be found invalid under 35 U.S.C. § 103(a).

2. Claim 2

The method of claim 1, wherein the real-time operations of applications are monitored continuously.

Cooper discloses “real-time operations of applications are monitored continuously.” EX1003, ¶229.

Cooper discloses continuous scanning or monitoring (“monitored continuously”):

The invention relates generally to the field of network security management. More particularly, the invention relates to scanning for vulnerabilities on a continuous basis and interpreting the resulting traffic in the context of policy.

EX1006, 1:21-24.

Further, Cooper discloses that such continuous scanning or monitoring is with respect to vulnerabilities in customer applications (“real-time operations of applications”):

The vulnerability assessment system presents current vulnerability state of the infrastructure. When continuous scanning is used, the vulnerability assessment system provides changes to this state based on new vulnerabilities found or existing vulnerabilities removed. Throughout the process the policy adds context in terms specific to the

customer applications allowing prioritized action to be taken to remediate problems.

EX1006, 3:6-13.

A PHOSITA would have understood this disclosure of Cooper to correspond to this claim limitation. EX1003, ¶230-232. As discussed above and as indicated in the foregoing quotes, a PHOSITA would have been motivated to implement the incident identification and handling system of Cooper in the system to provide the system of Mukherjee “the ability to actively respond to attacks against the system in real time.” EX1003, ¶232.

Therefore, a PHOSITA would have found the limitations recited in claim 2 to be obvious based on Mukherjee in view of Cooper. EX1003, ¶¶232-233. As shown, the Mukherjee-Cooper combination discloses all limitations of claims 1 from which claim 2 depends. Accordingly, claim 2 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Mukherjee in view of Cooper.

3. Claim 3

The method of claim 1, wherein the event and behavior correlation engine receives one or more of:

**system events and infection profiles from a network activity sensor;
integrity measurement and verification reports of scans from a
system configuration sensor;
system events pertaining to processor, network, memory, and storage
resource consumption from a resource utilization sensor; and
system events pertaining to image profiles and local execution
context from an application integrity sensor.**

Mukherjee discloses an “event and behavior correlation engine [that] receives one or more of” these elements. In particular, the “DIDS director” disclosed in Mukherjee contains a component called a “communications manager”:

The communications manager is responsible for the transfer of data between the director and each of the host and the LAN monitors. It accepts the notable event records from each of the host and LAN monitors and sends them to the expert system.

EX1005, p.12 (emphasis added).

As described regarding claim limitation 1[b], Mukherjee’s DIDS director (or the communications manager thereof) is capable of receiving data from sensors.

Further, Mukherjee teaches a LAN monitor capable of monitoring network activity. EX1003, ¶239.

Like the host monitor, the LAN monitor consists of a LAN event generator (LEG) and a LAN agent. The LEG is currently a subset of the NSM [17, 27, 29]. Its main responsibility is to observe all of the traffic on its segment of the LAN in order to monitor host-to-host connections, services used, and volume of traffic. The LAN monitor reports on network activities such as rlogin and telnet connections, the use of security-related services, and changes in network traffic patterns.

EX1005, p.12.

The LAN monitor disclosed in Mukherjee then transmits these events using the communication manager (corresponding to “event and behavior correlation engine”). EX1003, ¶240.

The communications manager is responsible for the transfer of data between the director and each of the host and the LAN monitors. It accepts the notable event records from each of the host and LAN monitors and sends them to the expert system.

EX1005, p.12 (emphasis added).

A PHOSITA would have understood these disclosures of filtering events so as to accept the notable event records from each of the host and LAN monitors to involve receiving “system events and infection profiles from a network activity sensor” as in claim 3. EX1003, ¶243.

Mukherjee discloses a host monitor that collects information from an individual machine, including about “changes to the security state of the system”. It also “report[s] anomalous behavior aggregated over time”. EX1005, pp.11-12; EX1003, ¶245. In addition, as outlined above, the host agent transmits such information to the DIDS director. EX1003, ¶¶245-246.

Thus, Mukherjee in view of Cooper discloses at least one of the alternative limitations listed in claim 3, any one of which is sufficient to render the entire element obvious. *See e.g., Titanium Metals Corp. of Am. v. Banner*, 778 F.2d at 782.

Therefore, a PHOSITA would have found the limitations recited in claim 3 to be obvious based on Mukherjee in view of Cooper. EX1003, ¶247. As shown, the Mukherjee-Cooper combination discloses all limitations of claims 1 from which claim 3 depends. Accordingly, claim 3 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Mukherjee in view of Cooper.

4. Claim 4

The method of claim 3, wherein the scans are performed as per a schedule, and wherein the schedule is one or more of hourly, daily, weekly, bi-weekly, or monthly, by a third party collaboration service to score the severity of an application based on the vulnerability, compliance, and patch status to affirm positive presence of a threat.

Mukherjee and Cooper both disclose “wherein the scans are performed as per a schedule, and wherein the schedule is one or more of hourly, daily, weekly, bi-weekly, or monthly”. EX1003, ¶249.

Mukherjee discloses that its monitoring can run daily:

IDES monitors target system activity as it is recorded in audit records generated by the target system. Due to the fact that these profiles are updated daily, IDES is able to adaptively learn a subject’s behavior patterns; as users alter their behavior, the profiles change to reflect the most recent activity.

EX1005, p.5.

Cooper describes a system that is capable of running on various schedules, including several of those recited in Claim 4. EX1003, ¶¶252. In particular, Cooper discloses:

Using the policy-based enterprise manager system user interface the user configures how often a network scan is to take place, e.g. daily or weekly, and the maximum network bandwidth that it should utilize. This allows a network operator to schedule scans during off-peak periods and to limit network bandwidth utilization to an acceptable level.

Using the policy-based enterprise manager system user interface the user can also download and/or install updates of the vulnerability definitions. Such updates can be downloaded automatically or on demand from a central website, such as that of the Securify, Inc.

EX1006, 8:55-65 (emphasis added).

Thus, Mukherjee and Cooper each disclose at least one of the alternative schedules listed in claim 4, any one of which is sufficient to render the entire element obvious. EX1003, ¶¶253-255. *See e.g., Titanium Metals*, 778 F.2d at 782.

Mukherjee in view of Cooper further discloses “wherein the scans are performed...by a third party collaboration service to score the severity of an application based on the vulnerability, compliance, and patch status to affirm positive presence of a threat.” EX1003, ¶254.

Mukherjee discloses a security monitor that works with its IDES system: “The Information Security Officer’s Assistant (ISOA) is a real-time security monitor implemented on a UNIX-based workstation that supports automated as well as interactive audit trail analysis [12, 13].” EX1005, p.5; EX1003, ¶255. The

ISOA is capable of examining “concern levels” based on the events received (e.g., events about an application):

ISOAs analysis capabilities include both statistical as well as expert system components. These components cooperate in the automated examination of various “concern levels” of data analysis. As recognized indicators (sets of indicators) are matched, concern levels increase and the system begins to analyze increasingly detailed classes of audit events for the user or host in question.

EX1005, p.6.

A POSITA would have recognized that “concern levels” indicate higher or lower levels of potentially problematic (e.g., threatening) behavior being analyzed by the system, and would have recognized that such “concern levels” teach a concept corresponding to “scor[ing] the severity of an application” relative to its potential threat. EX1003, ¶256.

Further, Cooper discloses obtaining severity information about vulnerabilities, for example, from a third-party tool called Nessus.

For each of the states hereinabove, a severity can be selected from the scanner, such as Nessus. For example, Nessus severity selections are: severe, informational, and important.

EX1006, 10:12-15.

A POSITA would have recognized that the “severity selections” disclosed by Cooper (“severe, informational, important”) indicate higher or lower levels of

potentially problematic (e.g., threatening) behavior being analyzed by the system, and would have recognized that such “severity selections” teach a concept similar to the claimed “scor[ing] the severity of the application”. EX1003, ¶¶258-259.

Further, Cooper discloses that information it collects about vulnerabilities includes the following: “Common Identifier, a string containing one or more identifiers from common vulnerability repositories such as the Common Vulnerabilities and Exposures (CVE) number”. EX1006, 13:14-16. A POSITA would have understood that such CVE numbers identify that a recognized, documented, published vulnerability exists, and that such identification “affirm[s] positive presence of a threat.” EX1003, ¶259.

Therefore, a PHOSITA would have found the limitations recited in claim 4 to be obvious based on Mukherjee in view of Cooper. EX1003, ¶¶260-263. As shown, the Mukherjee-Cooper combination discloses all limitations of claims 1 and 3 from which claim 4 depends. Accordingly, claim 4 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Mukherjee in view of Cooper.

5. Claim 5

The method of claim 1, wherein the plurality of runtime dashboards are a plurality of application runtime reputation score dashboards that comprise one or more of:

a global view dashboard;

a system configuration dashboard;

a resource utilization dashboard;

an application integrity dashboard; and

a network activity dashboard.

As shown above regarding claim limitation 1[e], Mukherjee, or alternatively Mukherjee in view of Cooper, discloses or suggests a “plurality of runtime dashboards” in the form of user interfaces or GUIs for presenting information about the respective monitored systems or applications.

Mukherjee discloses or suggests that its user interfaces “are a plurality of application runtime reputation score dashboards” in that Mukherjee describes that the expert system of the DIDS architecture “produces a numeric value between one and 100 which represents the overall security state of the network. The higher the number, the less secure is the network. This value is a function of all threats for all the subjects on the system.” EX1005, p. 13; EX1003, ¶97. A PHOSITA would understand that such numeric value or score representing the security state of a network (including its applications) is an “application runtime reputation score.” EX1003, ¶269.

As shown above regarding claim limitation 1[e], Mukherjee explains that the DIDS director includes a user interface for permitting the System Security Officer (SSO) views into and control over the network or system, including network traffic. EX1003, ¶¶267-268; EX1005, p.12. A PHOSITA would understand that in order for the SSO to monitor and take actions to address network security, such numeric value or score would be presented or displayed to the SSO in a user interface or GUI (corresponding to “dashboard”).

The claim language “one or more” only requires that one, not all, of the claimed dashboards in the list be present or disclosed. *See e.g., Titanium Metals Corp. of Am. v. Banner*, 778 F.2d at 782. EX1003, ¶275. A PHOSITA would understand that such user interface or dashboard presenting numeric value or score representing the security state of a network is a “network activity dashboard”. EX1003, ¶¶270-271. Moreover, as described in the general motivation to combine, Mukherjee anticipates combining its disclosures with additional components or tools. EX1003, ¶¶111-112, 270-271.

As such, Mukherjee teaches or suggests the claim 5 limitations.

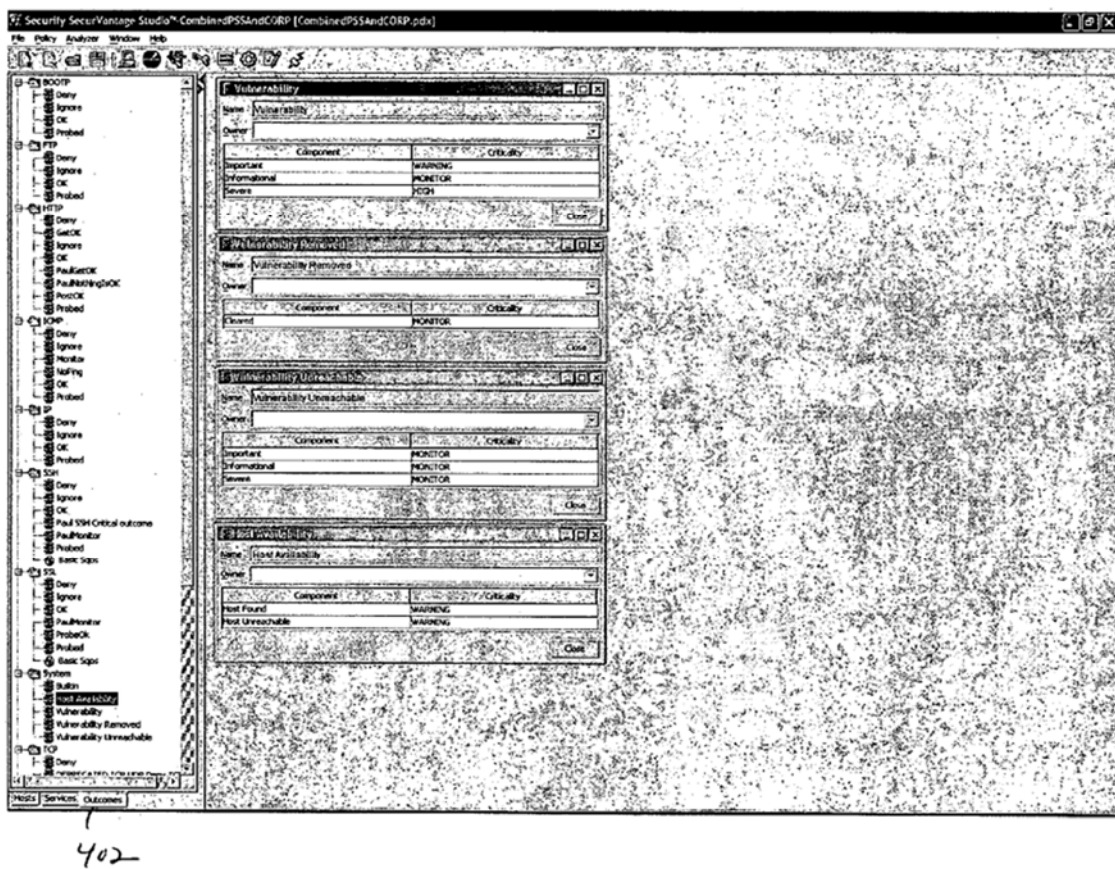
Alternatively, Mukherjee in view of Cooper teaches the limitation.

As shown above regarding claim limitation 1[e], Cooper discloses multiple webpage displays that a PHOSITA would have understood to disclose one or more of the dashboards listed in claim 5, any one of which is sufficient to render the

entire element obvious. See e.g., *Titanium Metals Corp. of Am. v. Banner*, 778 F.2d at 782. EX1003, ¶¶272.

Cooper figure 5 reproduced below shows “possible outcomes of events.”
EX1006, 16:33-34.

Figure 5



Cooper figure 6 reproduced below shows “a complete list of vulnerabilities
is shown in order of the criticality component, severe, important, and
informational.” EX1006, 16:41-43.

Figure 6

Securify SecurVantage™ Vulnerabilities - Microsoft Internet Explorer

Address: https://10.6.100.129/sven/varesummary.do?domainID=0&startTime=200403102000&reportingElement=Alamo&guid=%7B2561d2be-c7ec-486f-b7c1-0d5e624...

User Name: administrator Logged In: Monitor System: sample_domain Policy Domain: sample_domain Page Refreshed: 12:49PM PST 03/11/04

SECURVANTAGE MONITOR

Live Data Traffic Vulnerabilities Manage

Previous: Vulnerability Data By Scanner > VA1

Summary: Alamo Page 1 of 1 or go to page 60

Scanner Name VA1 Last Scan Began 03/11/04 12:46PM PST Last Scan Finished >0% completed Scanner State Scanning

Click table headers to sort. Display: 50 Rows | 100 Rows | All Rows

	Vulnerability Name	Service	Port	State	Count	Component	Scanner Target	CVE	Days Old	Last Seen
1	View OpenSSH < 3.7.1	Ssh	22	Persistent	2	Severe	Alamo	CAN-2003-0602	1	03/11/04 12:46PM PST
2	View format string attack against statd	Udp	32768	Persistent	2	Severe	Alamo	CVE-2000-0666	1	03/11/04 12:46PM PST
3	View Apache < 2.0.43	Tcp	4578	Persistent	2	Important	Alamo	CAN-2002-1156	1	03/11/04 12:46PM PST
4	View Apache < 2.0.45	Tcp	4578	Persistent	2	Important	Alamo	CAN-2003-0132	1	03/11/04 12:46PM PST
5	View Apache < 2.0.46	Tcp	4578	Persistent	2	Important	Alamo	CAN-2003-0243	1	03/11/04 12:46PM PST
6	View Apache < 2.0.47	Tcp	4578	Persistent	2	Important	Alamo	CAN-2003-0192	1	03/11/04 12:46PM PST
7	View Apache < 2.0.48	Tcp	4578	Persistent	2	Important	Alamo	CVE-2002-0061	1	03/11/04 12:46PM PST
8	View Apache2 double slash dir index	Tcp	4578	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
9	View OpenSSH Reverse DNS Lookup bypass	Ssh	22	Persistent	2	Important	Alamo	CAN-2003-0386	1	03/11/04 12:46PM PST
10	View Portable OpenSSH PAM timing attack	Ssh	22	Persistent	2	Important	Alamo	CAN-2003-0190	1	03/11/04 12:46PM PST
11	View Remote host replies to SYN+FIN	Tcp	0	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
12	View SSH protocol version 1 enabled	Ssh	22	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
13	View Unconfigured web server	Tcp	4578	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
14	View fam service	Tcp	32769	Persistent	2	Important	Alamo	CVE-1999-0059	1	03/11/04 12:46PM PST
15	View http TRACE XSS attack	Tcp	4578	Persistent	2	Important	Alamo		1	03/11/04 12:46PM PST
16	View icmp timestamp request	Icmp	0	Persistent	2	Important	Alamo	CAN-1999-0324	1	03/11/04 12:46PM PST
17	View statd service	Udp	32768	Persistent	2	Important	Alamo	CVE-1999-0018	1	03/11/04 12:46PM PST
18	View Directory Scanner	Tcp	4578	Persistent	2	Informational	Alamo		1	03/11/04 12:46PM PST
19	View HMAP	Tcp	4578	Persistent	2	Informational	Alamo		1	03/11/04 12:46PM PST

A PHOSITA would have understood that Cooper Figures 5 and 6 disclose “application runtime reputation score dashboards” as they show, among other things, a list of vulnerabilities in order of the criticality component, severe, important, and informational (“application runtime reputation score”). EX1003, ¶¶273-276. Further, Figure 14 of Cooper discloses a webpage with a display that a

PHOSITA would understand corresponds to a global view dashboard as reciting in claim 5. *Id.*, ¶¶277-278.

Therefore, a PHOSITA would have found the limitations recited in claim 5 to be obvious based on Mukherjee in view of Cooper. EX1003, ¶279. As shown, the Mukherjee-Cooper combination discloses all limitations of claim 1 from which claim 5 depends. Accordingly, claim 5 should be found invalid as unpatentable under 35 U.S.C. § 103 based on Mukherjee in view of Cooper.

D. NO BASIS FOR DISCRETIONARY DENIAL

There is no basis to deny institution under 35 U.S.C. § 325(d).

There is parallel court litigation, so *Fintiv* factors are addressed preliminarily. First, no stay has been entered in the parallel litigation. Second, trial is currently set to begin February 5, 2024. Third, it is early in the parallel litigation as neither party has responded to written fact discovery nor taken any other type of discovery, nor have the parties begun the Markman process. Fourth, Defendants in the parallel litigation served preliminary invalidity contentions on March 28, 2023. Fifth, Petitioner and Taasera are parties in the parallel litigation as identified in the Mandatory Notices below and the Challenged Claims are the same as the claims identified in Taasera infringement contentions. As such, the first five *Fintiv* factors do not favor discretionary denial. Moreover, even if that

were not the case, the merits of the grounds raised in this Petition are particularly strong for all of the reasons discussed above.

Discretionary denial is strongly disfavored here because the “same or substantially the same art or arguments” were not presented to the Office, and the Office “erred in a manner material” to patentability by not asserting the Halperin, Inoue, and Patil references. *See Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH*, IPR2019-01469, Paper 6, 8-9 (Feb. 13, 2020). Analysis of this issue is guided by the “*Becton*” factors. *Id.*, at 9-11.

Mukherjee and Cooper were not cited, individually or in combination, in any of the Office communications. *See* EX1006. Regarding *Becton* factors (a) (material differences between asserted art and prior art) and (b) (cumulative nature of the asserted art and prior art), Mukherjee and Cooper are materially different from the combination of references asserted in Office communications because Mukherjee and Cooper disclose all of the limitations discussed above. Thus, the individual references of Mukherjee and Cooper are not merely cumulative of references the Office relied on. Regarding *Becton* factor (d) (extent of overlap), the arguments in this Petition do not overlap with arguments made during examination.

Even if the Board finds that the “same or substantially the same” art or arguments were presented to the Office, discretionary denial under § 325(d) is still

not proper because the Office made a “material error” in not considering the Mukherjee and Cooper references.

Regarding *Becton* factors (c) (evaluation of art during examination), none of Mukherjee and Cooper were used in any claim rejections, weighing strongly against exercising discretion. *See Cellco Partnership v. Huawei Device Co.*, IPR2020-01117, Paper 10, 13 (Feb. 3, 2021).

Regarding *Becton* factor (e) (examiner error) and (f) (other facts), this petition specifically points out and explains how Mukherjee and Cooper discloses all limitations of the claims.

E. CONCLUSION

Inter partes review of claims 1-5 of U.S. Patent No. 8,990,948 is requested based on the grounds described above.

F. Mandatory Notices (37 C.F.R. § 42.8(a)(1))

REAL PARTY IN INTEREST:

The real parties-in-interest are Trend Micro, Inc., Trend Micro America, Inc. and Trend Micro Incorporated. Trend Micro, Inc. is a wholly owned subsidiary of Trend Micro America, Inc., which is a wholly owned subsidiary of Trend Micro Incorporated, a Japanese Corporation.

RELATED MATTERS:

Pursuant to 37 C.F.R. § 42.8(b)(2), Petitioner identifies the following related matters.

The '948 patent is at issue in the following district court litigations:

- *In re: Taasera Licensing LLC, Patent Litigation*, 2:22-md-03042 (E.D. Tex.), which was consolidated by the Judicial Panel on Multidistrict Litigation on August 3, 2022.

A petition for *inter partes* review is being filed concurrently with respect to U.S. Patent No. 9,092,616, which is a continuation of the '948 patent.

LEAD AND BACKUP COUNSEL:

Pursuant to 37 C.F.R. § 42.8(b)(3) and 37 C.F.R. § 42.10(a), Petitioner appoints:

Robert M. Hansen (Reg. No. 43,656) as lead counsel,

Scott B. Amankwatia (Reg. No. 78,226) as back-up counsel.

Matthew Anderson (Reg. No. 78,257) as back-up counsel, and

Pursuant to 37 C.F.R. § 42.10(b), a power of attorney is being filed with this designation of counsel.

SERVICE INFORMATION:

Petitioner provides the following service information for designated counsel:

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Tel: (703)-391-2900

Fax: (703)-391-2900

Petitioner consents to electronic service by email to the email addresses:

rhansen@marburylaw.com;

samankwatia@marburylaw.com;

manderson@marburylaw.com; and

pat-docketing@marburylaw.com.

PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to
Deposit Account No. 18-1579.

Respectfully submitted,
The Marbury Law Group, PLLC

A handwritten signature in cursive script, reading "Robert M. Hansen", written in black ink over a horizontal line.

Robert M. Hansen, Reg. No. 43,656
Attorney for Petitioner

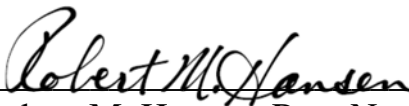
Date: April 6, 2023

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: (703) 391-2900

CERTIFICATION OF SERVICE (37 C.F.R. §§ 42.6(e), 42.105(a))

The undersigned hereby certifies that the above-captioned **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 8,990,948**, Petitioner's power of attorney, and all associated exhibits were served in their entireties on April 6, 2023, on the following party via FedEx:

Buchanan, Ingersoll & Rooney PC
1737 King Street
Suite 500
Alexandria, VA 22314


Robert M. Hansen, Reg. No. 43,656
Attorney for Petitioner

Date: April 6, 2023

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Tel: (703)-391-2900

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION

Pursuant to 37 C.F.R. § 42.24 *et seq.*, the undersigned certifies that this Petition complies with the 14,000-word type-volume limitation. The Petition contains **13,079** words, excluding the table of contents, table of authorities, Table of Exhibits, mandatory notices under § 42.8, the signature block, the claims appendix, certificate of service, certificate of word count, appendix of exhibits, and words that appear in figures copied from the cited prior art.

The Marbury Law Group, PLLC

A handwritten signature in black ink, reading "Robert M. Hansen", written over a horizontal line.

Robert M. Hansen, Reg. 43,656
ATTORNEY FOR PETITIONER

Date: April 6, 2023

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: 703-391-2900