

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Matthew L. Boney
U.S. Patent No.: 8,181,244 Attorney Docket No.: 54971-0005IP1
Issue Date: May 15, 2012
Appl. Serial No.: 11/408,145
Filing Date: April 20, 2006
Title: BACKWARD RESEARCHING TIME STAMPED EVENTS TO
FIND AN ORIGIN OF PESTWARE

Mail Stop Patent Board

Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

**PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES PATENT
NO. 8,181,244 PURSUANT TO 35 U.S.C. §§ 311–319, 37 C.F.R. § 42**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	REQUIREMENTS FOR IPR	2
A.	Grounds for Standing.....	2
B.	Challenge and Relief Requested Under 37 C.F.R. §42.104(b)	2
III.	THE '244 PATENT	3
A.	Brief Description.....	3
B.	Prosecution History.....	5
C.	Level of Ordinary Skill	6
D.	Claim Construction	6
IV.	THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	7
A.	[GROUND 1A] – CLAIMS 1-10 AND 13-16 WOULD HAVE BEEN OBVIOUS OVER HARTRELL, FAN, AND ROCKWOOD	7
1.	Hartrell (EX1004).....	7
2.	Fan (EX1005)	10
3.	The combination of Hartrell and Fan	11
4.	Rockwood (EX1006).....	13
5.	The Combination of Hartrell, Fan, and Rockwood.....	15
6.	Element-by-Element Explanation of How Hartrell-Fan-Rockwood Combination Renders Challenged Claims Obvious.....	18
B.	[GROUND 1B] – CLAIMS 11 AND 12 WOULD HAVE BEEN OBVIOUS OVER HARTRELL, FAN, ROCKWOOD, AND POLYAKOV.....	53
1.	Polyakov (EX1007)	53
2.	The Combination of Hartrell, Fan, Rockwell, and Polyakov	54
3.	Element-by-Element Explanation of How Hartrell-Fan- Rockwood-Polyakov Combination Renders Challenged Claims Obvious	57
V.	PTAB DISCRETION SHOULD NOT PRECLUDE INSTITUTION.....	60
A.	The <i>Fintiv</i> Factors Favor Institution—§314(a)	60
1.	Factor 1	60
2.	Factor 2	61
3.	Factor 3	63
4.	Factor 4	64
5.	Factor 5	65
6.	Factor 6: The Merits Support Institution.....	65
B.	§ 325(d) Favors Institution	65

VI.	FEES	66
VII.	CONCLUSION.....	66
VIII.	MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1).....	66
	A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1).....	66
	B. Related Matters Under 37 C.F.R. § 42.8(b)(2)	66
	C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3).....	67
	D. Service Information	67

EXHIBITS

EX1001	U.S. Patent No. 8,181,244 (“the ’244 Patent”)
EX1002	Excerpts from the Prosecution History of the ’244 Patent (“the Prosecution History”)
EX1003	Declaration of Dr. Markus Jakobsson
EX1004	U.S. Patent Pub. No. 2007/0150957 to Gregory D. Hartrell <i>et al.</i> (“Hartrell”)
EX1005	U.S. Patent No. 7,784,098 to Paul Fan <i>et al.</i> (“Fan”)
EX1006	U.S. Patent No. 7,352,280 to Troy Dean Rockwood <i>et al.</i> (“Rockwood”)
EX1007	U.S. Patent No. 7,571,482 to Alexey A. Polyakov <i>et al.</i> (“Polyakov”)
EX1008	U.S. Patent No. 8,516,583 to Anil Francis Thomas <i>et al.</i> (“Thomas”)
EX1009	U.S. Patent No. 7,549,055 to Vincent J. Zimmer <i>et al.</i> (“Zimmer”)
EX1010	U.S. District Courts—Median Time Intervals From Filing to Disposition of Civil Cases Terminated, by District and Method of Disposition, During the 12-Month Period Ending March 31, 2023
EX1011	<i>Memorandum: Interim Procedure for Discretionary Denials in AIA Post-Grant Proceedings with Parallel District Court Litigation</i> (June 21, 2022) (“Director’s Memo”)

- EX1012 *Webroot, Inc. v. CrowdStrike, Inc.*, Case No. 6:22-cv-00243-ADA¹, Docket No. 252, “Order Granting Fifth Amended Scheduling Order” (WDTX Mar. 29, 2023)
- EX1013 *Webroot, Inc. v. CrowdStrike, Inc.*, Case No. 6:22-cv-00243-ADA, Docket No. 350, “Order Resetting Markman Hearing” (WDTX Jun. 8, 2023)
- EX1014 *Webroot, Inc. v. CrowdStrike, Inc.*, Case No. 6:22-cv-00241-ADA, Docket No. 263, “Opening Claim Construction Brief Regarding Patents Plaintiffs Webroot And OpenText Added By Amendment” (WDTX Apr. 12, 2023)
- EX1015 *Webroot, Inc. v. CrowdStrike, Inc.*, Case No. 6:22-cv-00243-ADA, Docket No. 304, “Plaintiffs’ Responsive Claim Construction Brief Regarding Patents Plaintiffs Added By Amendment” (WDTX May 5, 2023)

¹ The case number for *Webroot, Inc. v. CrowdStrike, Inc.* is 6:22-CV-00241-ADA-DTG, but the case was consolidated for all pretrial issues (except venue and disqualification) with 6:22-CV-00243-ADA, and the joint filings are available under the docket of 6:22-CV-00243-ADA.

CLAIM LISTING

Limitation	Claim Language
[1pre]	A method for identifying an origin of suspected pestware activity on a computer, the method comprising:
[1a]	monitoring, with a kernel-mode driver, activity on the computer;
[1b]	generating an activity log on a file storage device of the computer from the kernel-mode driver;
[1c]	receiving, from a user via an interface of the computer, a time of interest relating to a suspicion of pestware on the computer, wherein the time of interest includes a time interval;
[1d]	issuing a timestamp after receiving the time of interest;
[1e]	identifying, based upon the time of interest, indicia of pestware, wherein the identifying is initiated by the issuing the timestamp; and
[1f]	accessing, using a hardware processor of the computer, at least a portion of a recorded history of externally networked sources that the computer received files from
[1g]	so as to identify, based at least in part upon the identified indicia of pestware, a reference to an identity of an externally networked source that is suspected of originating pestware;
[1h]	wherein the recorded history of externally networked sources is stored on the file storage device.
[2a]	The method of claim 1, wherein the identifying includes: accessing at least one log of historical events on the computer;
[2b]	locating at least one event in the at least one log that corresponds to the time of interest; and
[2c]	determining that the at least one event is associated with the indicia of pestware.
[3]	The method of claim 2, wherein the determining that the at least one event is associated with the indicia of pestware includes following references in the at least one log to a suspected pestware object.

Limitation	Claim Language
[4]	The method of claim 3, wherein following the references includes following at least one reference in the activity log from the event to a suspected pestware process.
[5]	The method of claim 2, wherein the at least one log includes at least one log selected from the group consisting of an activity log, a browser history, browser cache, browser settings, operating system settings, an event log, a debugging log, a firewall log, file information and monitoring software logs.
[6]	The method of claim 1, wherein the user input is in response to an alert provided to the user regarding a detected suspicious activity on the computer.
[7]	The method of claim 1, including: reporting the identity of the externally networked source to an externally networked pestware research entity so as to enable the pestware research entity to research whether the externally networked source is a source of pestware.
[8]	The method of claim 1, wherein the externally networked source is identified by an identifier selected from the group consisting of an I.P. address and a URL.
[9pre]	A system for identifying a source of suspected pestware on a computer, the system comprising:
[9a]	a hardware processor;
[9b]	a file storage device coupled to the hardware processor;
[9c.i]	a timestamp module configured to be executed by the hardware processor, to generate a time of interest including a time interval in response to detected activity on the computer that is indicative of pestware,
[9c.ii]	to issue a timestamp after generating the time of interest, and
[9c.iii]	to initiate, in response to the issuance of the timestamp, identification of indicia of pestware;
[9d]	a research portion configured to be executed by the hardware processor and to access at least one log on a file storage device of the computer to relate the time of interest to at least one externally networked source of potential pestware activity on the computer;

Limitation	Claim Language
[9e]	a kernel-mode driver configured to monitor activity on the computer and to generate an activity log that is included within the at least one log; and
[9f]	a reporting portion configured to be executed by the hardware processor and to generate a report that identifies the at least one externally networked source of potential pestware activity.
[10a]	The system of claim 9, wherein the at least one log includes information that relates the time of interest to an event, and
[10b]	wherein the at least one log includes information that relates the event to a suspected pestware object, and
[10c]	wherein the at least one log includes information that relates the suspected pestware object to the at least one externally networked source of potential pestware activity.
[11]	The system of claim 10, including: a heuristics module configured to be executed by the hardware processor and to evaluate whether the event is a suspected pestware-related event before the research portion accesses information in the at least one log that relates the event to the suspected pestware object.
[12]	The system of claim 11, wherein the heuristics module determines that the at least one event is suspected to be pestware-related by analyzing other events in connection with the event, wherein the other events are also related to the time of interest.
[13]	The system of claim 9, wherein the at least one log includes at least one log selected from the group consisting of an activity log, a browser history, browser cache, browser settings, operating system settings, an event log, a debugging log, a firewall log, file information and monitoring software logs.
[14]	The system of claim 9, wherein the timestamp module is further configured to generate the time of interest as a point in time.
[15]	The system of claim 9, wherein the at least one externally networked source of potential pestware activity is identified by an identifier selected from the group consisting of an I.P. address and a URL.

Limitation	Claim Language
[16]	The system of claim 9, wherein the at least one log includes at least one log selected from the group consisting of an activity log, a browser history, browser cache, browser settings, operating system settings, an event log, a debugging log, a firewall log, file information and monitoring software logs.

I. INTRODUCTION

CrowdStrike, Inc. (“Petitioner”) petitions for IPR of claims 1-16 (“the Challenged Claims”) of U.S. Patent No. 8,181,244 (“the ’244 Patent”). As will be shown here, the Challenged Claims are not inventive, but rather are directed to long-known computer system architectures and associated methods.

Specifically, the ’244 Patent’s inventor claims to have developed a “system and method for identifying an origin of suspected pestware activity on a computer” based around a “time of interest.” EX1001, Abstract. But such methods were not new. Indeed, Hartrell similarly describes a “malware analysis system” in which, “[u]pon being informed of a suspected malware,” the system “creates a time bounded snapshot of the monitored activities that were conducted within a time frame prior to the notification of the suspected malware infection.” EX1004, Abstract. As will be described herein, Hartrell renders obvious a system having each of the claimed features that constitute the supposedly new system and method of the ’244 Patent.

Accordingly, Petitioner respectfully submits that an IPR should be instituted, and that the Challenged Claims should be canceled as unpatentable.

II. REQUIREMENTS FOR IPR

A. Grounds for Standing

Petitioner certifies that the '244 Patent is available for IPR. Petitioner is not barred or estopped from requesting this review of the Challenged Claims.

B. Challenge and Relief Requested Under 37 C.F.R. §42.104(b)

Petitioner requests IPR of the Challenged Claims on the below grounds, which are supported by the expert testimony of Dr. Markus Jakobsson. EX1003, ¶¶1-160.

Ground	Claims	Basis
1A	1-10 and 13-16	§103: Hartrell, Fan, and Rockwood
1B	11 and 12	§103: Hartrell, Fan, Rockwood, and Polyakov

Each reference pre-dates April 20, 2006, the earliest proclaimed priority date of the '244 Patent ("Critical Date"), and thus qualifies as prior art, as shown below.

Reference	Date	Section
Hartrell	12/02/2005 (filed)	102(e)
Fan	07/14/2005 (filed)	102(e)
Rockwood	09/01/2005 (filed)	102(e)
Polyakov	06/28/2005 (filed)	102(e)

III. THE '244 PATENT

A. Brief Description

The '244 Patent describes a “system and method for identifying an origin of suspected pestware activity on a computer.” EX1001, Abstract. FIGS. 2-4 of the '244 Patent largely relate to the collection of “historical logs” on a computer and the traversal of these historical logs to identify the “source” of suspected pestware. *See, e.g.*, EX1001, 6:1-9; 7:22-47; 8:5-31; EX1003, ¶43. However, as noted in the Abstract and Summary of the '244 Patent, the claims also focus on “establishing a time of interest relating to a suspicion of pestware on the computer.” *See* EX1001, Abstract, 2:19-26. Indeed, as will be described in the next section, the “time of interest” features of the independent claims were a particular focus of the Applicant’s arguments in differentiating the prior art and of the Examiner’s allowance of the claims. Section III.B, *infra*.

The '244 Patent primarily describes the provision and use of a “time of interest” with respect to FIG. 5. *See* EX1001, FIG. 5, 8:59-61 (“in this embodiment a time of interest is initially established based upon a suspicion that pestware has infected a computer”). The '244 Patent describes a couple of non-limiting examples of how a user might provide a time of interest. *See* EX1001, 8:61-9:4; EX1003, ¶44. “In some embodiments, for example, a user may establish the time of interest based upon events the user observed (e.g., pop-ups or a system

crash).” EX1001, 8:61-63. Alternatively, “the user is provide[d] with an alert in response to the heuristics module 110 identifying an activity that is suspicious and the user is given an option to initiate research at or around the time of interest.” EX1001, 9:1-4. According to the ’244 Patent’s description of FIG. 1, “once a time of interest is established, suspicious activity is identified on the computer based upon the time of interest.” EX1001, 9:9-11.

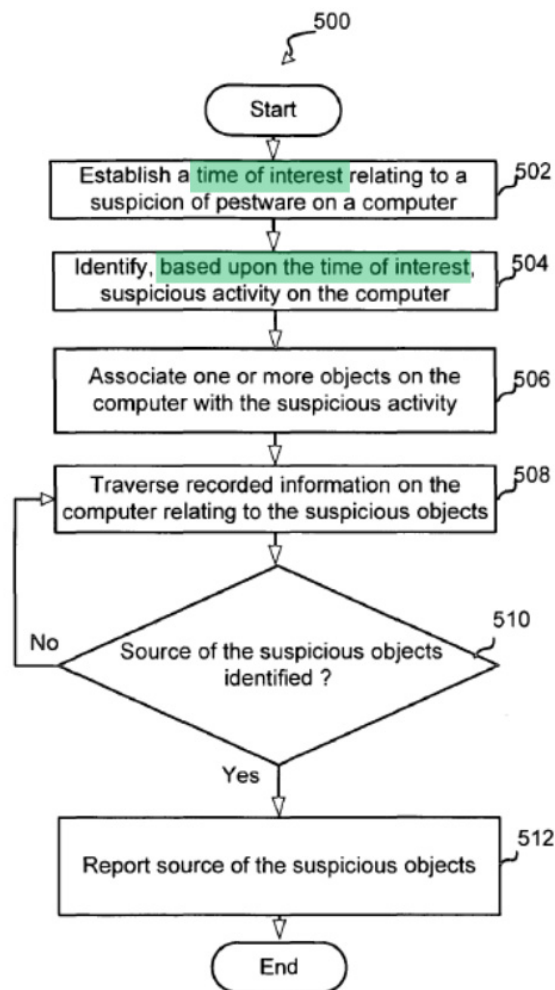


FIGURE 5

’244 Patent (EX1001), Highlighted FIG. 5

B. Prosecution History

The '244 Patent issued from U.S. Patent App. No. 11/408,145 (the '145 application). The claims were rejected over several references, including: U.S. Pub. No. 2004/0024864 (Porrás); U.S. Pub. No. 2005/0268112 (Wang); U.S. Pub. No. 2004/0225877 (Huang); U.S. Pub. No. 2007/0168694 (Maddaloni); and U.S. Pub. No. 2007/0094725 (Borders). EX1003, ¶¶45-47. In response, the Applicant amended the claims several times. *Id.*

Ultimately, though, the Examiner proposed and gained acceptance of an extensive Examiner's Amendment that led to allowance. EX1002, 37-40. The amendment included several substantive additions to the claims, including:

- a “kernel-mode driver”
- that the “time of interest” includes a “time interval”
- that a “timestamp” is issued after receiving the “time of interest,” and
- that identification of indicia of pestware is “in response to the issuance of the timestamp.”

Id. In his “Reasons for Allowance,” the Examiner explained that “[n]one of the prior art of records [sic], either taken by itself or in any combination, would have anticipated or made obvious at or before the time the application was filed where *a timestamp is issued after receiving the time of interest and the identifying of*

pestware is initiated by the issuing of timestamp wherein monitoring and generating an activity log is performed from the kernel-mode driver.” Id., 42.

However, these were common features of the prior art. For example, as will be explained below, Hartrell (EX1004) describes these features. EX1003, ¶49. Neither Hartrell nor any prior art like it was considered by the Examiner during original prosecution. *Id.*

C. Level of Ordinary Skill

A person of ordinary skill in the art relating to the subject matter of the '244 Patent as of April 20, 2006 (“POSITA”) would have possessed a bachelor’s degree in computer science, computer engineering, or an equivalent, as well as two years of industry experience and would have had a working knowledge of endpoint monitoring systems, software security analysis, and dynamic malware analysis. EX1003, ¶26. Greater education or greater experience could compensate for a deficiency in either of these criteria. EX1003, ¶26.

D. Claim Construction

All claim terms should be construed according to the *Phillips* standard. *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005); 37 C.F.R. § 42.100. Here, because the Challenged Claims are obvious under any reasonable interpretation and because “claim terms need only be construed to the extent necessary to resolve

the controversy,” no express constructions are required in this proceeding.

Wellman, Inc. v. Eastman Chem. Co., 642 F.3d 1355, 1361 (Fed. Cir. 2011).

Petitioner reserves the right to address any construction proposed by Patent Owner or the Board. Petitioner also reserves the right to pursue constructions in district court that are necessary to decide matters of infringement. Furthermore, Petitioner is not conceding that each challenged feature satisfies all statutory requirements such as 35 U.S.C. §112. As this is an IPR petition, Petitioner is pursuing prior art-based grounds. Petitioner is not waiving any arguments concerning other grounds that can only be raised in district court.

IV. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. [GROUND 1A] – CLAIMS 1-10 AND 13-16 WOULD HAVE BEEN OBVIOUS OVER HARTRELL, FAN, AND ROCKWOOD

The sections below provide a brief introduction to Hartrell, Fan, Rockwood and the combination thereof.²

1. Hartrell (EX1004)

Hartrell describes a “malware analysis system for automating cause and effect analysis of malware infections.” EX1004, Abstract. “Upon being informed of a suspected malware infection, the malware analysis system creates a time-bounded snapshot of the monitored activities that were conducted within a time

² Petitioner incorporates this discussion into Grounds 1A-1B.

frame prior to the notification of the suspected malware infection.” *Id.* “For example, the malware analysis system can provide a snapshot of the last ten minutes of the monitored activities (e.g., local system activity, network activity, etc.) prior to the detection of the infection.” *Id.*, [0014]. Hartrell describes analyzing the snapshot “to determine the ‘root cause’ or source of the infection.” *Id.*. EX1003, ¶54.

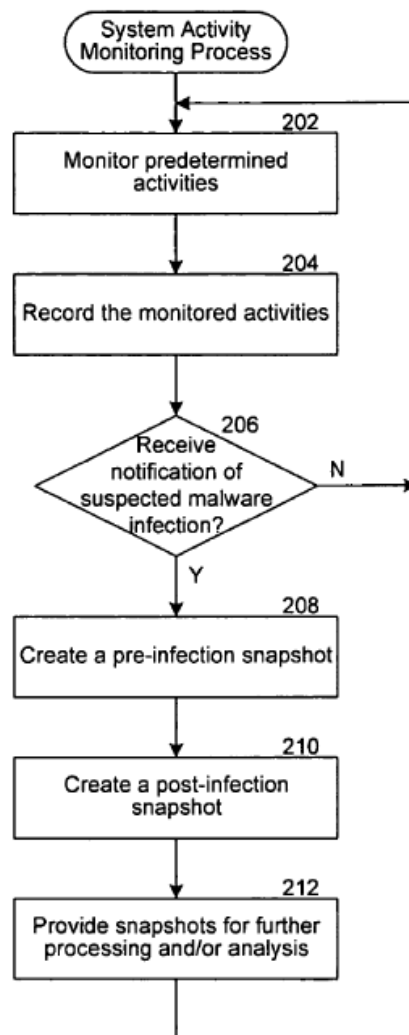


FIG. 2

Hartrell (EX1004), FIG. 2

In FIG. 2, Hartrell provides “a flow diagram that illustrates the processing of the system activity monitor component of the malware analysis system.” EX1004, [0007], [0027]. As shown in FIG. 2, Hartrell’s system “monitors predetermined activities on or about the computer system” (202) and records the monitored activity (204). EX1004, [0027]; EX1003, ¶55. The “system activity monitor component determines whether it received notification of a suspected malware infection” (206), and if “a notification of a suspected malware infection is received, then, in block 208, the system activity monitor component creates a pre-infection snapshot of the monitored activities.” *Id.*

“The system activity monitor component can use the report of the infection as the base time (e.g., t0) from which to create the time-bounded snapshots of recorded activities.” *Id.*, [0030]. Starting at this t0 timestamp of the reported infection, “the system activity monitor component may create a snapshot of the previous X seconds of the monitored activities that were recorded prior to receiving the notification event.” *Id.*, [0027]. “The snapshot of the activities can then be analyzed to determine the ‘root cause’ or source of the infection (i.e., the activities that led to the infection), including, for example, how the infected computer system was discovered, how the infected computer system was accessed, what operating system objects were manipulated, and the like.” *Id.*, [0014].

Hartrell describes that “the system activity monitor component may execute as a runtime process that may use any of a variety of *well-known* monitoring techniques *to monitor operating system* and/or network *activities*.” EX1004, [0027]. In other words, Hartrell does not limit the operation of the activity monitor component to any specific implementation, but instead leaves the details of its implementation to POSITAs. EX1003, ¶57. Fan provides details about how such an activity monitor may be configured to monitor operating system activities. *Id.*

2. Fan (EX1005)

Like Hartrell, Fan describes software that “monitors system operations and registers an event when an operation potentially characteristic of a malware infection occurs.” EX1005, 2:35-37. Fan’s software includes a “behavior monitor module 210”³ that “is real-time software including two functional kernel mode drivers that *monitor[] operating system operations*.” EX1005, 6:19-24. In other words, Fan describes a software module that performs the same type of monitoring as the activity monitor of Hartrell. *See* EX1004, [0027]; EX1003, ¶58.

³ Notably, Fan describes that behavior monitor module 210 preferably “operates in kernel mode.” EX1005, 6:19-25. For purposes of the combinations and mappings, Dr. Jakobsson and the Petition assume that behavior monitor module 210 operates in kernel mode consistent with Fan’s preferred embodiment. EX1003, ¶58.

Fan describes that “[m]odule 210 monitors and intercepts operating system operations such as file and registry access, process creation and process termination, and certain system APIs such as CreateProcess() CreateRemoteThread() and VirtualAlloc() [sic].” EX1005, 6:33-36. The module 210 can also intercept network traffic and events. *Id.*, 6:37-41; EX1003, ¶59. Fan teaches that some types of malware (e.g., rootkits) “hook into the operating system at the kernel level to hide their presence,” making them “very hard to detect.” *Id.*, 4:45-53. Because the behavior monitor module 210 includes two functional kernel mode drivers, it is “suitable for use when malware has infected a kernel mode driver.” *See id.* 3:10-13, 6:19-60.

3. The combination of Hartrell and Fan

Like the '244 Patent, Hartrell and Fan both describe systems for monitoring computer systems for suspected malware. *See, e.g.*, EX1004, Abstract (“A malware analysis system for automating cause and effect analysis of malware infections is provided. The malware analysis system monitors and records computer system activities.”); EX1005, 2:35-37 (“In one embodiment the present invention monitors system operations and registers an event when an operation potentially characteristic of a malware infection occurs.”); EX1001, 1:36-39; EX1003, ¶60.

As described above, Hartrell describes a system activity monitor component 102 that “provides run-time *monitoring of the operating system resources* for changes to the file system, common file formats, configurations (registry), network activities, use of common application program interfaces (APIs), or any other operating system object.” EX1004, [0019]. However, Hartrell does not limit or specify further implementation details for the operation of the system activity monitor component 102. Rather, Hartrell describes that “the system activity monitor component may execute as a runtime process that may use any of a variety of *well-known* monitoring techniques *to monitor operating system* and/or network *activities*.” EX1004, [0027]. Fan provides an example of one well-known technique for monitoring operating system activities. EX1003, ¶61.

Thus, a POSITA would have found it obvious to implement Hartrell’s system activity monitor component 102 using “functional kernel mode drivers,” as described by Fan. EX1003, ¶62. A POSITA would have been motivated to use “functional kernel mode drivers” to implement Hartrell’s system activity monitor component 102, because Fan teaches that they are “suitable for use when malware has infected a kernel mode driver,” whereas “[t]raditional cleanup software is unable to remove malware from such drivers.” EX1005, 3:10-13, 6:19-60; EX1003, ¶62. The kernel mode drivers of Fan are assumed to be “trusted software and thus can execute any instruction and reference any memory address.”

EX1005, 6:24-32. As a result, Fan's functional kernel mode drivers are able to provide access to the type of operating system data (e.g., the registry, network activities, and other operating system objects) to which Hartrell's system activity monitor component 102 requires access. *See* EX1004, [0019]; EX1005, 6:19-54; EX1003, ¶62. Further, a POSITA would have had a reasonable expectation of success in implementing this combination, because Hartrell describes that these techniques were "well known," and Fan gives specific implementation details for them. *See* EX1004, [0027]; EX1005, 6:19-54; EX1003, ¶62.

4. Rockwood (EX1006)

Rockwood describes a "system and method for intruder tracking using advanced correlation in a network security system." EX1006, 1:8-11. According to Rockwood, prior art systems would detect attacks by correlating "incoming data with stored data according to rules designed to detect specific patterns in network traffic, audit trails, and other data sources." EX1006, 1:22-26. However, these prior art systems did not "determine the likely identity of attackers." EX1006, 1:25-26. Thus, Rockwood describes a "method for correlating event information" that "reduces the number of operations required to correlate detected events with stored events, resulting in faster correlations," and that "enables an intrusion detection system to determine the likely identity of attackers." EX1006, 1:36-67.

Among other things, Rockwood describes a correlation engine 140 that “is operable to correlate detected events 110 to detect an attack” EX1006, 4:49-51. “Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 . . . to identify the source of an attack.” EX1006, 4:51-54. The correlation engine 140 utilizes a ruleset 152 “to process, correlate, aggregate, and/or filter event information of detected events 110.” EX1006, 5:16-20.

Rockwood also describes a console 50 that includes a graphical user interface 52 (GUI) (*see* EX1006, 5:39-59) through which “a user, such as an operator (not shown), may view and process the event information of detected events 110” (*see* EX1006, 5:27-34) and provide “proximity limits” used by the correlation engine 140 to correlate various detected events to identify an attack and attacker (*see* EX1006, 8:6-34). EX1003, ¶65. Proximity limits “define a portion of n-dimensional space surrounding” detected events and are used to correlate those events. *See* EX1006, 8:6-34. The proximity limits provided by the user through the console’s GUI may include, for example, a time period around a target event. *See* EX1006, 9:4-16. In this regard, Rockwood empowers the user of console 50 to provide input to the correlation engine 140 to refine the correlation process. EX1003, ¶65.

5. The Combination of Hartrell, Fan, and Rockwood

Hartrell describes that the “system activity monitor component can be notified of a malware infection.” EX1004, [0019]. “Upon receiving the notification, the system activity monitor component creates and provides *a time-bounded snapshot of activities that occurred before* and/or after *the notification of infection* to the other components of the malware analysis system.” *Id.* “For example, the system activity monitor component may create a snapshot of the previous X seconds of the monitored activities that were recorded prior to receiving the notification event.” *Id.*, [0027].

Hartrell describes that its malware analysis system also includes a commonality analysis component 106 that analyzes “the normalized activities in the pre-infection snapshot and/or the post infection snapshot to find any recurring activities” so that “the malware analysis system can tailor and provide a recommendation for responding to the detected infection.” EX1004, [0018]. One function of “the commonality analysis component tags similar activities as being suspicious.” EX1004, [0035]. However, Hartrell does not specifically describe or otherwise limit the manner in which the commonality analysis component determines activities are “similar.” EX1003, ¶67. Rather, Hartrell describes that “[o]ne skilled in the art will appreciate that the commonality analysis component may use any of a variety of well-known comparative analysis techniques to

identify statistically significant *commonalities*.” EX1004, [0035]. Thus, Hartrell leaves these implementation details to a POSITA. EX1003, ¶67.

Such a POSITA would have found it obvious to utilize the teachings of Rockwood when implementing Hartrell’s commonality analysis component. As described previously, Rockwood’s correlation engine 140 performs a similar function to Hartrell’s commonality analysis component (i.e., finding commonalities between detected events to identify attacks and attackers). *See* Section IV.A.5, *supra*; EX1003, ¶68. A POSITA would have found it obvious to employ various techniques of Rockwood’s correlation engine 140, including, for example, providing a user of the computing device on which Hartrell’s malware analysis system is implemented a GUI through which to specify proximity limits in order to define an n-dimensional space by which Hartrell’s commonality analysis component would be able to identify commonalities between detected events. EX1003, ¶68. As described above, the proximity limits provided by the user through the console’s GUI may include, for example, a time period around a target event for identifying related/common events. *See* EX1006, 9:4-16; EX1003, ¶68. This is consistent with Hartrell’s creation of a “snapshot of the previous X seconds of the monitored activities that were recorded prior to receiving” notification of a suspected malware infection. *See* EX1004, [0027]; EX1003, ¶68. Hartrell does not describe or limit who defines the period of its snapshots, yet Rockwood would

have taught a POSITA that it would have been obvious to allow a user of Hartrell's computing device a GUI through which to provide this definition. EX1003, ¶68.

A POSITA would have been motivated to utilize the teachings of Rockwood when implementing Hartrell's commonality analysis component, because Rockwood describes that its techniques "reduces the number of operations required to correlate detected events with stored events, resulting in faster correlations," and provides the controls that allow a user to "better . . . respond to attacks, preempt future attacks, and gather evidence against attackers." EX1006, 1:56-67. A POSITA would have recognized that a user of Hartrell's computing device would have been well positioned to identify suspicious activity on the computing device and the parameters around which a search ought to take place. EX1003, ¶69. Indeed, it was well known to allow a user, through a user interface, to indicate that they believe a malware infection had taken place and/or to alert a user of a possible infection and request their confirmation to take action. *See, e.g.*, EX1007, 52-61 ("Those skilled in the art and others will appreciate that programs designed to protect a computer from malware may initiate a scan for malware in many different circumstances. For example, existing programs in which the RootKit detector may be integrated, such as antivirus software, anti-spyware software, system utilities and the like, may be configured to perform a scan for malware 'on demand' *when*

a user or software system generates a command using a Graphical User

Interface or similar input system.”); EX1003, ¶69.

6. Element-by-Element Explanation of How Hartrell-Fan-Rockwood Combination Renders Challenged Claims Obvious

(i) Claim 1

[1pre] “A method for identifying an origin of suspected pestware activity on a computer, the method comprising:”

To the extent the preamble is deemed to be limiting, the Hartrell-Fan-Rockwood combination renders it obvious. EX1003, ¶¶77-78.

The '244 Patent describes that “[p]ersonal computers and business computers are continually attacked by trojans, spyware, and adware, collectively referred to as ‘malware,’ ‘spyware’ or ‘pestware.’” EX1001, 1:43-45. In other words, the '244 Patent uses the terms “malware” and “pestware” interchangeably. EX1003, ¶77.

Hartrell describes that “[a] difficulty with the information provided after an infection is detected by today’s anti-malware software is that system administrators are unable to determine the root cause or *origin* of that malicious code.” EX1004, [0003]. Hartrell describes a system that solves this problem. *See* EX1004, [0014]; EX1003, ¶78. Specifically, “[u]pon being informed of the detection of a *malware* infection or suspected infection, for example, by locally executing *anti-malware software*, the malware analysis system provides a snapshot (or evidence) of the

activities that were conducted within a specified time frame before the detected or suspected infection for analysis.” EX1004, [0014]. “The snapshot of the activities can then be *analyzed to determine the ‘root cause’ or source of the infection* (i.e., the activities that led to the infection), including, for example, how the infected computer system was discovered, how the infected computer system was accessed, what operating system objects were manipulated, and the like.” *Id.* In one example scenario, Hartrell describes that “[t]he malware analysis system performs the commonality analysis and discovers that each of these ten computer systems visited the same web site within one minute prior to the infection.” *Id.*, [0037].

[1a] “monitoring, with a kernel-mode driver, activity on the computer;”

The Hartrell-Fan-Rockwood combination renders [1a] obvious. EX1003, ¶¶79-80.

For example, Hartrell describes a system activity monitor component 102 that “provides run-time *monitoring of the operating system resources* for changes to the file system, common file formats, configurations (registry), network activities, use of common application program interfaces (APIs), or any other operating system object.” EX1004, [0019]. “The system activity monitor component may *run on and monitor the activity of a computer system*, such as, by way of example, a local desktop operating system, a server or network device.” *Id.*

As described in Sections IV.A.1-3 (which are incorporated here by reference), a POSITA would have found it obvious to implement Hartrell's system activity monitor component 102 using "functional kernel mode drivers," as described by Fan. EX1003, ¶80. Specifically, Fan describes a "behavior monitor module 210" that "is real-time software including *two functional kernel mode drivers*" and that "*monitors operating system operations.*" EX1005, 6:19-24. Fan describes that "[m]odule 210 monitors and intercepts operating system operations such as file and registry access, process creation and process termination, and certain system APIs such as CreateProcess() CreateRemoteThread() and VirtualAlloc()." EX1005, 6:33-36. The module 210 can also intercept network traffic and events. *Id.*, 6:37-41.

[1b] "generating an activity log on a file storage device of the computer from the kernel-mode driver;"

The Hartrell-Fan-Rockwood combination renders [1b] obvious. EX1003, ¶¶81-84.

The '244 Patent describes that "[t]he activity monitor 116 in several embodiments monitors activities on the computer and stores information about the activities in the activity log 121. EX1001, 5:21-23. The activity log 121 may "include information (e.g., historical logs) relating to events on the computer 100." EX1001, 3:34-40; 6:4-9. The '244 Patent also describes activity log 220 "may include information about processes running on the computer 100 and files

corresponding to the processes,” and/or may include “a history of API calls and respective times made by processes running on the protected computer.” EX1001, 6:9-16.

Hartrell describes that “[t]he computing device on which the malware analysis system is implemented may include a central processing unit, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), and *storage devices* (e.g., disk drives).” EX1004, [0024]. “While executing, the system activity monitor component *records the monitored activities in a data store*, which may be in memory, on physical media, or other logical data store.” EX1004, [0019]. In this regard, the records stored by Hartrell’s systems activity monitor are the activity logs described in the ’244 Patent. *See* EX1001, 5:21-32; EX1003, ¶82.

A POSITA would have understood or at least found it obvious that the record of monitored activities stored by the computer executing Hartrell’s activity monitor is an activity log. EX1003, ¶83. For example, Hartrell describes that:

The system activity monitor component may be configured to record information regarding the monitored activity, such as, by way of example: the identified operating system object or network object involved in the monitored activity (e.g., *file name*, socket, IP address, logical paths, etc.); the details of the change (e.g., create file, listen on network socket, etc.); if applicable, the details of the object prior to the change; the source(s) of the change (e.g., *process id*, user security

context, logical storage identity from which data originated, logical storage identity where the change occurred, network sources such as uniform resource locator (URL) or internet protocol (IP) address, the API call used to make the change, etc.); a date and time stamp in which the event occurred.

EX1004, [0019].

Indeed, Fan describes that “*a log is kept* of all events received” by its behavior monitor module, which “two functional kernel mode drivers that monitors operating system operations.” EX1005, 8:35-37, 6:19-21. Thus, it would have at least been obvious to a POSITA that, in combination with Fan’s functional kernel mode drivers, Hartrell’s activity monitor would generate an activity log on a file storage device of the computer from the kernel-mode driver in the same manner as Fan describes the operation of its behavior monitor module. EX1003, ¶84.

[1c] “receiving, from a user via an interface of the computer, a time of interest relating to a suspicion of pestware on the computer, wherein the time of interest includes a time interval;”

The Hartrell-Fan-Rockwood combination renders [1c] obvious. EX1003, ¶¶85-91.

The ’244 Patent describes the reception of a “time of interest” with respect to FIG. 5. EX1003, ¶85. In particular, “a time of interest is initially established based upon a suspicion that pestware has infected a computer.” EX1001, 8:59-61.

In some embodiments, for example, a user may establish the time of interest based upon events the user observed (e.g., pop-ups or a

system crash). In one embodiment, as discussed with reference to FIG. 1, in response to a user request, the time stamp module 112 may issue a time stamp and initiate research that is related to activity occurring at or around the time of the time stamp. In variations, the user is provide with an alert in response to the heuristics module 110 identifying an activity that is suspicious and the user is given an option to initiate research at or around the time of interest.

EX1001, 8:61-9:4. In other words, the '244 Patent provides various examples by which a user may provide a "time of interest," some of which simply involve a user approving "research at or around the time of interest." *Id.*; EX1003, ¶85. The '244 Patent also describes that the "time of interest may be established, for example, to include a time period before the time stamp is issued so that a search for suspicious activity is limited to a particular time period." EX1001, 9:11-14.

Hartrell describes that the "system activity monitor component can be notified of a malware infection." EX1004, [0019]. "Upon receiving the notification, the system activity monitor component creates and provides *a time-bounded snapshot of activities that occurred before* and/or after *the notification of infection* to the other components of the malware analysis system." *Id.* Hartrell teaches that "[t]he system activity monitor component can use *the report of the infection* as the base time (e.g., t0) from which to create the time-bounded snapshots of recorded activities." *Id.*, [0030]. "For example, the system activity monitor component may create a snapshot of the previous X seconds of the

monitored activities that were recorded prior to receiving the notification event.”

Id., [0027].

In other words, Hartrell describes that the system activity monitor component receives a time of interest that includes a time interval (i.e., the period of the snapshot, which can be “X seconds of the monitored activities that were recorded prior to receiving the notification event”). This is the same type of “time of interest” described in the ’244 Patent. *See* EX1001, 8:59-9:14 (describing that the “time of interest” is related to an alert of suspicious activity and that the “time of interest may be established, for example, to include a time period before the time stamp is issued so that a search for suspicious activity is limited to a particular time period.”); EX1003, ¶87. Furthermore, Hartrell describes that at least one purpose of this snapshot is to identify “certain reoccurring activities” that may be “*categorized* as either potentially normal or *suspicious* with a malware state.” EX1004, [0021], [0035]. In other words, in Hartrell’s own words, the snapshot of period X seconds is related to the categorization of certain activities that are “*suspicious* with a malware state.” *Id.*; EX1003, ¶87.

Under PO’s own interpretation, a POSITA would have understood or at least found it obvious that Hartrell’s use of its snapshot to categorize certain activities that are “*suspicious* with a malware state” means that the period of the snapshot (i.e., the claimed “time of interest”) is related “to a suspicion of pestware on the

computer,”⁴ as recited in [1c]. EX1003, ¶88. In claim construction briefing, PO contends that “terms like ‘suspicion,’ ‘suspicious,’ or ‘suspected’ referring to pestware” is a well-understood concept and requires no construction. EX1015 (Responsive Markman Brief), 14-15. For example, PO suggests a “suspected pestware object” (a term used in related U.S. Patent No. 8,719,932⁵) is simply the object identified ‘based upon the activity’”—referring to the monitored activity in the limitations above. *Id.* PO argues that the ’244 Patent uses “the ‘suspicious’ terms in similar ways. *Id.* PO points to examples in the specification that it contends support its interpretation. *Id.*, 16-17 (citing ’932 Patent, 4:58-66 and characterizing disclosure as “identifying a suspected pestware object by analyzing activity based on weighted factors associated with pestware activity and

⁴ Petitioner proposes the term “suspicion of pestware” is indefinite in litigation and reserves its right to maintain this proposal. EX1014 (Opening Markman Brief), 13-15. For the purposes of this IPR, Petitioner assumes this term is definite and applies PO’s purported plain meaning. EX1015 (Responsive Markman Brief), 14-18.

⁵ As PO explains in its briefing, U.S. Patent No. 8,719,932 and the ’244 Patent challenged in this IPR share “the same title and figures, and substantially the same specification.” EX1015, 10.

determining that one or more of the weighted factors exceeds a threshold”). PO contends that “[a] POSITA would understand that activities that exceed the threshold may be... ‘suspected’ of pestware.” *Id.*, 17. This is effectively the same process utilized by Hartrell when evaluating the “certain reoccurring activities” in the snapshot. EX1004, [0021], [0035]; EX1003, ¶88.

With respect to the time-bounded snapshot created by Hartrell’s system activity monitor component, Hartrell generally describes receiving a notification of malware infection, but does not explicitly describe that this notification or the period of the snapshot created as a result of the notification are received “from a user via an interface of the computer.” *See* EX1004, [0019], [0027]; EX1003, ¶89. However, as described in Sections IV.A.4-5, *supra*, a POSITA would have found it obvious in light of the teachings of Rockwood that the period of the snapshot and initiation of a scan would be received “from a user via an interface of the computer,” as recited in [1c].

As described in Section IV.A.4, *supra*, Rockwood describes a “system and method for intruder tracking using advanced correlation in a network security system.” EX1006, 1:8-11. Rockwood describes a correlation engine 140 that “is operable to correlate detected events 110 to detect an attack” EX1006, 4:49-51. Rockwood also describes a console 50 that includes a graphical user interface 52 (GUI) (*see* EX1006, 5:39-59) through which “a user, such as an operator (not

shown), may view and process the event information of detected events 110” (*see* EX1006, 5:27-34) and provide “proximity limits” used by the correlation engine 140 to correlate various detected events to identify an attack and attacker (*see* EX1006, 8:6-34). EX1003, ¶90. Proximity limits “define a portion of n-dimensional space surrounding” detected events and are used to correlate those events. *See* EX1006, 8:6-34. The proximity limits provided by the user through the console’s GUI may include, for example, a time period around a target event. *See* EX1006, 9:4-16; EX1003, ¶90.

Based on these teachings, a POSITA would have found it obvious to, for example, provide a user of the computing device on which Hartrell’s malware analysis system is implemented a GUI through which to specify proximity limits in order to define an n-dimensional space by which Hartrell’s commonality analysis component would be able to identify commonalities between detected events. EX1003, ¶91. As described above, the proximity limits provided by the user through the console’s GUI may include, for example, a time period around a target event for identifying related/common events. *See* EX1006, 9:4-16; EX1003, ¶91. This is consistent with Hartrell’s creation of a “snapshot of the previous X seconds of the monitored activities that were recorded prior to receiving” notification of a suspected malware infection. *See* EX1004, [0027]; EX1003, ¶91. Hartrell does not describe or limit who defines the period of its snapshots, yet Rockwood would

have taught a POSITA that it would have been obvious to allow a user of Hartrell's computing device a GUI through which to provide this definition. EX1003, ¶91.

Accordingly, it would have been obvious to a POSITA that the period of the snapshot (i.e., the "time of interest") received by Hartrell's system activity monitor component is received "from a user via an interface of the computer." EX1003, ¶91.

[1d] "issuing a timestamp after receiving the time of interest;"

The Hartrell-Fan-Rockwood combination renders [1d] obvious. EX1003, ¶¶92-93.

As described with respect to [1c], Hartrell describes that the "system activity monitor component can be notified of a malware infection." EX1004, [0019].

"Upon receiving the notification, the system activity monitor component creates and provides a time-bounded snapshot of activities that occurred before and/or after the notification of infection to the other components of the malware analysis system." *Id.* Hartrell teaches that "[t]he system activity monitor component can use the report of the infection *as the base time (e.g., t0) from which to create the time-bounded snapshots of recorded activities.*" *Id.*, [0030]. In other words, the "base time (e.g., t0)" is a timestamp that is issued after it receives a notification of a potential malware infection. EX1003, ¶92.

Hartrell describes a categorization component 104 that “normalizes the snapshot data for use by the commonality analysis component.” EX1004, [0019]-[0020]. FIG. 6 illustrates an example of the resulting normalized data, which includes a “time field [that] specifies, *relative to ‘t0’ (i.e., the time at which the notification of the malware infection is received by the malware analysis system)*, the time the activity identified in the corresponding recorded activity field was recorded by the system activity monitor component of the malware analysis system.” EX1004, [0032]. A POSITA would have recognized or at least found it obvious that to display monitored and stored activities “relative to ‘t0’”, the system would need to issue and store t0 at the time it receives the notification of the malware infection. EX1003, ¶93. Thus, it would have at least been obvious to a POSITA that Hartrell’s system activity monitor component issues a timestamp (e.g., “base time (e.g., t0)” and/or the “time field” included with each of the normalized data) after receiving the time of interest. EX1003, ¶93.

[1e] “identifying, based upon the time of interest, indicia of pestware, wherein the identifying is initiated by the issuing the timestamp;”

The Hartrell-Fan-Rockwood combination renders [1e] obvious. EX1003, ¶¶94-96.

For example, as described with respect to [1d], Hartrell describes a categorization component 104 that “normalizes the snapshot data for use by the commonality analysis component.” EX1004, [0019]-[0020]. “In some

embodiments, the categorization component may map the normalized activities along a malware state model, *with the initial infection notification as the reference* for pre- and post-infection activities.” EX1004, [0020]. In other words, the creation of the snapshot and the subsequent processing of the snapshot by the categorization component is initiated by the received notification of a suspected malware infection and issuance of the base time t_0 . *See* EX1004, [0027], [0029]-[0030]; EX1003, ¶94.

Hartrell describes that the “the categorization component maps each normalized activity to a malware state as provided by a malware state model.” EX1004, [0029]. The “malware states comprise a propagation state 402, an activation state 404, a carrier state 406, an execution state 408, and a persistence state 410.” EX1004, [0030]. “The categorization component can use the table of mappings of normalized actions to intelligently map each normalized activity to a malware state.” EX1004, [0031]. “FIG. 5 illustrates a table 500 of example mappings of normalized actions to pre- and post-notification malware states.” *Id.* Hartrell teaches that, “if the normalized action is contained in the pre-infection snapshot, the normalized action is assigned the malware state that is specified in the corresponding pre-notification malware state field,” and that, “if the normalized action is contained in the post-infection snapshot, the normalized action is assigned the malware state that is specified in the corresponding post-

notification malware state field.” *Id.* In other words, the categorization component identifies, based upon the time of interest (i.e., before or after the notification of infection), indicia of pestware (i.e., assigned malware state). EX1003, ¶95.

Additionally, Hartrell describes that the commonality analysis component of the malware analysis system “compares the normalized activities that are recorded for a particular malware infection to previously stored normalized activities for the same malware infection,” and “tags similar activities as being suspicious.”

EX1004, [0035]. Thus, the commonality analysis component also identifies indicia of pestware (i.e., similar activities as being suspicious). EX1003, ¶96.

Because the normalized activities compared by the commonality analysis component come from snapshots (i.e., a list of activities from a period of time before a notification of infection), the comparison performed by the commonality analysis component is “based upon the time of interest.” *See* EX1004, [0018], [0031], [0035]; EX1003, ¶96.

[1f] “accessing, using a hardware processor of the computer, at least a portion of a recorded history of externally networked sources that the computer received files from”

The Hartrell-Fan-Rockwood combination renders [1f] obvious. EX1003, ¶¶97-100.

For example, Hartrell describes that “the system activity monitor component may execute as a runtime process that may use any of a variety of well-known

monitoring techniques to monitor operating system and/or **network activities**.”

EX1004, [0027]. Specifically, the “system activity monitor component may be configured to record information regarding the monitored activity, such as, by way of example: . . . the details of the change (e.g., **create file**, listen on network Socket, etc.); [and] . . . the **source(s) of [a] change** (e.g., . . . **network sources** such as uniform resource locator (URL) or internet protocol (IP) address, the API call used to make the change, etc.)” EX1004, [0019].

Once Hartrell’s system receives a notification of malware infection, a snapshot is created and “the categorization component normalizes the activities contained in the snapshots provided by the system activity monitor component into predetermined set of common categories of system or **network events** (also referred to as ‘normalized actions’).” EX1004, [0029]. For example, one of the example of normalized actions illustrated in FIG. 6 is “File Transfer.” EX1004, FIG. 6. In other words, the categorization component accesses a portion of a recorded history of externally networked sources that the computer received files from. EX1003, ¶98.

Hartrell gives an example of the malware analysis system “monitoring all activities that modify local data stores (e.g., file system, registry, databases, etc.) and network activities (e.g., TCP/IP connections, web site visits, etc.).” EX1004, [0037]. In this example, the “malware analysis system performs the commonality

analysis and discovers that each of [] ten computer systems visited the same web site within one minute prior to the infection.” *Id.* A POSITA would have recognized or at least found it obvious that for a computer to “visit” a web site, it would request and receive data from the web site (e.g., an HTML file). EX1003, ¶99.

Hartrell describes that “the system activity monitor component may execute on all of the computer systems that are to be monitored, and one or more of the categorization component, the commonality analysis component, and the response recommendation generator *may execute on one of the aforementioned computer systems.*” EX1004, [0023]. In other words, Hartrell describes that a single computer may execute all four components (i.e., a system activity monitor component 102, a categorization component 104, a commonality analysis component 106, and a response recommendation generator 108). *Id.*; EX1003, ¶100. Furthermore, “[t]he computing device on which the malware analysis system is implemented *may include a central processing unit*, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), and storage devices (e.g., disk drives).” EX1004, [0024]. Thus, the categorization component 104 and a commonality analysis component 106 accessing the network activity contained in the snapshot created by the system

activity monitor component would have been accessed “using a hardware processor of the computer,” as recited in [1f]. EX1003, ¶100.

[1g] “so as to identify, based at least in part upon the identified indicia of pestware, a reference to an identity of an externally networked source that is suspected of originating pestware”

The Hartrell-Fan-Rockwood combination renders [1g] obvious. EX1003, ¶¶101-102.

For example, as described with respect to [1f], Hartrell describes that the “system activity monitor component may be configured to record information regarding the monitored activity, such as, by way of example:. . . the details of the change (e.g., *create file*, listen on network Socket, etc.); [and] . . . the *source(s) of [a] change* (e.g., . . . *network sources* such as uniform resource locator (URL) or internet protocol (IP) address, the API call used to make the change, etc.)”

EX1004, [0019]. Hartrell describes that the “snapshot of the activities can then be *analyzed to determine the ‘root cause’ or source of the infection* (i.e., the activities that led to the infection), including, for example, how infected computer system was discovered, how the infected computer system was accessed, what operating system objects were manipulated, and the like.” EX1004, [0014].

Hartrell gives an example of the malware analysis system “monitoring all activities that modify local data stores (e.g., file system, registry, databases, etc.) and network activities (e.g., TCP/IP connections, web site visits, etc.).” EX1004,

[0037]. In this example, the “malware analysis system performs the commonality analysis and *discovers that each of [] ten computer systems visited the same web site within one minute prior to the infection.*” *Id.* A POSITA would have recognized or at least found it obvious that, in this example, the commonality analysis component is identifying, based at least in part upon the identified indicia of pestware, a reference to an identity of an externally networked source that is suspected of originating pestware. EX1003, ¶102.

[1h] “wherein the recorded history of externally networked sources is stored on the file storage device”

The Hartrell-Fan-Rockwood combination renders [1h] obvious. EX1003, ¶¶103-104.

For example, as noted with respect to [1f], Hartrell describes that a single computer may execute all four components (i.e., a system activity monitor component 102, a categorization component 104, a commonality analysis component 106, and a response recommendation generator 108) of the malware analysis system. *Id.*; EX1003, ¶103. Furthermore, Hartrell describes that “[t]he computing device on which the malware analysis system is implemented *may include* a central processing unit, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), and *storage devices (e.g., disk drives).*” EX1004, [0024]. Thus, when Hartrell describes that “the system activity monitor component *records the monitored activities in a data store*, which

may be in memory, on physical media, or other logical data store,” a POSITA would have understood or at least found it obvious that the monitored activities recorded by the system activity monitor component are store on the file storage device of the computer executing the malware analysis system. EX1004, [0019]; EX1003, ¶103.

Hartrell describes that “the system activity monitor component may execute as a runtime process that may use any of a variety of well-known monitoring techniques to monitor operating system and/or *network activities*.” EX1004, [0027]. Thus, the monitored activities recorded by the system activity monitor component that are stored on the file storage device of the computer executing the malware analysis system include the *network activities* that make up the recorded history of externally networked sources, which were described with respect to [1f]. EX1003, ¶104.

(ii) **Claim 2**

[2a] “The method of claim 1, wherein the identifying includes: accessing at least one log of historical events on the computer”

The Hartrell-Fan-Rockwood combination renders [2a] obvious. EX1003, ¶¶105-108.

For example, Hartrell describes that a system activity monitor component 102 that “provides run-time *monitoring of the operating system resources* for changes to the file system, common file formats, configurations (registry), network

activities, use of common application program interfaces (APIs), or any other operating system object.” EX1004, [0019]. As described with respect to [1h], Hartrell describes that a single computer may execute all four components (i.e., a system activity monitor component 102, a categorization component 104, a commonality analysis component 106, and a response recommendation generator 108) of the malware analysis system. *Id.*; EX1003, ¶106. Furthermore, Hartrell describes that “[t]he computing device on which the malware analysis system is implemented **may include** a central processing unit, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), and **storage devices (e.g., disk drives).**” EX1004, [0024]. Thus, when Hartrell describes that “the system activity monitor component **records the monitored activities in a data store**, which may be in memory, on physical media, or other logical data store,” a POSITA would have understood or at least found it obvious that the monitored activities recorded by the system activity monitor component are stored on the file storage device of the computer executing the malware analysis system. EX1004, [0019]; EX1003, ¶106. As described above with respect to [1b], a POSITA would have at least found it obvious that the records stored by Hartrell’s systems activity monitor are the activity logs described in the ’244 Patent. EX1003, ¶106.

Accordingly, when the system activity monitor component 102 received the notification of malware infection and generates the snapshot, the system activity

monitor component is accessing the records of the monitored activities stored in the file storage device of the computer executing the malware analysis system, and passing the generated snapshot to categorization component 104. *See* EX1004, [0019]-[0020], [0023]-[0024], [0027]-[0029]; EX1003, ¶107.

As described with respect to [1e], the categorization component 104 “normalizes the snapshot data for use by the commonality analysis component,” which includes identifying, based upon the time of interest (i.e., before or after the notification of infection), indicia of pestware (i.e., assigned malware state). EX1004, [0019]-[0020]; EX1003, ¶108. Because the categorization component 104 makes this identification based on the snapshot data received from the system activity monitor component, the identifying includes accessing at least one log of historical events on the computer (i.e., the snapshot). EX1003, ¶108.

[2b] “locating at least one event in the at least one log that corresponds to the time of interest; and”

The Hartrell-Fan-Rockwood combination renders [2b] obvious. EX1003, ¶109.

For example, as described with respect to [1e], Hartrel describes that “the categorization component *maps each normalized activity to a malware state* as provided by a malware state model.” EX1004, [0029]. In other words, the categorization component locates and maps each of the events in the snapshot. EX1003, ¶109. And the entire snapshot corresponds to the time of interest. *See*

EX1004, [0014] (“the malware analysis system provides a snapshot (or evidence) of the activities that were conducted within a specified time frame before the detected or suspected infection for analysis.”).

[2c] “determining that the at least one event is associated with the indicia of pestware.”

The Hartrell-Fan-Rockwood combination renders [2c] obvious. EX1003, ¶¶110-111.

As described with respect to [1e], Hartrell describes that the “the categorization component maps each normalized activity to a malware state as provided by a malware state model.” EX1004, [0029]. The “malware states comprise a propagation state 402, an activation state 404, a carrier state 406, an execution state 408, and a persistence state 410.” EX1004, [0030]. “The categorization component can use the table of mappings of normalized actions to intelligently map each normalized activity to a malware state.” EX1004, [0031]. In other words, the categorization component identifies indicia of pestware (i.e., assigned malware state).

Additionally, Hartrell describes that the commonality analysis component of the malware analysis system “compares the normalized activities that are recorded for a particular malware infection to previously stored normalized activities for the same malware infection,” and “tags similar activities as being suspicious.”

EX1004, [0035]. Thus, the commonality analysis component also identifies indicia of pestware (i.e., similar activities as being suspicious). EX1003, ¶111.

(iii) **Claim 3**

[3] “The method of claim 2, wherein the determining that the at least one event is associated with the indicia of pestware includes following references in the at least one log to a suspected pestware object.”

The Hartrell-Fan-Rockwood combination renders [3] obvious. EX1003, ¶¶112-115.

Hartrell describes that the “the commonality analysis component compares the normalized activities that are recorded for a particular malware infection to previously stored normalized activities for the same malware infection.” EX1004, [0035]. As part of this process, “the commonality analysis component tags similar activities as being suspicious,” and “assigns a malware state to the suspicious activity.” *Id.* Hartrell describes an example scenario implementing this process in which:

The malware analysis system [] normalizes the snapshot data . . . , and conducts commonality analysis on the ten provided snapshots to find any recurring activities. The malware analysis system performs the commonality analysis and *discovers that each of these ten computer systems visited the same web site* within one minute prior to the infection.

EX1004, [0037]. A POSITA would have recognized or at least found it obvious that the commonality analysis would have included “following references in the at

least one log to a suspected pestware object,” because the commonality analysis would have considered the various progressive states assigned by the categorization component to find the state in which the source of the malware infection. EX1003, ¶115. This approach is consistent with Hartrell’s description of the malware state model (*see* EX1004, [0030]), and Hartrell’s objective of analyzing the snapshot “to determine the ‘root cause’ or *source of the infection (i.e., the activities that led to the infection)*” (*id.*, [0014]). EX1003, ¶115.

(iv) **Claim 4**

[4] “The method of claim 3, wherein following the references includes following at least one reference in the activity log from the event to a suspected pestware process.”

The Hartrell-Fan-Rockwood combination renders [4] obvious. EX1003, ¶¶116-118.

As described with respect to [3], Hartrell describes analyzing the snapshot “to determine the ‘root cause’ or *source of the infection (i.e., the activities that led to the infection)*,” which a POSITA would have recognized or at least found obvious included “following references in the at least one log to a suspected pestware object.” EX1003, ¶117. Hartrell further describes that:

The system activity monitor component may be configured to record information regarding the monitored activity, such as, by way of example: *the identified operating system object* or network object involved in the monitored activity (e.g., *file name*, socket, IP address,

logical paths, etc.); the details of the change (e.g., create file, listen on network socket, etc.); if applicable, the details of the object prior to the change; *the source(s) of the change* (e.g., *process id*, user security context, logical storage identity from which data originated, logical storage identity where the change occurred, network sources such as uniform resource locator (URL) or internet protocol (IP) address, the API call used to make the change, etc.); a date and time stamp in which the event occurred.

EX1004, [0019]. Accordingly, Hartrell describes that the source of the change can be a process, and when it does, the snapshot will include a `process_id` for that process. *Id.*; EX1003, ¶118. In these instances, when Hartrell describes analyzing the snapshot “to determine the ‘root cause’ or *source of the infection (i.e., the activities that led to the infection)*,” Hartrell is describing that following the references includes following at least one reference in the activity log from the event to a suspected pestware process. EX1003, ¶118.

(v) **Claim 5**

[5] “The method of claim 2, wherein the at least one log includes at least one log selected from the group consisting of an activity log, a browser history, browser cache, browser settings, operating system settings, an event log, a debugging log, a firewall log, file information and monitoring software logs.”

The Hartrell-Fan-Rockwood combination renders [5] obvious. EX1003, ¶¶119-122.

As described with respect to [1b], a POSITA would have understood or at least found obvious that the record of monitored activities stored by the computer

executing Hartrell's activity monitor is an activity log. EX1003, ¶120. For example, Hartrell describes that:

The system activity monitor component may be configured to record information regarding the monitored activity, such as, by way of example: the identified operating system object or network object involved in the monitored activity (e.g., *file name*, socket, IP address, logical paths, etc.); the details of the change (e.g., create file, listen on network socket, etc.); if applicable, the details of the object prior to the change; the source(s) of the change (e.g., *process id*, user security context, logical storage identity from which data originated, logical storage identity where the change occurred, network sources such as uniform resource locator (URL) or internet protocol (IP) address, the API call used to make the change, etc.); a date and time stamp in which the event occurred.

EX1004, [0019]. This is a record of activities performed on the computer, and is therefore a "log," as a POSITA would have understood that term. EX1003, ¶121. Indeed, the list of activities monitored and recorded by Hartrell's activity monitor are consistent with the logs described in the '244 Patent. EX1001, 6:9-13 ("an activity log 220, which may be generated by an activity monitor . . . and may include information about *processes* running on the computer 100 and *files* corresponding to the processes").

Because the stored record of activities monitored by Hartrell's activity monitor is an activity log, the snapshot created by Hartrell's activity monitor,

which is just a subset of the total activity log (e.g., “the previous X seconds of the monitored activities that were recorded prior to receiving the notification event”), is also an activity log. EX1003, ¶122.

(vi) **Claim 6**

[6] “The method of claim 1, wherein the user input⁶ is in response to an alert provided to the user regarding a detected suspicious activity on the computer.”

The Hartrell-Fan-Rockwood combination renders [6] obvious. EX1003, ¶¶123-125.

As described with respect to [1c], Hartrell does not explicitly describe that the period of the snapshot is received “from a user via an interface of the computer.” Rather, Hartrell generally describes receiving a notification of malware infection, and then provides non-limiting examples from where it may be received. *See* EX1004, [0019], [0027]; EX1003, ¶124. For the reasons explained with respect to [1c], a POSITA would have found it obvious in light of the teachings of Rockwood that the period of the snapshot and initiation of a scan

⁶ Petitioner notes that “the use input” lack antecedent basis. However, for purposes of this petition alone, Petitioner assumes that “the user input” refers to the reception of the time of interest from the user as recited in [1c].

would be received “from a user via an interface of the computer,” as recited in [1c]; EX1003, ¶124.

Furthermore, Fan describes that “*a user* or administrator of the computer system in question makes a determination or realizes that malware has infected the computer system or is present on the computer system, and desires to take action to remove the malware.” EX1005, 11:32-37. Indeed, it was well known to either allow a user, through a user interface, to indicate that they believe a malware infection had taken place or to alert a user of a possible infection and request their confirmation to take action. *See, e.g.*, EX1006, 52-61 (“Those skilled in the art and others will appreciate that programs designed to protect a computer from malware may initiate a scan for malware in many different circumstances. For example, existing programs in which the RootKit detector may be integrated, such as antivirus software, anti-spyware software, system utilities and the like, may be configured to perform a scan for malware ‘on demand’ *when a user or software system generates a command using a Graphical User Interface or similar input system.*”); EX1003, ¶125. Accordingly, it would have been obvious to a POSITA that the user input is in response to an alert provided to the user regarding a detected suspicious activity on the computer, as recited in claim 6. EX1003, ¶125.

(vii) **Claim 7**

[7] **“The method of claim 1, including: reporting the identity of the externally networked source to an externally networked pestware research entity so as to**

enable the pestware research entity to research whether the externally networked source is a source of pestware”

The Hartrell-Fan-Rockwood combination renders [7] obvious. EX1003, ¶¶126-129.

Hartrell describes that, “[i]n some embodiments, the malware analysis system can provide the snapshot of the activities to a *remote* (e.g., centralized) data store.” EX1004, [0014]. “For example, the system activity monitor component may provide the snapshots to a *remote data store* that is accessible by the other components of the malware analysis system and/or an administrator.” EX1004, [0027]. And while Hartrell provides several examples of where various components and their analysis may take place, Hartrell clarifies that “[o]ne skilled in the art will appreciate that the malware analysis system may be deployed in other ways.” EX1004, [0023]. Based on these teachings, a POSITA would have found it obvious to “report the identity of the externally networked source to an externally networked pestware research entity so as to enable the pestware research entity to research whether the externally networked source is a source of pestware,” as this would simply substitution of the user’s computer with an externally networked pestware research entity to achieve the same goal sought by Hartrell: “determine the ‘root cause’ or source of the infection (i.e., the activities that led to the infection).” EX1004, [0014]; EX1003, ¶128. Indeed, it was well known to the POSITA to configured malware detection systems like the one

described by Hartrell to share its analysis with third parties that a POSITA would have at least found obvious are externally networked pestware research entities in order to allow for the aggregation of threat data and broader dissemination of remediation solutions. *See, e.g.*, EX1007, 9:24-35 (“RootKit sample is reported to a ***remote computer associated with a trusted entity*** at block 318. . . . In this instance, data obtained that describes the state of the computer infected with RootKit is reported to a trusted entity so that developers may perform an analysis of the data.”); EX1008, 9:55-62 (“In an alternative embodiment of the present invention, the aggregation routine 310 maintains an Application Program Interface (API) that allows ***third-party providers*** to report suspicious events. In this instance, ***anti-malware services or other event detection system created by third parties may ‘plug-in’*** to the aggregation routine 310 and report Suspicious events by issuing one or more API calls.”); EX1009, 3:49-54 (“the virus database may be accessible via a network interface and maintained by a third party, such as, for example, an independent software vendor (ISV), independent BIOS vendor (IBV), an original equipment manufacturer (OEM), or an independent hardware vendor (IHV); however, these are merely a few non-limiting examples”); EX1003, ¶129.

(viii) **Claim 8**

[8] “The method of claim 1, wherein the externally networked source is identified by an identifier selected from the group consisting of an I.P. address and a URL.”

The Hartrell-Fan-Rockwood combination renders [8] obvious. EX1003, ¶¶130-132.

As described with respect to [1f], Hartrell describes that “the system activity monitor component may execute as a runtime process that may use any of a variety of well-known monitoring techniques to monitor operating system and/or *network activities*.” EX1004, [0027]. Specifically, the “system activity monitor component may be configured to record information regarding the monitored activity, such as, by way of example: . . . the details of the change (e.g., *create file*, listen on network Socket, etc.); [and] . . . the *source(s) of [a] change* (e.g., . . . *network sources such as uniform resource locator (URL) or internet protocol (IP) address*, the API call used to make the change, etc.)” EX1004, [0019].

Hartrell gives an example of the malware analysis system “monitoring all activities that modify local data stores (e.g., file system, registry, databases, etc.) and network activities (e.g., TCP/IP connections, web site visits, etc.).” EX1004, [0037]. In this example, the “malware analysis system performs the commonality analysis and discovers that each of [] ten computer systems *visited the same web site* within one minute prior to the infection.” *Id.* In light of Hartrell’s description

of the contents of the snapshot, a POSITA would have recognized or at least found it obvious that the identified web site is an externally networked sources that would be identified by uniform resource locator (URL) or internet protocol (IP) address.

See EX1004, [0019]; EX1003, ¶132.

(ix) **Claim 9**

[9pre]

See [1pre] of Ground 1A.⁷

[9a] **“a hardware processor”**

The Hartrell-Fan-Rockwood combination renders [9a] obvious. EX1003, ¶136.

For example, Hartrell describes that a single computer may execute all four components (i.e., a system activity monitor component 102, a categorization component 104, a commonality analysis component 106, and a response recommendation generator 108) of the malware analysis system. EX1004, [0023]; EX1003, ¶130. Furthermore, Hartrell describes that “[t]he computing device on which the malware analysis system is implemented *may include a central processing unit*, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), and storage devices (e.g., disk drives).” EX1004, [0024].

⁷ Citations of this form incorporate by reference the text from the cited section.

[9b] “a file storage device coupled to the hardware processor;”

The Hartrell-Fan-Rockwood combination renders [9b] obvious. EX1003, ¶137.

For example, Hartrell describes that a single computer may execute all four components (i.e., a system activity monitor component 102, a categorization component 104, a commonality analysis component 106, and a response recommendation generator 108) of the malware analysis system. EX1004, [0023]; EX1003, ¶137. Furthermore, Hartrell describes that “[t]he computing device on which the malware analysis system is implemented *may include* a central processing unit, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), and *storage devices (e.g., disk drives).*” EX1004, [0024].

[9c.i] “a timestamp module configured to be executed by the hardware processor, to generate a time of interest including a time interval in response to detected activity on the computer that is indicative of pestware,”

See [1c] & [1d] of Ground 1A.

[9c.ii] “issuing a timestamp after receiving the time of interest;”

See [1d] of Ground 1A.

[9c.iii] “to initiate, in response to the issuance of the timestamp, identification of indicia of pestware;”

See [1e] of Ground 1A.

[9d] “a research portion configured to be executed by the hardware processor and to access at least one log on a file storage device of the computer to relate

the time of interest to at least one externally networked source of potential pestware activity on the computer;”

See [1f] & [1g] of Ground 1A.

[9e] “a kernel-mode driver configured to monitor activity on the computer and to generate an activity log that is included within the at least one log; and”

See [1a] & [1b] of Ground 1A.

[9f] “a reporting portion configured to be executed by the hardware processor and to generate a report that identifies the at least one externally networked source of potential pestware activity.”

The Hartrell-Fan-Rockwood combination renders [9f] obvious. EX1003, ¶¶143-144.

For example, Hartrell describes that the malware analysis system 100 includes a response recommendation generator 108 that “processes the information provided by the commonality analysis component and generates a recommendation for responding to the malware infection.” EX1004, [0019], [0022]. Specifically, “the response recommendation generator identifies a suspicious normalized activity,” such as an “activity that was tagged as being suspicious by the commonality analysis component.” *Id.*, [0036]. Next, “the response recommendation generator generates a recommendation or multiple recommendations for responding to the suspicious normalized activity.” *Id.* The “response recommendation generator can provide the recommended response to an administrator.” *Id.*

In an example scenario described by Hartrell, the “malware analysis system performs the commonality analysis and discovers that each of these ten computer systems visited the same web site.” *Id.*, [0037]. As a result, the response recommendation generator “may alert the system operator of all ten infected computer systems to the pattern, and provide a recommendation to block this web site at their organization’s egress points to prevent future infections.” *Id.* A POSITA would have recognized or found obvious that these response recommendations are reports that identify the at least one externally networked source of potential pestware activity, as recited in [9f]. EX1003, ¶144.

(x) Claim 10

[10a]-[10c]

See [2a]-[2c] of Ground 1A.

(xi) Claim 13

[13]

See [5] of Ground 1A.

(xii) Claim 14

[14] “The system of claim 9, wherein the timestamp module is further configured to generate the time of interest as a point in time.”

The Hartrell-Fan-Rockwood combination renders [1d] obvious. EX1003, ¶¶149-150.

As described with respect to [1e], Hartrell teaches that “the system activity monitor component determines whether it received notification of a suspected malware infection.” EX1004, [0027]. “The system activity monitor component can use the report of the infection *as the base time (e.g., t0) from which to create the time-bounded snapshots of recorded activities.*” *Id.*, [0030]. In other words, the “base time (e.g., t0)” is a timestamp, and a POSITA would have recognized that the time of interest and timestamp are both a “point in time,” because they are the time at which the system activity monitor component received notification of a suspected malware infection. EX1003, ¶150.

(xiii) **Claim 15**

[15]

See [8] of Ground 1A.

(xiv) **Claim 16**

[16]

See [5] of Ground 1A.

B. [GROUND 1B] – CLAIMS 11 AND 12 WOULD HAVE BEEN OBVIOUS OVER HARTRELL, FAN, ROCKWOOD, AND POLYAKOV

1. Polyakov (EX1007)

Polyakov describes a RootKit detector designed to “identify[] a RootKit on a computer that is designed to conceal malware that is resident on the computer.” EX1007, 3:32-34. Like the behavior monitor module described by Fan, Polyakov

describes that “the RootKit detector leverage[s] services provided by kernel debugger facilities to automatically obtain data from specified data structures that are maintained by an operating system.” EX1007, 3:35-38; EX1003, ¶70.

With respect to analyzing the data obtained by the kernel debugger facilities, Polyakov describes that the RootKit detector includes an integrity checker 212 that employs “three primary techniques” to identify a RootKit on a computer. EX1007, 5:17-23; EX1003, ¶71. One of the techniques described by Polyakov for analyzing the returned data is the use of “heuristics” to analyze “‘suspicious modifications’ that are made in data structures maintained by the operating system 204 to determine whether the modifications were generated by a RootKit.” EX1007, 5:42-45; EX1003, ¶71.

2. The Combination of Hartrell, Fan, Rockwell, and Polyakov

As described in Section IV.A.3, a POSITA would have found it obvious to implement Hartrell’s system activity monitor component 102 using “functional kernel mode drivers,” as described by Fan. EX1003, ¶72. A POSITA would have understood that Fan’s functional kernel mode drivers would have returned data similar to the kernel debugger facilities described by Polyakov—i.e., various “data structures that are maintained by an operating system.” *Compare* EX1005, 6:33-41, 6:56-60 *to* EX1007, 5:42-65; EX1003, ¶72. Both Fan and Polyakov utilize the data that can be obtained in “kernel mode” in order to identify malware that may

otherwise be able to disguise itself to “user mode” detection services (e.g., rootkits). *Compare* EX1005, 4:45-53 to EX1007, 3:32-46; EX1003, ¶72.

As noted in Section IV.B.1, *supra*, Polyakov teaches “three primary techniques” to identify a RootKit on a computer based on the data returned by its kernel debugger facilities. EX1007, 5:17-23. A POSITA would have found it obvious to implement Hartrell’s malware analysis system 100 (e.g., as part of or in communication with the system activity monitor component 102, which uses “any of a variety of well-known monitoring techniques to monitor operating system and/or network activities”) to use the “three primary techniques” to identify a RootKit taught by Polyakov to analyze the records of the monitored activities and help provide notification of whether there is a suspected malware infection. EX1003, ¶73 (citing EX1004, [0027]; EX1007, 5:17-6:12). Polyakov teaches that these techniques are employed by its integrity checker 212, and a POSITA would have found it obvious to integrate at least this functionality of the integrity checker 212 as a component or subcomponent into or in communication with Hartrell’s malware analysis system 100. EX1003, ¶73. Indeed, Polyakov describes the integrity checker 212 as a “software component” executed by computer 200 (*see* EX1007, 3:47-58, 5:6-12), and a POSITA would have found it obvious that the functionality of integrity checker 212 could be integrated as a software component or subcomponent into or in communication with Hartrell’s malware analysis

system 100, which describes that “locally executing anti-malware software” informs malware analysis system 100 of “the detection of a malware infection or suspected infection” (*see* EX1004, [0014]). EX1003, ¶73.

A POSITA would have been motivated to make this combination, because Hartrell does not specifically describe or otherwise limit how, in step 206 of FIG. 2, the system activity monitor component determines or receives notification that there is a suspected malware infection. EX1003, ¶74 (citing EX1004, [0027]). Rather, Hartrell leaves these details to a POSITA. *Id.* Polyakov provides at least three well-known techniques that a POSITA could utilize in implementing Hartrell’s malware analysis system 100 to provide the notification that there is a suspected malware infection. *Id.* Indeed, Polyakov specifically describes with respect to the second of these well-known techniques that “[t]hose skilled in the art and others will recognize that heuristic techniques that search for behaviors characteristic of malware *are particularly effective* in identifying previously unknown RootKit.” EX1007, 5:62-65. Thus, under Polyakov’s own explicit teachings, a POSITA would have recognized the effectiveness of employing the techniques it describes—and particularly the use of heuristics to identify malware—in detecting possible infections. EX1003, ¶74.

3. Element-by-Element Explanation of How Hartrell-Fan-Rockwood-Polyakov Combination Renders Challenged Claims Obvious

(i) Claim 11

[11] The system of claim 10, including: a heuristics module configured to be executed by the hardware processor and to evaluate whether the event is a suspected pestware-related event before the research portion accesses information in the at least one log that relates the event to the suspected pestware object.

The Hartrell-Fan-Rockwell-Polyakov combination renders [11] obvious.

EX1003, ¶¶155-157.

As described in Sections IV.B.1-2, *supra*, a POSITA would have found it obvious to implement Hartrell’s malware analysis system 100 to use the “three primary techniques” to identify a RootKit taught by Polyakov to analyze the records of the monitored activities and help provide notification of whether there is a suspected malware infection. EX1003, ¶156 (citing EX1004, [0027]; EX1007, 5:17-6:12). As further described in Sections IV.B.1-2, *supra*, a POSITA would have found it obvious that the functionality of Polyakov’s integrity checker 212 could be integrated as a software component or subcomponent into or in communication with Hartrell’s malware analysis system 100, which describes that “locally executing anti-malware software” informs malware analysis system 100 of “the detection of a malware infection or suspected infection” (*see* EX1004, [0014]). EX1003, ¶156. A POSITA would have understood or at least found it obvious that, in the proposed combination, a software component employing the

“heuristic techniques” described by Polyakov would be a “heuristics module configured to be executed by the hardware processor,” as recited in claim 11.

EX1003, ¶156. Polyakov describes that the software components of its system are executed by computer 200 (*see* EX1007, 3:46-58), and that the computer 200 includes a Central Processing Unit (CPU) that executes the processes on computer 200 (*see* EX1007, 7:30-37), which a POSITA would have understood to include the software components of its RootKit detector. EX1003, ¶156.

Further, Hartrell describes that the “notification of a suspected malware infection” is received by the system activity monitor component in step 206 of FIG. 2, which is before the system activity monitor component accesses the records of monitored activities to create the pre-infection and post-infection snapshots in steps 208 and 210 of FIG. 2. *See* EX1004, [0027]; EX1003, ¶157. Thus, a POSITA would have understood or at least found it obvious that the integrated software components/sub-components implementing Polyakov’s “heuristic techniques” would have “evaluate whether the event is a suspected pestware-related event before the research portion accesses information in the at least one log that relates the event to the suspected pestware object,” as recited in claim 11. EX1003, ¶157.

(ii) **Claim 12**

[12] The system of claim 11, wherein the heuristics module determines that the at least one event is suspected to be pestware-related by analyzing other

events in connection with the event, wherein the other events are also related to the time of interest.

The Hartrell-Fan-Rockwell-Polyakov combination renders [11] obvious.

EX1003, ¶¶158-160.

With respect to its “heuristic techniques,” Polyakov describes that “heuristics are employed to analyze ‘suspicious modifications’ that are made in data structures maintained by the operating system 204 to determine whether the modifications were generated by a RootKit.” EX1007, 5:42-45. Polyakov goes on to describe that “[t]hose skilled in the art and others will recognize that heuristic techniques that search for behaviors characteristic of malware are particularly effective in identifying previously unknown RootKit.” EX1007, 5:62-65. Thus, Polyakov’s “heuristic techniques” are meant to identify and analyze a number of different “behaviors,” not any one behavior. EX1003, ¶159. A POSITA would have understood or at least found it obvious that these combinations of multiple behaviors would help distinguish between a normal program that may have “legitimate reasons” for performing any one behavior and malicious programs that use a set of multiple known behaviors to conceal malware. *See* EX1007, 5:42-65; EX1003, ¶159. Indeed, this is consistent with Hartrell’s disclosure that the system activity monitor component monitors multiple “predetermined activities on or about the computer system.” *See* EX1004, [0027]; EX1003, ¶159.

Based on these teachings a POSITA would have at least found it obvious that the software component/sub-component in the proposed combination that utilizes Polyakov's heuristic techniques (i.e., the claimed heuristics module) would "determine[] that the at least one event is suspected to be pestware-related by analyzing other events in connection with the event, wherein the other events are also related to the time of interest." EX1003, ¶160. A POSITA would have understood or at least found it obvious from the teachings of Polyakov that known heuristic techniques often identified suspected malware based on a series of successive or parallel behaviors that together differentiate the behavior of malware (e.g., a rootkit) from legitimate programs. EX1003, ¶160 (citing EX1008, 6:65-7:35 (describing that any one activity may be suspicious but not dispositive of malware, but that a collection of suspicious activities provides a "strong heuristic indicator" of malware)).

V. PTAB DISCRETION SHOULD NOT PRECLUDE INSTITUTION

A. The *Fintiv* Factors Favor Institution—§314(a)

Under the Director's interim guidance, *Fintiv* Factors favor institution. EX1046 ("*Director's Memo*"); *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11, 6 (PTAB Mar. 20, 2020) (precedential) ("*Fintiv I*").

1. Factor 1

Parallel litigation is ongoing. While no stay has been requested, Petitioner may seek a stay if institution is granted, thus favoring institution. At worst, this

factor is neutral because the Board “will not attempt to predict” how the district court will proceed. *Sand Revolution II, LLC v. Continental Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 7 (PTAB Jun. 16, 2020).

2. Factor 2

“This factor looks at the proximity of the trial date to the date of [FWD] to assess the weight to be accorded a trial date set earlier than the expected [FWD] date.” *Shenzhen Carku Tech. Co., Ltd. v. The Noco Co.*, IPR2020-00944, Paper 20 at 61 (PTAB Nov. 12, 2020). Understanding “scheduled trial dates are unreliable and often change,” the *Director’s Memo* dictates “[a] court’s scheduled trial date...is not by itself a good indicator of whether the district court trial will occur before the statutory deadline for a [FWD].” *Samsung Elec.’s Co., Ltd. et al. v. Staton Techiya, LLC*, IPR2022-00302, Paper 13, at 10 (PTAB July 11, 2022). When analyzing this factor, “the parties [may] rely on time-to-trial statistics” along with “additional supporting factors such as the number of cases before the judge in the parallel litigation and the speed and availability of other case dispositions.” *Id.*, 10-12.

Based on the most recent WDTX statistics, the “actual trial date” is likely to be held mid-January, 2025—28.2 months from September 8, 2022, the date the first Amended Complaint was filed asserting the ’244 Patent against CrowdStrike. *See* EX1010, 37. While the District Court entered a schedule with an August 19,

2024 trial date assigned to each defendant in five separate lawsuits that have been consolidated for pretrial issues, PO has admitted that “median time-to-trial statistics are more appropriate” for consideration under this factor. IPR2023-00289, Paper 6, at 13; EX1012, 8. Either way, because the court’s consolidation is limited to pretrial issues, separate trials will be required, and trial cannot proceed on August 19, 2024 for all defendants.

The Board’s FWD deadline in this case is around December, 2024, a month before a trial would occur based on the time-to-trial statistics this Board should apply to its analysis. However, “additional supporting factors” are at play weighing towards a longer than average filing-to-trial window. First, the District Court litigation involves thirty-nine total patents, including twelve asserted by PO against Petitioner and two counterclaim infringement allegations brought by Petitioner. As PO previously quoted, U.S. Magistrate Judge Gilliland explained in a December 9, 2022 hearing that this is “a phenomenal case that I’m hoping one day maybe people will write songs about because it is about as complicated as you’re going to get with patent cases.” IPR2023-00289, Paper 6, at 2.

To accommodate the significant number of allegations, the Court scheduled three Markman hearings, the first was conducted on March 7, 2023, the second was recently held on May 3, 2023 relating to the Counterclaim patents, and the final Markman is scheduled for July 25, 2023, which will involve the Challenged

Patent. EX1013, 3, 5, 6. Accordingly, with multiple Markman hearings and a multitude of claims involved in a phenomenally complex case, the “actual trial date” is much more likely to be delayed until later, commensurate with the median time to trial (or later) and after FWD here.

Moreover, the '244 Patent was added to the litigation in September, 2022, as one of six additional asserted patents, introduced over six months after the original six patents were asserted against Petitioner. The '244 Patent, and other late asserted patents, were shoehorned into an already ongoing litigation, resulting in a shortened window to trial. Petitioner acted diligently in filing its Petition against this newly asserted patent, and any shortened window to trial for these patents is the result of PO's own litigation strategy.

In sum, the Board's FWD deadline in this case falls before a trial would occur based on WDTX time-to-trial statistics. But the additional complicating factors at issue here suggest trial will occur even later in time than the statistics suggest, weighing heavily in favor of institution.

3. Factor 3

This factor favors institution because Petitioner worked diligently to overcome the challenging circumstances of the Litigation and file this Petition three months ahead of the statutory time bar. *Fintiv I*, 11. Beyond claim construction, the parties have invested little in the parallel proceeding. The first

and second Markman hearings were just recently conducted on March 7, 2023 and May 3, 2023, respectively. The final Markman is not scheduled until July 25, 2023. Fact discovery just opened on April 18, 2023 and is not set to conclude until January 18, 2024. Expert discovery does not close until March 12, 2024 and dispositive motions are not due until May 7, 2024. This factor weighs in favor of institution.

4. Factor 4

Petitioner asks the Board to consider the unique challenges raised in the Petition. *See Fintiv*, 12-13. Petitioner stipulates that, if the Board institutes *inter partes* review, Petitioner will not seek resolution in the District Court of any ground of invalidity for the '244 Patent that utilizes Hartrell, Fan, Rockwood, and Polyakov—the prior art references relied upon in the grounds of the instant petition. Petitioner's proposal is comprehensive and addresses the concern of narrowing the issues, as this stipulation ensures there is no overlap between the '244 Patent invalidity theories before the Board and at issue in the District Court. Petitioner's proposed stipulation is far more restrictive than a *Sand Revolution* stipulation, which excludes only the same grounds advanced in the IPR, allowing the Petitioner to rely on the same references in the two tribunals so long as the reliance is not identical. Accordingly, this factor weighs heavily in favor of institution.

5. Factor 5

Petitioner and Patent Owner are parties to the Litigation. However, members of the Board have noted *Fintiv* addresses only the scenario in which the petitioner is unrelated to a defendant in a parallel proceeding, finding this should weigh against denying institution, but that *Fintiv* “says nothing about situations in which the petitioner is the same as, or is related to, the district court defendant.” *Cisco Sys., Inc. v. Ramot at Tel Aviv Univ. Ltd.*, IPR2020-00122, Paper 15 at 10-11 (PTAB May 15, 2020) (APJ Crumbley, dissenting).

6. Factor 6: The Merits Support Institution

“[W]here the PTAB determines that the information presented at the institution stage presents a compelling unpatentability challenge, that determination alone demonstrates that the PTAB should not discretionarily deny institution under *Fintiv*.” *Hewlett Packard Enterprise Co., et al. v. Billjco LLC*, IPR2022-00420, Paper 17, at 6 (PTAB July 12, 2022). The strength of Petitioner’s proposed grounds presented above weighs strongly in favor of institution. *Id.* (refusing to deny institution under the *Fintiv* framework based solely on the strength of Petitioner’s grounds).

B. § 325(d) Favors Institution

Hartrell, Fan, Rockwood, and Polyakov were not considered or cited in the prosecution of the ’244 Patent, and consequently, are being considered for the first time in this IPR proceeding. Thus, none of the Grounds involve the same or

substantially the same prior art or arguments previously presented to the Office.

35 U.S.C. § 325(d).

VI. FEES

Petitioner authorizes the Patent and Trademark Office to charge Deposit Account No. 06-1050 for the fee set in 37 C.F.R. § 42.15(a) for this Petition.

VII. CONCLUSION

Petitioner requests cancellation of all Challenged Claims.

VIII. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(a)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

CrowdStrike, Inc. and CrowdStrike Holdings, Inc. are the real parties-in-interest.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

Petitioner is not aware of any disclaimers, reexamination certificates or petitions for *inter partes* review for the '244 Patent. The '244 Patent is the subject of the following civil actions:

- *Webroot, Inc. et al. v. CrowdStrike, Inc. et al.*, Case No. 6:22-cv-00241-ADA (WDTX)
- *Webroot, Inc. et al v. AO Kaspersky Lab*, Case No. 6-22-cv-00243 (WDTX)
- *Webroot, Inc. et al v. Sophos Limited*, Case No. 6-22-cv-00240 (WDTX)

- *Webroot, Inc. et al v. Trend Micro, Inc.*, Case No. 6-22-cv-00239

(WDTX)

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Petitioner provides the following designation of counsel.

Lead Counsel	Backup Counsel
W. Karl Renner, Reg. No. 41,265 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR54971-0005IP1@fr.com	David Holt, Reg. No. 65,161 Fish & Richardson P.C. 60 South Sixth Street, Suite 3200 Minneapolis, MN 55402 Tel: 202-783-5070 Fax: 877-769-7945 Email: IPR54971-0005IP1@fr.com

D. Service Information

Please address all correspondence and service to the address listed above.

Petitioner consents to electronic service by email at IPR54971-0005IP1@fr.com.

Respectfully submitted,

Dated: June 30, 2023

/W. Karl Renner/

W. Karl Renner, Reg. No. 41,265
David Holt, Reg. No. 65,161
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
T: 202-783-5070
F: 877-769-7945

CERTIFICATION UNDER 37 CFR § 42.24

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,736 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated: June 30, 2023

/W. Karl Renner/

W. Karl Renner, Reg. No. 41,265

David Holt, Reg. No. 65,161

Fish & Richardson P.C.

60 South Sixth Street, Suite 3200

Minneapolis, MN 55402

T: 202-783-5070

F: 877-769-7945

Counsel for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on June 30, 2023, a complete and entire copy of this Petition for *Inter Partes* Review, Power of Attorney, and all supporting exhibits were provided, via Federal Express, to the Patent Owner by serving the correspondence address of record as follows:

Sheridan Ross PC
1560 Broadway
Suite 1200
Denver, CO 80202
UNITED STATES

/Michael Stanwyck/
Michael Stanwyck
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
(202) 626-7790