

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner

v.

TAASERA LICENSING LLC,
Patent Owner

Inter Partes Review Case No. IPR2023-01465
U.S. Patent No. 8,850,517

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,850,517**

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	GROUND FOR STANDING (37 C.F.R. § 42.104(A)).....	1
III.	IDENTIFICATION OF CHALLENGE (37 C.F.R. § 42.104(B))	1
	A. CITATION OF PRIOR ART	1
	B. STATUTORY GROUNDS FOR CHALLENGE.....	2
IV.	THE '517 PATENT.....	2
	A. PRIORITY DATE OF THE '517 PATENT	2
	B. SPECIFICATION OF THE '517 PATENT	2
	C. SUMMARY OF THE PROSECUTION HISTORY	5
	D. LEVEL OF ORDINARY SKILL IN THE ART	6
	E. CLAIM CONSTRUCTION	7
V.	HOW THE CLAIMS ARE UNPATENTABLE.....	7
	A. OVERVIEW OF ASSERTED PRIOR ART	8
	1. WRIGHT.....	8
	2. DASWANI.....	10
	3. MOTIVATION TO COMBINE WRIGHT AND DASWANI.....	11
	B. GROUND 1: CLAIMS 1-24 ARE UNPATENTABLE UNDER 35 U.S.C. § 103(A) AS OBVIOUS IN VIEW OF WRIGHT AND DASWANI	16
	1. CLAIM 1	16
	a) 1[a] A method for assessing runtime risk for an application program that executes on a device, comprising:.....	16
	b) 1[b] storing, in a rules database, a plurality of rules, wherein each rule identifies an action sequence;	17
	c) 1[c] storing, in a policy database, a plurality of assessment policies, wherein each assessment policy includes at least one rule of the plurality of rules;	25
	d) 1[d] identifying, using at least one assessment policy, a runtime risk for an application program that executes on a device, wherein the identified runtime risk indicates a risk or threat of the identified action sequence of the application; and	27

e)	1[e] identifying, by a runtime monitor including a processing device, a behavior score for the application program that executes on the device based on the identified runtime risk, wherein	31
f)	1[f] wherein the action sequence is a sequence of at least two performed actions, and each performed action is at least one of: a user action, an application action, and a system action.	36
2.	CLAIM 2	39
a)	The method of claim 1, wherein the user action is any form of explicit input from a user of a computing system.	39
3.	CLAIM 3	40
a)	The method of claim 1, wherein the application action is any activity performed by the application initiated programmatically by a task in execution of a computing system.	40
4.	CLAIM 4	43
a)	The method of claim 1, wherein the system action is any operation performed by a computing system on behalf of, or as a consequence of, a user action or application action that changes the state of the computing system.	43
5.	CLAIM 5	44
a)	The method of claim 1, wherein a first action of the at least two performed actions is performed at a first time, and the second action of the at least two performed actions is performed at a second time.	44
6.	CLAIM 6	46
a)	The method of claim 5, wherein the first action is a user action, the second action is a system action, and the second time occurs after the first time.	46
7.	CLAIM 7	47
a)	The method of claim 5, wherein the first action is a user action, the second action is an application action, and the second time occurs after the first time.	47
8.	CLAIM 8	49
a)	The method of claim 5, wherein the first action is an application action, the second action is a system	

	action, and the second time occurs after the first time.....	49
9.	CLAIM 9.....	50
	a) The method of claim 1, wherein the action sequence includes three performed actions, a first action of the three performed actions is a user action, a second action of the three performed actions is an application action, a third action of the three performed actions is a system action, the second action is performed temporally after the first action, and the third action is performed temporally after the second action.....	50
10.	CLAIM 10.....	51
	a) The method of claim 1, wherein the runtime risk is identified based on matching rules and probabilistic weights and scores of performed actions on the device.	51
11.	CLAIM 11.....	56
	a) The method of claim 1, wherein the runtime risk is identified based on time proximity of the at least two performed actions of the action sequence and natural affinity inferred between the at least two performed actions.	56
12.	CLAIM 12.....	57
	a) 12[a] The method of claim 1, further comprising: assessing, by the processing device, that the at least two performed actions of the action sequence describe a forensic chain of events based on a plurality of factors, ...	57
	b) 12[b] wherein the plurality of factors includes at least one of: ordered sequences, unordered sequences, proximity analysis, outliers, probabilistic weights and scores, and probable action inferences based on an application's competency.....	58
13.	CLAIM 13.....	59
	a) 13[a] A system for assessing runtime risk for an application program that executes on a device, comprising:	59

b)	13[b] a rules database storing a plurality of rules, wherein each rule identifies an action sequence;	60
c)	13[c] a policy database storing a plurality of assessment policies, wherein each assessment policy includes at least one rule of the plurality of rules; and	60
d)	13[d] a runtime monitor including a processing device	60
e)	13[e] identifying, using at least one assessment policy, a runtime risk for an application program that executes on a device, wherein the identified runtime risk indicates a risk or threat of the identified action sequence of the application, and	61
f)	13[f] identifying a behavior score for the application program that executes on the device based on the identified runtime risk, wherein, ...	61
g)	13[g] the action sequence is a sequence of at least two performed actions, and	61
h)	13[h] each performed action is at least one of: a user action, an application action, and a system action.	61
14.	CLAIM 14.....	62
a)	The system of claim 13, wherein the user action is any form of explicit input from a user of a computing system.....	62
15.	CLAIM 15.....	62
a)	The system of claim 13, wherein the application action is any activity performed by the application initiated programmatically by a task in execution of a computing system.....	62
16.	CLAIM 16.....	63
a)	The system of claim 13, wherein the system action is any operation performed by a computing system on behalf of, or as a consequence of, a user action or application action that changes the state of the computing system.....	63
17.	CLAIM 17.....	63
a)	The system of claim 13, wherein a first action of the at least two performed actions is performed at a	

	first time, and the second action of the at least two performed actions is performed at a second time.....	63
18.	CLAIM 18.....	64
	a) The system of claim 17, wherein the first action is a user action, the second action is a system action, and the second time occurs after the first time.....	64
19.	CLAIM 19.....	64
	a) The system of claim 17, wherein the first action is a user action, the second action is an application action, and the second time occurs after the first time.....	64
20.	CLAIM 20.....	64
	a) The system of claim 17, wherein the first action is an application action, the second action is a system action, and the second time occurs after the first time.....	64
21.	CLAIM 21.....	65
	a) The system of claim 17, wherein the action sequence includes three performed actions, a first action of the three performed actions is a user action, a second action of the three performed actions is an application action, a third action of the three performed actions is a system action, the second action is performed temporally after the first action, and the third action is performed temporally after the second action.....	65
22.	CLAIM 22.....	65
	a) The system of claim 13, wherein the runtime risk is identified based on matching rules and probabilistic weights and scores of performed actions on the device.	65
23.	CLAIM 23.....	66
	a) The system of claim 13, wherein the runtime risk is identified based on time proximity of the at least two performed actions of the action sequence and natural affinity inferred between the at least two performed actions.	66
24.	CLAIM 24.....	66
	a) 24[a] The system of claim 13, wherein the processing device assesses that the at least two	

	performed actions of the action sequence describe a forensic chain of events based on a plurality of factors, and	66
b)	24[b] the plurality of factors includes at least one of: ordered sequences, unordered sequences, proximity analysis, outliers, probabilistic weights and scores, and probable action inferences based on an application's competency.....	67
VI.	NO BASIS FOR DISCRETIONARY DENIAL	67
VII.	CONCLUSION.....	70
VIII.	MANDATORY NOTICES (37 C.F.R. § 42.8(A)(1)).....	70
A.	REAL PARTY IN INTEREST:	70
B.	RELATED MATTERS:.....	70
C.	LEAD AND BACKUP COUNSEL:	71
D.	SERVICE INFORMATION:	72
E.	PAYMENT OF FEES	72

TABLE OF AUTHORITIES

Cases

<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH</i> , IPR2019-01469, Paper 6 (Feb. 13, 2020)	69
<i>Cellco Partnership v. Huawei Device Co.</i> , IPR2020- 01117, Paper 10 (Feb. 3, 2021).....	70
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966)	9
<i>KSR Int’l Co. v. Teleflex, Inc.</i> , 550 U.S. 398 (2007)	9
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (<i>en banc</i>).....	8
<i>Titanium Metals Corp. of Am. v. Banner</i> , 778 F.2d 775 (Fed. Cir. 1985)...	39, 60, 68

Statutes

35 U.S.C. § 102(a)/(b) (pre AIA)	2
35 U.S.C. § 102(e) (pre AIA)	2
35 U.S.C. § 103(a) (pre AIA)	passim
35 U.S.C. § 325(d).....	68

Regulations

37 C.F.R. § 42.100(b)	8
37 C.F.R. § 42.104(a)	1
37 C.F.R. § 42.104(b)	2
37 C.F.R. § 42.105.....	75
37 C.F.R. § 42.24.....	74
37 C.F.R. § 42.6(e)	75
37 C.F.R. § 42.8.....	74
37 C.F.R. § 42.8(a)(1)	70
37 C.F.R. § 42.8(b)(2)	70
37 C.F.R. § 42.8(b)(3)	71

APPENDIX OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 8,850,517 to Kumar, et al.
1002	Prosecution History of 8,850,517.
1003	Declaration of Seth James Nielson, Ph.D. under 37 C.F.R. § 1.68.
1004	Curriculum Vitae of Seth James Nielson, Ph.D.
1005	U.S. Published Patent Application No. 2011/0023115 to Wright.
1006	U.S. Patent No. 8,806,647 to Daswani, et al.
1007	Andrew S. Tanenbaum, <i>Modern Operating Systems 3</i> , Pearson Prentice Hall, (2008).
1008	Brian Clifton, <i>Advanced Web Metrics with Google Analytics 3</i> , John Wiley & Sons, (2012).
1009	U.S. Provisional Application No. 61/479,841 to Daswani et al.
1010	United States Judicial Panel on Multidistrict Litigation Transfer Order
1011	In re: Taasera Licensing LLC, Patent Litigation, No. 2:22-MD-03042-JRG Eighth Amended Docket Control Order
1012	In re: Taasera Licensing LLC, Patent Litigation, No. 2:22-MD-03042-JRG Dkt. 189 Order on Trend's Motion to Transfer

I. INTRODUCTION

CrowdStrike, Inc. (“Petitioner”) petitions for *inter partes* review of claims 1-24 of U.S. Patent No. 8,850,517 (“the ’517 patent,” EX1001).

II. GROUNDS FOR STANDING (37 C.F.R. § 42.104(A))

Petitioner certifies the ’517 patent is available for *inter partes* review and Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

III. IDENTIFICATION OF CHALLENGE (37 C.F.R. § 42.104(B))

A. Citation of Prior Art

U.S. Patent Publication No. 2011/0023115 A1 to Wright (“Wright”, EX1005) is prior art under pre-AIA 35 U.S.C. § 102(a)/(b) because it published on January 27, 2011, more than 1 year before the priority date of the ’517 patent. Wright was not cited or considered during prosecution of the ’517 patent.

U.S. Patent No. 8,806,647 to Daswani et al. (“Daswani”, EX1006) is prior art under pre-AIA 35 U.S.C. § 102(e) because the application leading to the patent (App. No.: 13/458,982) was filed April 27, 2012, and claimed priority to U.S. Provisional Application 61/478,841 (EX1009) that was filed April 27, 2011, more than a year before the ’517 patent priority date. Daswani was not cited or considered during prosecution of the ’517 patent. There is no evidence the ’517 patent is entitled to a priority date earlier than January 15, 2013. If Patent Owner attempts to establish a

priority date earlier than April 27, 2012, Petitioner is prepared to show the '841 provisional application meets both the *Dynamic Drinkware* and *In re Giacomini* requirements.

B. Statutory Grounds for Challenge

GROUND 1: Claims 1-24 are unpatentable under pre-AIA 35 U.S.C. § 103(a) as obvious in view of Wright (EX1005) and Daswani (EX1006).

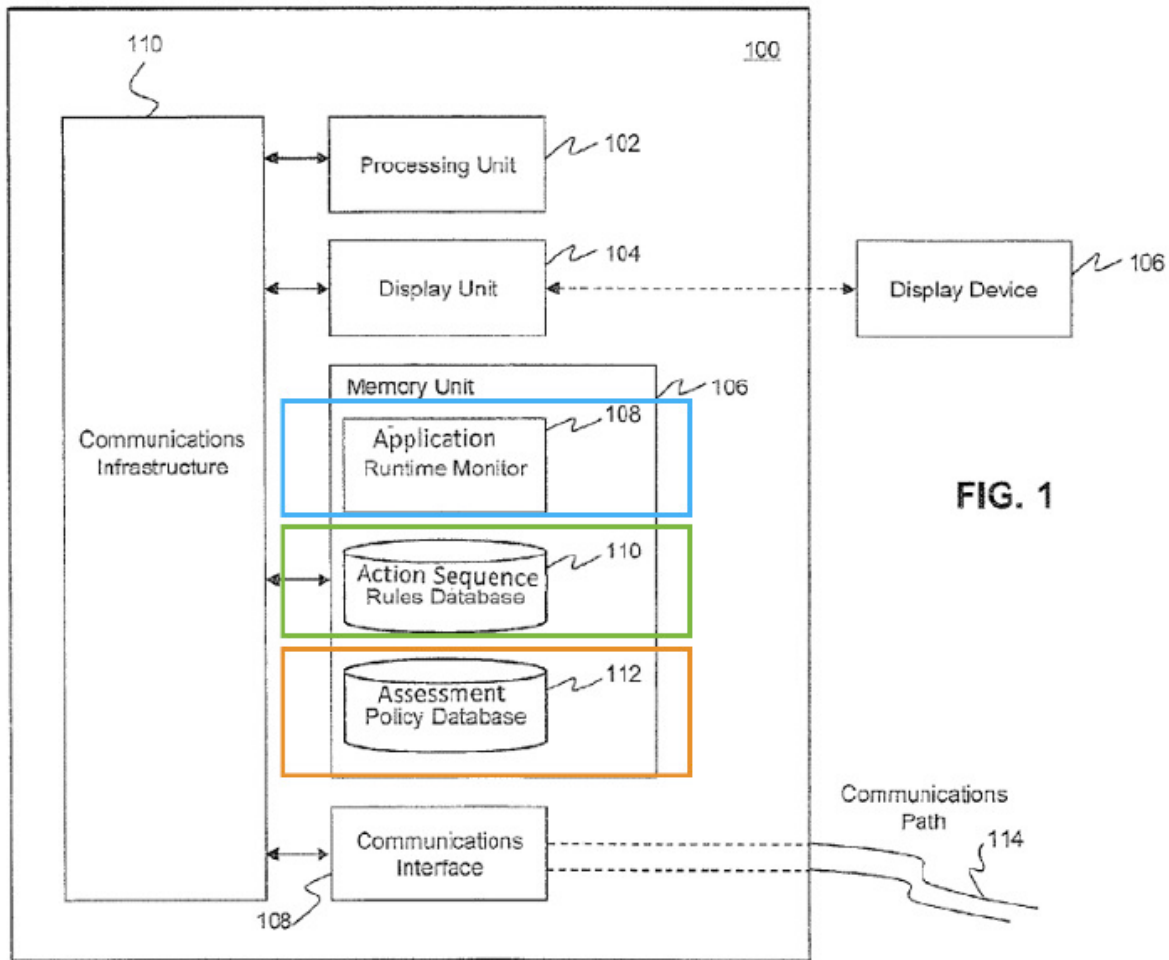
IV. THE '517 PATENT

A. Priority Date of the '517 patent

The '517 patent issued September 30, 2014 from U.S. Patent Application No. 13/741,878, which was filed January 15, 2013. *See* EX1001, (21) (22), (45). Thus, the claims should be evaluated under pre-AIA statutes.

B. Specification of the '517 patent

The '517 patent is directed to detecting risks to computer security, including advanced persistent threats, using behavioral analysis techniques. *See* EX1001, 1:7-39; EX1003, ¶80. The '517 patent discloses a computing system and related method “configured to assess the runtime risk of an application or device and identify a behavior score based on the assessed runtime risk.” EX1001, 2:40-54; EX1003, ¶81. The specification explains that “a runtime risk may identify a physical damage threat, accidental threat, deliberate threat, a threat of technical failure, a compromise of function or information, a combination thereof, etc.” EX1001, 6:9-12.



EX1001, FIG. 1 (annotated); EX1003, ¶83.

In the methods, a runtime monitor 108 identifies and considers action sequences. EX1001, 7:27-8:21. The specification explains that an action sequence “may include a sequence of at least two performed actions,” where each action may be one of three types: a user action, an application action, and a system action. EX1001, 4:14-17. The patent explains:

The user action 302 may be any form of explicit input from a user of the computing system 100. The application action 304 may be any

activity performed by the application initiated programmatically by a task in execution of the computing system 100. The system action 306 may be any operation performed by the computing system 100 on behalf or, or as a consequence of, a user action 302 or application action 304 that changes the state of the computing system 100.

Id., 9:17-24. “For example, the user action may include a program launch, keyboard input, click of a mouse, touch input on a touch screen, moving of a mouse, scrolling or clicking on a click wheel, voice command into a microphone, gestures into a device configured to read human gestures, etc.” *Id.*, 6:54-62. An action sequence “may indicate that there is a forensic chain of events or a natural progression between the performed actions.” *Id.*, 7:59-61.

A rules database 110 stores a plurality of rules, each rule identifying an action sequence. *Id.*, 9:67, 10:13-14. The rules are included in “assessment policies,” which are stored in a policy database 112. *Id.*, at 9:66-10:3; EX1003, ¶82. “Each assessment policy may comprise one or more rules stored in the rules database 110 and may identify the runtime risk based on matching rules and probabilistic weights and scores.” EX1001, 6:16-22. Using assessment policies, the ’517 patent discloses “identifying, using at least one assessment policy, a runtime risk for an application program that executes on a device, wherein the identified runtime risk indicates a risk or threat of the identified action sequence of the application[.]” *Id.*, 10:4-8; EX1003, ¶82.

In addition, the runtime monitor 108 produces a “behavior score for the application” based on the identified risk. EX1001, 10:9-12. The specification explains that “[t]he behavior score may be a number or other value (e.g., a rating) that may indicate the risk or threat of the application or device to the computing system 100 or other device (e.g., a network connected to the computing system 100 via the communications interface 108, etc.).” *Id.*, 6:35-45.

Finally, the patent discloses various ways these assessments may be made, such as action sequences which “describe a forensic chain of events” that may include “ordered sequences, unordered sequences, proximity analysis, outliers, probabilistic weights and scores, and probable action inferences based on an application’s competency.” *Id.* at 10:65-11:5; EX1003, ¶82.

C. Summary of the Prosecution History

A May 7, 2013 non-final Office Action rejected all claims as either anticipated by or unpatentable based on U.S. Pub. No. 2013/0096980 to Basavapatna.¹ EX1002, pp.164-170. In response, Applicant stated: “The claims of the present application are directed towards a method and system for assessing the runtime risk of an application program that executes on a device. To achieve this assessment, rules that identify *action sequences* of actions performed by users, applications, or systems are

¹ The Office Actions cite “Basavapatna-McAfee,” presumably because the assignee is McAfee LLC.

used in assessment policies, which *identify and predict* specific types of threats as part of the risk assessment.” *Id.* at p.147 (emphases original). Applicant further argued: “[N]ot only does Basavapatna fail to disclose or suggest the rules identifying action sequences, Basavapatna also fails to disclose or suggest action sequences generally. Basavapatna does not disclose any *sequence of performed actions*, let alone a sequence of performed actions where the performed actions are at least one of a user action, an application action, and a system action, as recited in the claims.” *Id.* at p.148 (emphases original).

Taasera responded to an August 15, 2013, Final Office Action reasserting all rejections by amending claims 1 and 13 to recite “wherein the identified runtime risk indicates a risk or threat of the identified action sequence of the application”, and “a runtime monitor including a processing device.” *Id.* at p.93-99.

In an Applicant initiated Examiner interview conducted November 8, 2013, Taasera argued that the amendments distinguished over Basavapatna. *Id.* at p.89. Thereafter, a Notice of Allowance issued February 12, 2014, with no Examiner’s reasons for allowance. *Id.* at pp.60-64.

D. Level of Ordinary Skill in the Art

As established by the Declaration of Dr. Seth Nielson dated April 4, 2023, a person having ordinary skill in the art (PHOSITA) as of for the ’517 patent priority date would have had a bachelor’s degree in computer science, or a related field, or

an equivalent technical degree or equivalent work experience, and an additional one to two years of education or experience in the field of computer security. More education can supplement practical experience and vice versa. EX1003, ¶87.

E. Claim Construction

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

For purposes of this proceeding only, Petitioner proposes that claim terms should be afforded their plain and ordinary meanings and no specific claim constructions are necessary to apply the asserted prior art to the '517 claims.² See *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011) (“[C]laim terms need only be construed to the extent necessary to resolve the controversy.”).

V. HOW THE CLAIMS ARE UNPATENTABLE

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

Petitioner evaluates the scope and content of the prior art, any differences between the art and the claim, and the knowledge of person of ordinary skill in the

² Petitioner reserves the right in other proceedings to construe the '517 claims and pursue indefiniteness theories.

art in accordance with *Graham v. John Deere Co.*, 383 U.S. 1 (1966) and *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398 (2007). Petitioner is unaware of any secondary considerations relevant to the ’517 patent and a more than adequate written rationale is provided to support the combination of Wright and Daswani under *KSR* 550 U.S. at 418.

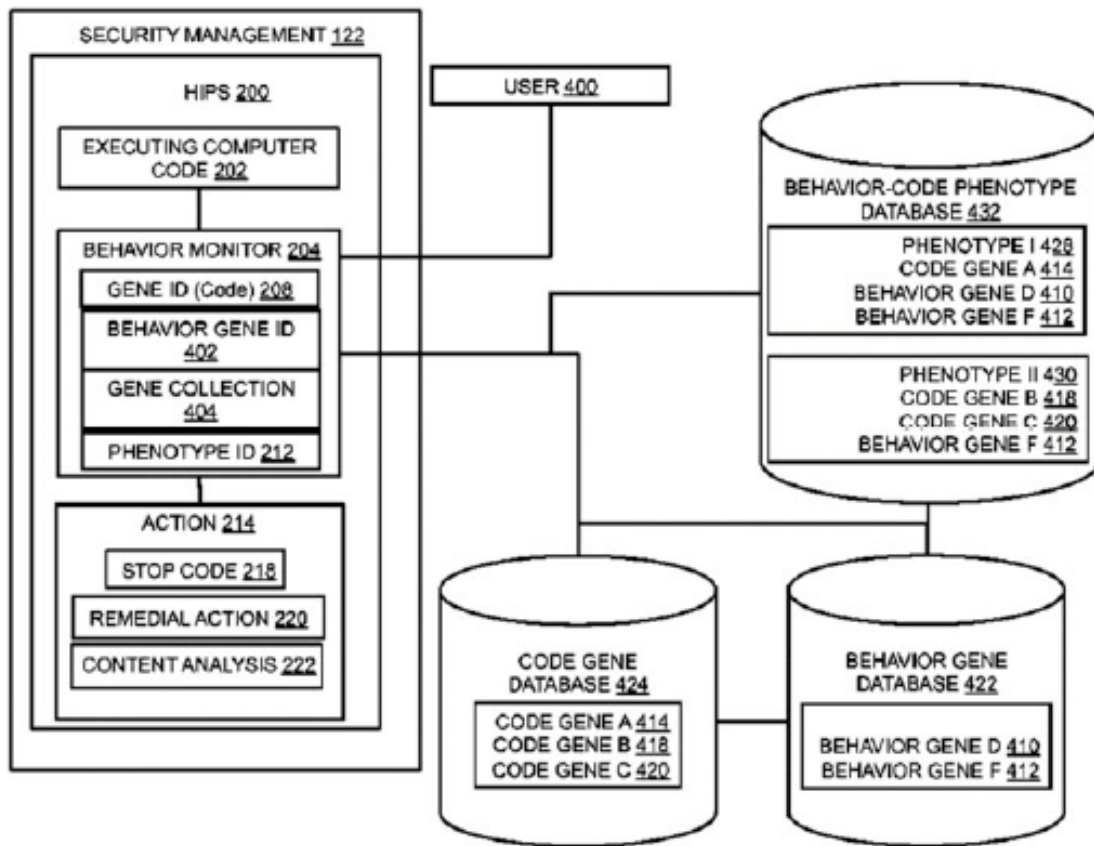
A. Overview of Asserted Prior Art

Petitioner submits that claims 1-24 of the ’517 patent are invalid as unpatentable under 35 U.S.C. § 103(a) in view of the combination of Wright and Daswani.

1. Wright

Wright is directed to methods, systems and computer program products that monitor a user’s interaction with a computer to detect whether the user is engaged in a type of malicious usage session with the computer. EX1005, claim 1. While Wright is in the field of computer technology, it uses terminology from the field of life sciences to describe some of the relevant concepts. EX1003, ¶90. Wright discloses monitoring user interactions with a computer, comparing user interactions with predetermined behaviors, referred to as “behavioral genes,” and monitoring a computer code process for indications of a code operation, referred to as a “code gene.” EX1005, [0007]-[0008]. For example, a behavioral gene called “Internet download”—which summarizes the user behaviors of 1) executing a hyperlink on

the Internet, and 2) performing a “save” operation—may be created and stored in association with the computer network. *Id.*, [0007]. A code gene, for example, may be a virus definition, a malware definition, an adware definition, or some other code behavior. *Id.*, [0008]. A plurality of code operation indications are collected and compared to a predetermined collection of user behavior-code operation indications, referred to as a “phenotype,” which is typically present in a type of malicious usage session. *Id.* If this phenotype is detected, additional monitoring is performed to obtain a second set of user behaviors and code operations, which are then compared to a second phenotype associated with a known family of malware. *Id.* A protective action is taken if malicious behavior is detected. *Id.*, [0008]-[0009] A system of Wright is illustrated in FIG. 4:



EX1005, Fig. 4; EX1003, ¶91

2. Daswani

Daswani is directed to computer-implemented methods and systems for analyzing mobile applications for malware. Daswani discloses receiving a mobile application from an application server and loading it into an emulated environment that simulates a mobile device. A static analysis is performed on the application to determine a set of simulated user inputs. A user interface exerciser is loaded, which is configured to simulate the user inputs, and the resulting behavior of the simulated

mobile device is observed. The behaviors are classified as hard signals or soft signals, and the soft signals are combined to calculate a probability of maliciousness. The mobile application is then classified as malicious or non-malicious based on the hard signals and the probability of maliciousness. Daswani further discloses loading a mobile application into an emulated environment that simulates a mobile device and determining a set of simulated user inputs. A user interface exerciser simulates the user inputs, and the resulting behavior of the simulated mobile device is observed. The probability of malware in the behaviors resulting from the simulation of the user inputs is assessed, and prepares a vulnerability report based on the assessed malware probabilities is prepared. EX1005, 11:9-45.

3. Motivation to Combine Wright and Daswani

A PHOSITA would have been motivated to combine the teachings of Wright and Daswani because both Wright and Daswani disclose using behavioral analysis to identify unknown malware and adversarial operations in computer networks. EX1003, ¶103, EX1005, 6:61-4, 9:48-52, and EX1006, 1:18-23, 2:29-36.

Specifically, Wright states:

The behavior monitoring facility 204 may be able to identify a phenotype 212 of user and code behaviors in relation to an executing code by comparing a collection of observed user and code behaviors with the predetermined collections of known malicious user and code behaviors stored as phenotypes in a behavior-code phenotype database 432, either temporarily or for a longer term. Behavior-code phenotypes

may capture a combination or a series of user and code behaviors that may be ranked to create increasing levels of confidence that the runtime object being monitored is executing a behavior pattern comparable to a known family of malware. The predetermined collections of behaviors, or phenotypes, may be created based on suspicious or risky user behavior (e.g. downloading a file from a website, running an executable, opening a signed file with an unverifiable digital signature, or other potentially suspicious or risky behavior) and scaled analysis of families of malware in runtime, such as in a threat research facility 132 of a threat management facility 100.

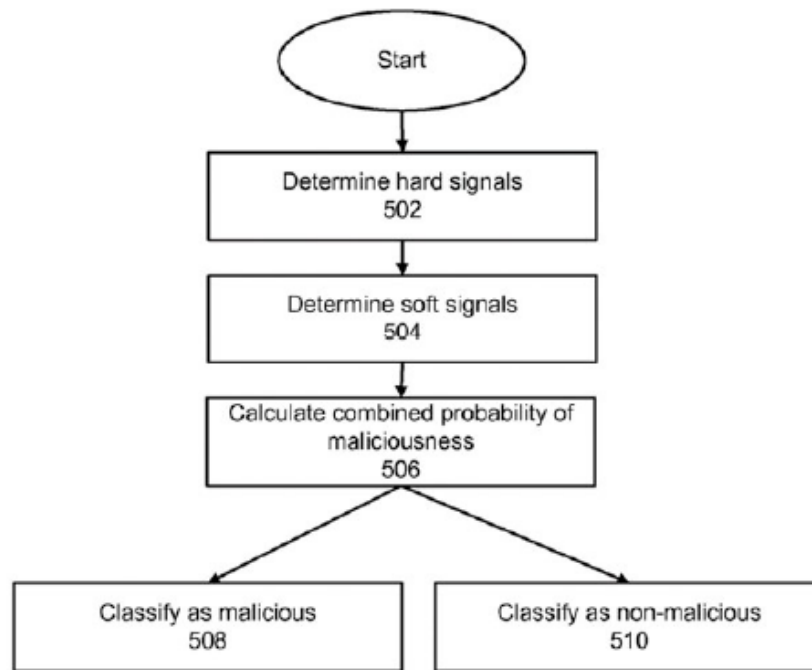
EX1005, [0089] (emphasis added)

Similar to Wright, Daswani uses ranking of phenotypes (comparisons or collections of user and code behaviors) based on confidence to identify unknown software as belonging to a certain family of malware. EX1003, ¶107. For example, Daswani discloses:

Signal combination module 306 combines multiple behaviors signals to determine an overall likelihood that an application is malicious. In one embodiment, some system calls or actions simulated in the emulated environment may be classified as soft signals or hard signals. In one embodiment, soft signals comprise system calls or actions which have an associated probability of the application containing malware. Hard signals comprise system calls or actions that are by definition malicious actions, or associated with a probability of the application containing malware, the probability exceeding a hard signal threshold.

EX1005, 6:45-55.

A PHOSITA would be motivated to combine Wright and Daswani because Daswani discloses how to create these confidence levels. Daswani discloses using hard signals (behaviors that are likely malicious) and soft signals (behaviors only malicious in certain contexts) as inputs to a probabilistic signal combination module that computes the likelihood that an application is malicious. EX1003, ¶¶107-109, EX1005, 6:45-47, and Fig. 5.



EX1005, Fig. 5; EX1003, ¶109

A PHOSITA would have recognized that Daswani's hard signals and soft signals are generally applied to behaviors because Daswani discloses that:

In one embodiment, soft signal and hard signal classifications are assigned to behaviors to indicate a likelihood that a certain behavior is

due to malware on the application being tested. Additionally, in one embodiment detected soft signals and hard signals can be combined to determine an overall likelihood of the tested application containing malware.

EX1005, 4:2-8; EX1003, ¶111.

Thus, a PHOSITA would recognize that the signals of Daswani correspond to the behavioral genes of Wright. EX1003, ¶111.

A PHOSITA would be motivated to use Daswani's approach in Wright's system to create confidence levels because Daswani's method maximizes both flexibility in classification of phenotypes and permits minimization of false positives (minimal experimentation while giving the PHOSITA high confidence of success). EX1003, ¶112. Daswani's method has multiple configurations of calculating hard and soft signals together to compute a final probability. *Id.* at ¶111. Additionally, Daswani discloses that the calculations can be adjusted according to false positives. Specifically, Daswani states:

FIG. 5 is a flow diagram illustrating an example method for combining hard signals and soft signals to determine a likelihood of malware according to one embodiment of the present disclosure. After the analysis module 204 observes behaviors and classifies hard and soft signals, the analysis module 204 proceeds to assess the likelihood of an application being malicious based on hard and soft signal infection rates and a false positive rate. Analysis module 204 determines 502 observed hard signal behaviors and determines a probability of maliciousness for

each of the hard signals. Similarly, analysis module 204 determines 504 observed soft signal behaviors and determines a probability of maliciousness for each of the soft signals. In one embodiment, as previously discussed, individual hard signals have a probability of maliciousness high enough to classify the application as malicious. On the other hand, the probability of maliciousness for each of the observed soft signals is combined to calculate 506 a combined probability of maliciousness. If hard signal is observed, or if the combined probability of maliciousness exceeds a maliciousness threshold, the application is classified 508 as malicious. If no hard signal is observed, and if the combined probability of maliciousness fails to exceed the maliciousness threshold, the application is classified as non-malicious.

EX1006, 8:3-28 (emphasis added).

Thus, a PHOSITA would have been motivated to combine Wright and Daswani to minimize false positives. EX1003, ¶113.

Further, a PHOSITA would have known there was a likelihood of success in combining Wright and Daswani because hard and soft signals (based on the definition in Daswani) can be applied directly to the behavior and code genes of Wright, thus there are no difficult or complex steps to integrate the two systems. EX1003, ¶114. A PHOSITA would have recognized that the integration of the soft and hard signals to behavioral genes are most, if not all, of the combination process. Because Wright already teaches a ranking, once the signals are in place, Wright proceeds without modification. Thus, a PHOSITA would recognize that this

combination would require minimal experimentation while giving the PHOSITA high confidence of success. *Id.*

B. Ground 1: Claims 1-24 are unpatentable under 35 U.S.C. § 103(a) as obvious in view of Wright and Daswani

1. Claim 1

a) 1[a] A method for assessing runtime risk for an application program that executes on a device, comprising:

Wright discloses a “method for assessing runtime risk for an application program” in that it describes anti-malware methods for inspecting runtime behaviors of applications, executables, and processes where such run time behaviors can lead to damage and unauthorized access. *Id.* EX1005, 1:23-26. EX1003, ¶118. For example, Wright teaches “assessing the application during its runtime through an analysis of the applications behaviors, in which [t]he behavior monitoring facility 204 may guard against unknown threats by analyzing behavior at runtime for executing software code 202.” EX1005, [0066], EX1003, ¶120.

Wright discloses assessing the “runtime risk” for an application consistent with how the term is used in the ’517 patent because Wright’s system detects sequences of behaviors that correspond to risky or dangerous processes and then groups behaviors to identify malware. EX1003, ¶121.

Wright discloses that detection of risky or dangerous processes as assessed during runtime of the “application program ... execut[ing] on a device” in that it

describes, for example, that the computer code process or “application” being monitored is “executing during the usage session.” EX1005, 1:60-65.

For these reasons, Wright discloses the preamble 1[a] limitations. EX1003, ¶124.

b) 1[b] storing, in a rules database, a plurality of rules, wherein each rule identifies an action sequence;

Wright discloses “a plurality of rules, wherein each rule identifies an action sequence” in that describes behavioral genes and code genes (each of which corresponding to “rules”), and the behavioral and code genes (“rules”) identifies a user behavior and code behavior (each of which corresponding to “action sequence”). For example, Wright teaches that “[t]he indication of the user behavior may be a result of comparing the user interaction, or plurality of user interactions, with a predetermined behavior (hereinafter, a ‘behavioral gene’).” EX1005, [0007] (emphasis added). Wright further teaches that “the indication of the code operation may be a result of comparing an operation with a predetermined code behavior (hereinafter, a ‘code gene’). For example, a code gene may be a virus definition, a malware definition, an adware definition, or some other code behavior.” *Id.*, at [0008] (emphasis added). Thus, Wright discloses that a behavioral gene is compared to a user’s behavior while a code gene is compared to actions of executing software (i.e., code behavior) to identify malware. EX1003, ¶129.

This comparison of user actions or code operations to a predetermined behavior (referred to as “genes”) in Wright matches the ’517 patent explanation that “the runtime risk may be identified based on matching rules and probabilistic weights and scores.” EX1001, 9:8-10 (emphasis added); also 6:14-19; EX1003, ¶131. Based on the descriptions of the behavior genes in Wright, a PHOSITA would have understood the purpose and use of Wright’s “genes” is the same as the “rules” recited in the ’517 patent and claims. See e.g., EX1005, [0007]-[0008]; EX1003, ¶¶127-129.

Wright discloses that all of the behavioral genes and code genes are stored in a database, which corresponds to the recited “rules database.” EX1003, ¶132; EX1005, [0007]-[0008].

As discussed below with respect to claim 1[f] limitation, claim 1 itself defines an “action sequence” as “a sequence of at least two performed actions, and each performed action is at least one of: a user action, an application action, and a system action.”

Wright discloses such action sequences as sequences of actions performed by the user, an application, or the system that cause some sort of effect, or intentional non-effect as a user interacts with the computer and applications. EX1005, [0011], 1:53-2:33; EX1003, ¶133. Wright’s user interactions (behavior genes) and code

actions (code genes) encompass user actions, application actions, and system actions, as Wright discloses:

the user interaction with a computer may include, but is not limited to, executing a link, executing an Internet link, executing an intranet link, downloading a content item from a server, forwarding an URL, forwarding an email, opening an application, closing an application, updating an application, patching an application, bookmarking a website, placing an VoIP call, adding an item to a shopping cart, entering a financial datum in a field, making a purchase, using a webcam, creating an application macro, an application shortcut, entering a novel keystroke combination, remotely accessing a computer, submitting a password, attaching a peripheral device, a temporal measure (e.g., time spent on a page, or time between a first and second user action), a change in a network behavior, a change in a firewall setting, a bridging of two previously unconnected networks, or some other type of user interaction.

EX1005, [0010].

A PHOSITA would understand these example actions correspond to the “action sequences” recited in claim limitation 1[b]. EX1003, ¶134. For example, Wright states:

In embodiments of the present invention, a behavioral based host-intrusion prevention method and system may be used for monitoring a user interaction with a computer, software application, operating system, graphic user interface, or some other component of a computer (e.g. a client) within a computer network, and performing an action to

protect the computer and/or computer network based at least in part on the user interaction. Such protection may be provided based at least in part by monitoring a user interaction with a computer, and/or computer network client device, during a usage session for an indication of a user behavior and monitoring a computer code process executing during the usage session for an indication of a code operation. The indication of the user behavior may be a result of comparing the user interaction, or plurality of user interactions, with a predetermined behavior (hereinafter, a “behavioral gene”). For example, markers of user behaviors with a computer or client device of a computer network may be monitored, recorded, stored, classified, and/ or analyzed based at least in part on the indicia of behavior, and data relating to the behavior, that are created by the user behavior.

EX1005, [0007] (emphasis added).

Wright makes clear that “[t]he indication of the user behavior may be a result of comparing the user interaction, or plurality of user interactions, with a predetermined behavior (hereinafter, a “behavioral gene”). EX1005, [0075] (emphasis added). Wright provides an example:

For example, a behavioral gene called “Internet download” may be created and stored in association with the computer network. This behavioral gene may summarize the user behaviors of 1) executing a hyperlink on the Internet, and 2) performing a “save as” action in which a file is saved to a location in the computer network, such as a hard drive. A single user behavior (e.g., closing an application) may be

recorded and stored as a behavioral gene. Alternatively, a plurality of user behaviors may be recorded within a single behavioral gene.

EX1005, [0007] (emphasis added)

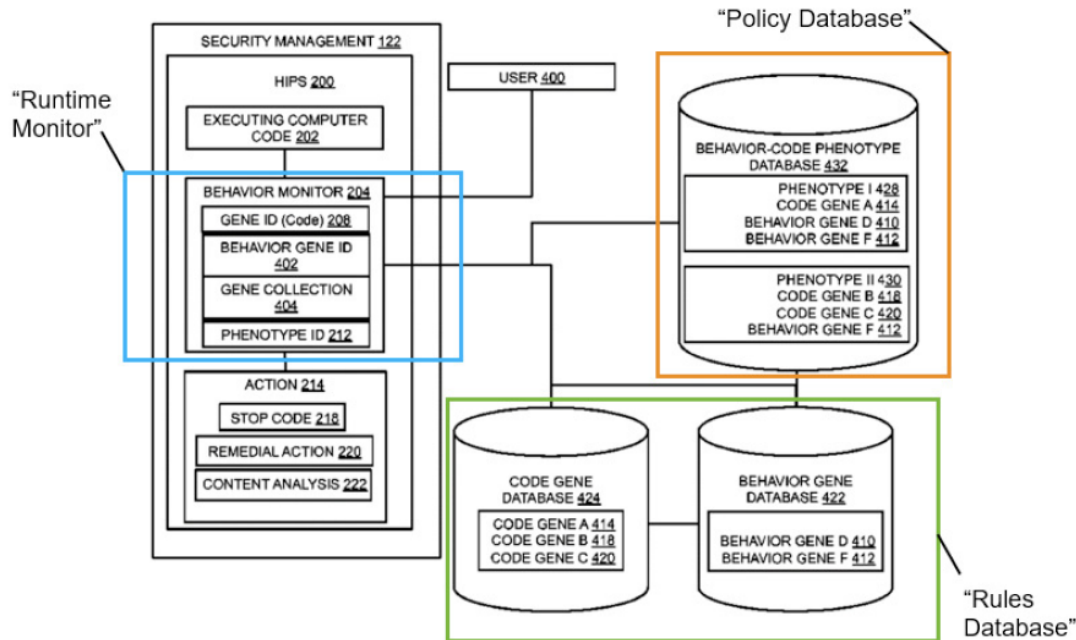
Wright discloses storing behavioral genes (“rules”) in a code gene database and behavior gene database (“storing in a rules database”). EX1005, [0007]-[0008], FIG. 4 (reproduced below); EX1003, ¶92.

For each of the user behaviors occurring within the computer network, or derived from behaviors recorded from interaction with another computer network, a behavioral gene may be stored in a database that is associated with the computer network in which the data relating to the user behavior is recorded and stored....

The behavioral gene and the code gene may be stored for reference in a database.

EX1005, [0007]-[0008] (emphasis added).

Wright Fig. 4 shows the behavioral gene stored in a database.



EX1005, Fig. 4 (annotated); EX1003, ¶133

Wright’s code gene database and behavior gene database, either separately or in combination, are a “rules database” that stores behavioral genes and code genes, each of which corresponding to “rules.” EX1003, ¶132.

Regarding “application actions,” Wright discloses that code operations are associated with code genes to include “system startup, login, scheduled job process, [and] system service.” EX1005, 23:23-25; EX1003, ¶139. A PHOSITA would have understood that a system service is a combination of multiple component operations. EX1003, ¶139. For example, a system startup would involve at least booting a kernel and launching a user interface, and logging in would involve at least comparing a user’s login credentials to a database as well as initiating a controlling process (e.g.,

a terminal, or GUI session) under the user's identity. *Id.* These actions in Wright match the '517 patent examples of "application action":

The application action 304 may include, for example, a start type (e.g., console, service, parent process, script), first time execution, fork process (e.g., child process creation), code injection methods, keyboard capture methods, screen capture methods, dynamic code blocks, open data documents, automatic or self-termination, abnormal terminations (e.g., exception conditions), multiple restarts, opening of device(s) (e.g., the device for which the runtime risk is identified), process enumeration, cryptography, communication protocols, performance metrics, etc. as will be apparent to persons having skill in the relevant art.

EX1001, 7:4-14 (emphasis added.)

Thus, Wright discloses code genes that contain "action sequences" (system startup, login, job processes, or a system service etc.) corresponding to "application action." EX1003, ¶139. Wright discloses changing operating system (OS) tools in a registry and making system modifications as examples of code operations that may be stored as code genes. *Id.* Wright discloses:

In embodiments, indication of a code operation may include, but is not limited to, an automatic code behavior, Such as a system startup, login, scheduled job process, system service, or some other type of code operation. The gene may be at least one of: disables OS tools in a registry, disables a firewall in a registry, adds itself to a fire wall authorized applications list in a registry, injects code into an

application, copies itself to a system folder, sets a run key to itself in a registry, sets a second run key to itself in a registry in a different location, and opens a hidden file with write access. The gene may be at least one of a system modification and a behavior of process. The phenotype may be a combination of more than one gene.

EX1005, 3:44-55.

A PHOSITA would recognize that Wright discloses action sequences of at least two performed “system actions.” EX1003, ¶144. Wright discloses that “system code operations, modifications or code behaviors, also known as code genes...may be identified 208 in the executing code based on code genes (414; 418; 420) that are stored in a code gene database 424.” EX1005, 23:66-24:3, Fig. 4. A PHOSITA would have known that system startup, logging in, starting a scheduled process, and system services include multiple “system actions” including, at least, system file creation, system folder modification, registry modification or additions, application setting modifications, driver install or uninstall, network activity, execution from temporary system files, and so forth. EX1003, ¶¶63-69 (citing EX1007 and EX1008), 144.

For these reasons, Wright discloses 1[b] limitations. EX1003, ¶146.

- c) 1[c] storing, in a policy database, a plurality of assessment policies, wherein each assessment policy includes at least one rule of the plurality of rules;***

As noted in the summary of the '517 patent, the specification explains that “[e]ach assessment policy may comprise one or more rules stored in the rules database 110 and may identify the runtime risk based on matching rules and probabilistic weights and scores.” EX1001, 6:16-22.

As explained regarding limitation 1[b], Wright discloses “rules” in the form of behavior genes and code genes. EX1003, ¶149. Wright further discloses storing in a phenotype database (corresponding to “policy database”) a plurality of phenotypes (corresponding to “assessment policy”) each including a plurality of behavioral genes and code genes (“rules”). EX1005, [0008]; EX1003, ¶149.

Wright discloses “a plurality of assessment policies, wherein each assessment policy includes at least one rule of the plurality of rules” because Wright discloses phenotypes that identify potential malware based on at least one behavioral gene combined with at least one code gene (each corresponding to a “rule”). *Id.* Specifically, Wright discloses that:

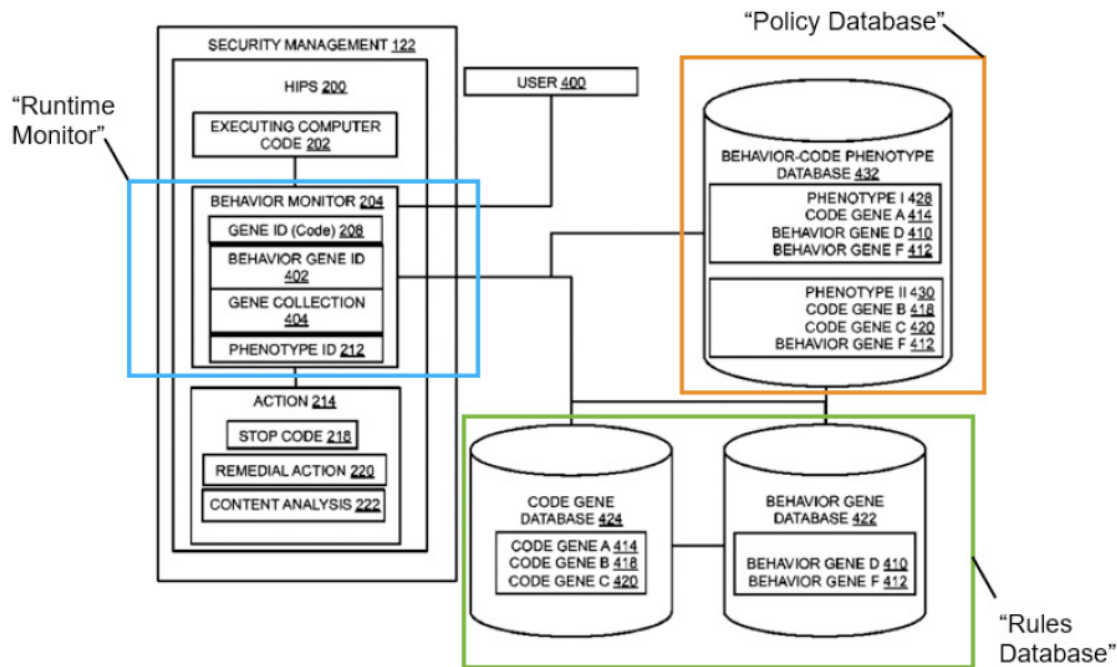
[a] combination of one or more behavioral genes may be combined with one or more code genes and the combination compared to a predetermined collection of user behavior-code operation indications (“phenotypes”). The phenotypes may comprise a grouping of specific behavioral and code gene combinations that are typically present in a type of malicious usage session or operation with a computer.

EX1005, [0008] (emphasis added)

See also Wright FIG. 4 reproduced above regarding limitation 1[b].

Wright discloses that different phenotypes are associated with suspicious behaviors, different phenotypes can be compared against the runtime object's behavior, and a confidence level can be associated with each of the different phenotypes. EX 1003, ¶150. Thus, a PHOSITA would understand that the evaluations of known behavior against actual behavior taught in Wright to be an “assessment policy.” *Id.* This is highlighted by the fact that the '517 patent only describes the “assessment policy” as identifying the runtime risk. *Id.*

Wright discloses “storing, in a policy database” because Wright discloses that the phenotypes are stored in a database. Specifically, Wright states that, “the behavior monitor facility 204 may match an observed phenotype with one stored in a phenotype database 224.” EX1005, [0067] (emphasis added), FIG. 4.



EX1005, Fig. 4 (annotated); EX1003, ¶149

For these reasons, Wright discloses the 1[c] limitations. EX1003, ¶152.

- d) *1[d] identifying, using at least one assessment policy, a runtime risk for an application program that executes on a device, wherein the identified runtime risk indicates a risk or threat of the identified action sequence of the application; and*

Wright discloses assessing the runtime risk of an application that executes on a device because Wright discloses a user's interaction with a software application. EX1003, ¶154. Specifically, Wright describes "a behavioral based host-intrusion prevention method and system for monitoring a user interaction with a computer, software application, ... and performing an action to protect the computer network based at least in part on the user interaction and a computer code process executing

during or in association with a computer usage session.” EX1005, Abstract, (emphasis added). Wright’s system assesses the runtime risk of an application:

HIPS 200 may enable the assignment and temporary storage of Behavioral Genotype data against processes currently running, in order to construct a sequence of runtime behaviors. Matching this runtime genotype data with known combinations stored in the phenotype database 224, with or without an additional content analysis 222, may enable the identification and interruption of malware while it is executing. By matching combinations of behaviors in this way, detection of malware may be improved over solutions where only singular behaviors and a static content analysis is utilized.

EX1005, [0066]

Describing FIG. 4 (reproduced above), Wright explains that “[t]he behavior monitoring facility 204 may guard against unknown threats by analyzing behavior at runtime for executing software code 202.” EX1005, 19:6-8.

The behavior monitoring facility 204 in Wright uses phenotypes (“assessment policy”) to analyze runtime behavior on an application program that executes on a device, in which the runtime behavior indicates a risk or threat of the identified user behavior and code behavior (each an “action sequence”). EX1003, ¶155. “As genes are identified 208 in the executing process, the genes may be collected 210 into a gene collection and may be associated as a phenotype 228,238 that may include the identified genes or operating processes.” EX1005, [0068]. Wright explains that:

The predetermined collections of behaviors, or phenotypes, may be created based on scaled analysis of families of malware in runtime, such as in a threat research facility 132 of a threat management facility 100. Including greater combinations of behaviors in the phenotype database 224 may enable more detailed analysis of a suspicious code. For example, an analysis of a suspicious code may be moved from ‘Accessing the registry in this location is bad’ to ‘This process is exhibiting the same behavior as the family of malware Conficker’ based on a comparison of a detected phenotype with an existing phenotype. This more detailed analysis may eliminate many unwanted detections as the probability of a legitimate application attempting to execute a series of bad behaviors is significantly lower.

Id., at [0069].

Wright discloses “identifying, using at least one assessment policy, a runtime risk for an application program that executes on a device” because Wright discloses that a runtime risk can be identified based on the connection between a recognized phenotype and a known family of malware. EX1003, ¶155. Specifically, Wright discloses:

The behavior monitoring facility 204 may be able to identify a phenotype 212 of user and code behaviors in relation to an executing code by comparing a collection of observed user and code behaviors with the predetermined collections of known malicious user and code behaviors stored as phenotypes in a behavior-code phenotype database 432, either temporarily or for a longer term. Behavior-code phenotypes may capture a combination or a series of user and code behaviors that

may be ranked to create increasing levels of confidence that the runtime object being monitored is executing a behavior pattern comparable to a known family of malware.

EX1005, [0089] (emphasis added).

Wright ranks phenotypes according to their relationship to known families of malware. EX1003, ¶156. Wright explains that phenotypes may identify specific families of malware because Wright states:

Including greater combinations of user and code behaviors in the phenotype database 432 may enable more detailed analysis of a suspicious user behavior or a suspicious code. For example, an analysis of a suspicious code downloaded from a Suspicious site by a user may be moved from “Accessing the registry in this location is bad” to “This process is exhibiting the same behavior as the family of malware Conficker” based on a comparison of a detected phenotype with an existing phenotype.

EX1005, [0069].

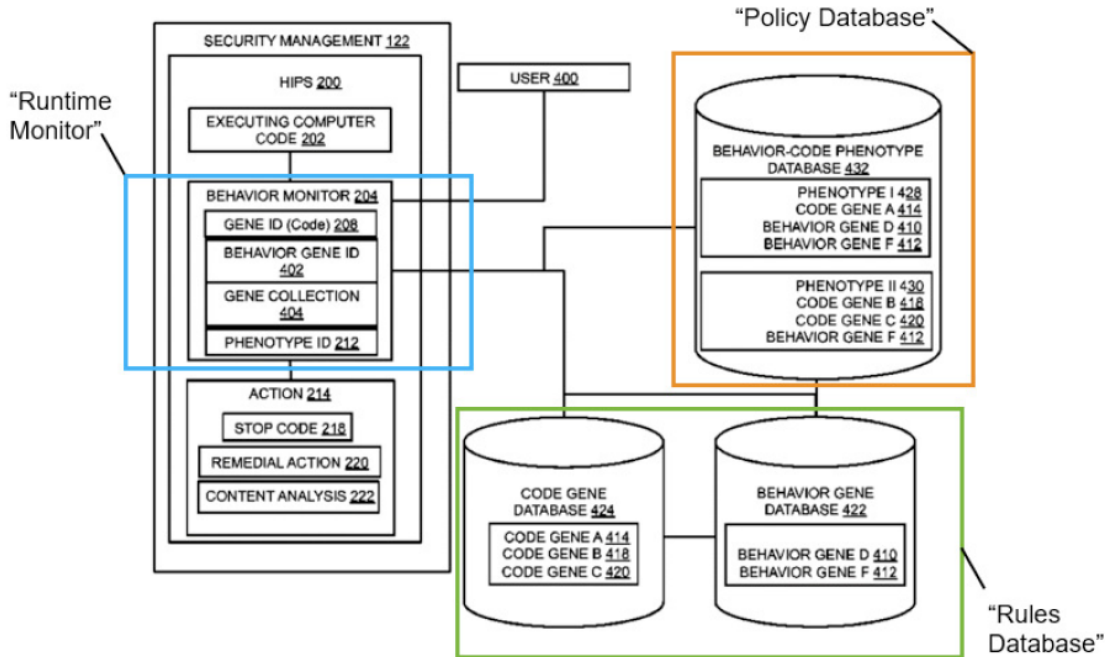
Wright discloses “wherein the runtime risk indicates a risk or threat of the identified action sequence of the application” because Wright identifies a known family of malware as a runtime risk. EX1003, ¶158. Thus, a PHOSITA would recognize an identification of a known family of malware to be a “risk or threat of the identified action sequence.” EX1003, ¶162; EX1001, 6:9-12.

For these reasons, Wright discloses the 1[d] limitations. EX1003, ¶163.

- e) 1[e] identifying, by a runtime monitor including a processing device, a behavior score for the application program that executes on the device based on the identified runtime risk, wherein*

Wright and Daswani disclose a behavior monitoring facility (corresponding to a “runtime monitor including a processing device”) that identifies a “behavior score” for the application program that executes on the device based on the identified “runtime risk.” EX1003, ¶165.

Wright discloses “a runtime monitor” in that it discloses a behavior monitoring facility. EX1005, [0069]. Wright states that, the “behavior monitoring facility 204 may guard against unknown threats by analyzing the behavior of a user 400 and the behavior at runtime for executing software code 202.” EX1005, 23:29-32 (emphasis added). The behavior monitor that is equivalent to the runtime monitor is also illustrated in FIG. 4 reproduced again below.



EX1005, Fig. 4 (annotated); EX1003, ¶149

The behavior monitoring facility (“runtime monitor”) includes a “processing device” because Wright states that a “processor may be any kind of computational or processing device capable of executing program instructions, codes, binary instructions and the like.” *Id.*, 27:4-6. The behavior monitoring facility is a processor because the behavior monitoring facility: “may be software resident on an endpoint. The behavior monitoring facility 204 may monitor executing code and processes, such as access to resources, API calls, registry access, process manipulation, file system access, and the like, in run time by hooking the process and other such resources.” *Id.*, 20:42-46.

As noted in the summary of the '517 patent, the specification explains that a “behavior score” may be a value that indicates the risk or threat posed by the application. *See* EX1001, 6:33-45.

Wright discloses the “behavior score for the application program that executes on the device based on the identified runtime risk” because Wright discloses a ranking system for which a PHOSITA would understand that a “score” is the basis for the ranking probability that the monitored software code (corresponding to the application program that executes on the device) is malware, and thus presenting a risk or threat. EX1003, ¶163. In particular, Wright discloses that the “behavior monitoring facility 204 may be able to identify a phenotype 212 of behaviors in an executing code by comparing a collection of observed behaviors with the predetermined collections of known malicious behaviors” and such phenotypes “may capture a combination or a series of behaviors that may be ranked to create increasing levels of confidence that the runtime object being monitored is executing a behavior pattern comparable to a known family of malware.” EX1005, [0069] (emphasis added). Similarly, Wright discloses that “[b]ehavior-code phenotypes may capture a combination or a series of user and code behaviors that may be ranked to create increasing levels of confidence that the runtime object being monitored is executing a behavior pattern comparable to a known family of malware. *Id.* A PHOSITA would recognize that this ranking of captured user and code behaviors is

equivalent to the “behavior score” recited in this claim limitation as both relate to a measure of the risk posed by a monitored application. EX1003, ¶169.

Alternatively, Wright in view of Daswani discloses the “behavior score for the application program that executes on the device based on the identified runtime risk.” *Id.* at ¶165. While Wright indicates that the combination or series of behaviors will be ranked to create increasing levels of confidence that an application is malware, Daswani discloses a probabilistic scoring to determine an overall likelihood that an application is malicious, and thus poses a threat to the system. *Id.* Daswani discloses hard and soft signals that represent indications of how much a system call or action is likely to contain malware, a significant indicator of risk. Daswani models behaviors as signals:

Behavioral analysis of a mobile application is performed to determine whether the application is malicious. During analysis, various user interactions are simulated in an emulated environment to activate many possible resulting behaviors of an application. The behaviors are classified as hard or soft signals.

EX1006, Abstract.

In one embodiment, some system calls or actions simulated in the emulated environment may be classified as soft signals or hard signals. In one embodiment, soft signals comprise system calls or actions which have an associated probability of the application containing malware. Hard signals comprise system calls or actions that are by definition

malicious actions, or associated with a probability of the application containing malware, the probability exceeding a hard signal threshold.

EX1006, 6:47-55.

Daswani's signal combination module outputs a determination for an application, referred to as the "probability of maliciousness" or " X ," based on a calculation. EX1003, ¶172. Daswani explains how the probability of maliciousness (X) is calculated:

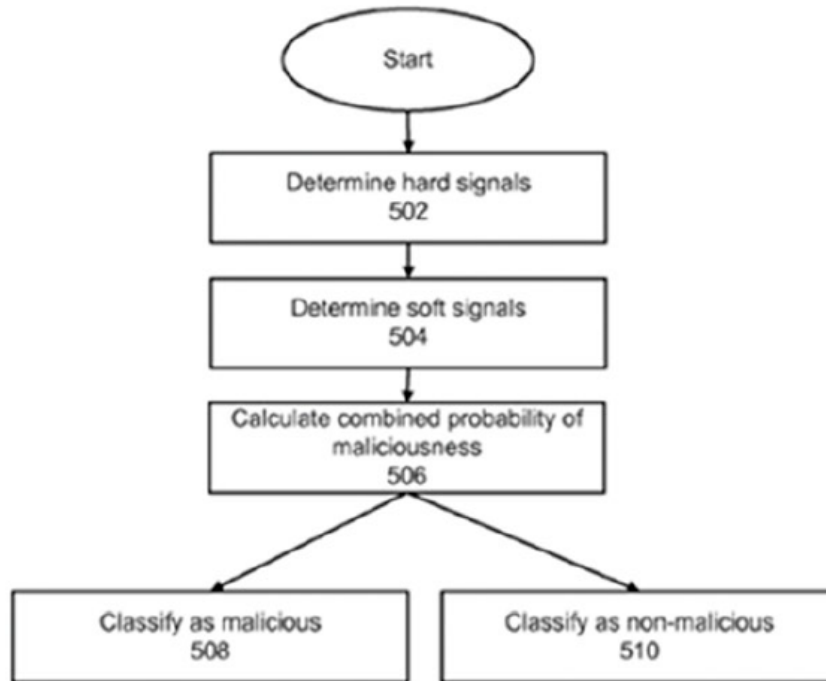
Likewise if one or more independent Soft signals are observed, each of the soft signals has their own probability of indicating that an application is malicious. According to one embodiment, the probability of maliciousness, X , that an application is malicious when a single independent soft signal is observed is shown in equation (1) below, where P is the overall probability that any application contains malware without specific information about the application, D is the detection rate or probability that the soft signal is observed for a malicious application, and F is the false positive rate or probability that the soft signal is observed for a legitimate.

$$X = (D * P) / (D * P + F * (1 - P)) \quad (1)$$

EX1006, 7:9-20.

Consistent with the '517 patent's disclosure for "behavior score", Wright's probability of maliciousness, X , is a number or other value (e.g., a rating) that may indicate the risk or threat of the application or device to the computing system. A PHOSITA would thus recognize or understand Wright's calculated probability of

maliciousness to be a “behavior score” as described in the ’517 Patent. EX1003, ¶174.



EX1005, Fig. 5; EX1003, ¶172

For these reasons, Wright, and Wright in view of Daswani disclose the 1[e] limitations. *Id.*, ¶176.

- f) 1[f] wherein the action sequence is a sequence of at least two performed actions, and each performed action is at least one of: a user action, an application action, and a system action.*

As discussed above regarding claim 1[b] limitation, Wright discloses “wherein the action sequence is a sequence of at least two performed actions” because Wright discloses that “user behavior may be a result of comparing the user

interaction, or plurality of user interactions, with a predetermined behavior” EX1005, [0075] (emphasis added); EX1003, ¶178. Thus, user behavior monitored in Wright can include a sequence of user interactions. Additionally, Wright discloses that observations of user interactions (behavior genes) and code operations (code genes) may be monitored (i.e., at least two actions performed actions). “A combination of one or more behavioral genes may be combined with one or more code genes and the combination compared to a predetermined collection of user behavior-code operation indications (“phenotypes”). The phenotypes may comprise a grouping of specific behavioral and code gene combinations that are typically present in a type of malicious usage session or operation with a computer.” EX1005, [0008].

Also as explained regarding claim 1[b] limitation, user interactions (behavior genes) and code actions (code genes) identified in Wright encompass user actions, application actions, and system actions, as Wright discloses:

the user interaction with a computer may include, but is not limited to, executing a link, executing an Internet link, executing an intranet link, downloading a content item from a server, forwarding an URL, forwarding an email, opening an application, closing an application, updating an application, patching an application, bookmarking a website, placing an VoIP call, adding an item to a shopping cart, entering a financial datum in a field, making a purchase, using a webcam, creating an application macro, an application shortcut,

entering a novel keystroke combination, remotely accessing a computer, Submitting a password, attaching a peripheral device, a temporal measure (e.g., time spent on a page, or time between a first and second user action), a change in a network behavior, a change in a firewall setting, a bridging of two previously unconnected networks, or some other type of user interaction.

EX1005, [0010].

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

For these reasons, Wright discloses the 1[f] limitations. EX1003, ¶¶178-80.

Because Wright in view of Daswani discloses every limitation recited in claim 1 as explained above, a PHOSITA would have found claim 1 obvious in view of Wright and Daswani. *Id.*, ¶192. Accordingly, claim 1 should be found invalid under 35 U.S.C. § 103.

2. Claim 2

- a) The method of claim 1, wherein the user action is any form of explicit input from a user of a computing system.**

Wright discloses that the user behavior (“user action”) can be “any form of explicit input from a user of a computing system” because Wright discloses:

The indication of the user behavior may be a result of comparing the user interaction, or plurality of user interactions, with a predetermined behavior (hereinafter, a “behavioral gene”). For example, markers of user behaviors with a computer or client device of a computer network may be monitored, recorded, stored, classified, and/or analyzed based at least in part on the indicia of behavior, and data relating to the behavior, that are created by the user behavior. For example, a user clicking on a link may be detected, recorded, stored, classified, and/or analyzed. Other sample behaviors may include, but are not limited to, a user scrolling down a page, opening or closing a window, clicking on a link on a webpage, downloading a file, saving a file, running a file, typing a keyword, or some other user interaction. (emphasis added).

EX1005, [0010].

As explained regarding claim limitation 1[b], Wright discloses the action of executing a link as a “user interaction.” EX1003, ¶195. Specifically, Wright discloses user interactions with a computer including:

executing a link, executing an Internet link, executing an intranet link, downloading a content item from a server, forwarding an URL, forwarding an email, opening an application, closing an application, updating an application, patching an application, bookmarking a

website, placing an VoIP call, adding an item to a shopping cart, entering a financial datum in a field, making a purchase, using a webcam, creating an application macro, an application shortcut, entering a novel keystroke combination, remotely accessing a computer, Submitting a password, attaching a peripheral device, a temporal measure (e.g., time spent on a page, or time between a first and second user action), a change in a network behavior, a change in a firewall setting, a bridging of two previously unconnected networks, or some other type of user interaction.

EX1005, [0010].

These examples are consistent with a user action as an “explicit input by the user.” EX1003, ¶196 (citing EX1001 6:56-62).

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 2 limitations as well as claim 1, and thus have found claim 2 obvious in view of Wright and Daswani. EX1003, ¶198. Therefore, claim 2 should be found invalid under 35 U.S.C. §103.

3. Claim 3

- a) The method of claim 1, wherein the application action is any activity performed by the application initiated programmatically by a task in execution of a computing system.***

Wright discloses the code process executed during a usage session (“application action”) is “any activity performed by the application initiated programmatically by a task in execution of a computing system.”

As shown regarding limitations 1[b] and 1[f], Wright discloses application actions that include “an automatic code behavior, such as a system startup, login, scheduled job process, [and] system service.” EX1005, [0081]; EX1003, ¶201. A PHOSITA would recognize that the “system startup”, “scheduled job process”, and “system service” involve several application actions initiated programmatically by a task (i.e., startup or login) in execution of a computing system. System startup involves at least booting a kernel and launching a user interface, a schedule job process is a program launched programmatically on a specific schedule or timer, and a system service is a background application that can be launched automatically on system start-up, login, or in response to another program. EX1003, ¶¶63-69, 203.

Further, Wright discloses monitor behaviors of software code executing on a device, including “[c]ertain system code operations, modifications or code behaviors ...in the executing code based on code genes (414, 418, 420) that are stored in a code gene database 424.” EX1005, [0084]. “The indication of a malicious code behavior may be a result of comparing an operation of the executing code with a predetermined code gene, such as a gene stored for reference in a code gene database.” *Id.*, at [0086].

Wright provides an example of monitoring code behaviors associated with the W32/Sality virus involving monitoring for application actions initiated programmatically including “disables OS tools in the registry, disables the MS

windows firewall in the registry, adds itself to the firewall authorized applications list in the registry, and injects code into explorer.exe.” *Id.*, at [0092]. Wright also provides an example of monitoring code behaviors associated with the W32/AutoRun worm involving monitoring for application actions initiated programmatically including: “copies itself to system folder, sets a run key to itself in the registry, sets a second run key to itself in the registry in a different location, disables OS tools in the registry, and opens a hidden file with name Autorun.inf with write access.” *Id.*, at [0093].

Further, Wright discloses a computer code process executed during at least one usage session. *Id.*, [0094]. Wright discloses that a usage session can be “interactions with one or more software applications, and the associated execution code.” *Id.*, [0083] (emphasis added).

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 3 limitations as well as claim 1, and thus have found claim 3 obvious in view of Wright and Daswani. EX1003, ¶205. Therefore, claim 3 should be found invalid under 35 U.S.C. §103.

4. **Claim 4**

- a) ***The method of claim 1, wherein the system action is any operation performed by a computing system on behalf of, or as a consequence of, a user action or application action that changes the state of the computing system.***

Wright discloses the system code operations (corresponding to “system action”) is “any operation performed by a computing system on behalf of, or as a consequence of” a user behavior (“user action”) or code process executed during a usage session (“application action”) that changes the state of the computing system.

As explained regarding limitations 1[b] and 1[f], a PHOSITA would recognize system startup, logging in, starting a scheduled process, and system services to be multiple “system actions,” such as system file creation, system folder modification, registry modification or additions, application setting modifications, driver install or uninstall, network activity, execution from temporary system files, etc. EX1003, ¶209. A PHOSITA would have known that these actions are created on behalf of users or applications because users and applications cannot perform these operations directly. *Id.* at ¶¶63-69, 209. For example, system file creation, system folder modification, and registry changes are restricted to system privileges (either the OS, a “System” user, or a “root” user), so users and applications cannot perform these operations directly. *Id.*

Further, Wright states that “[c]ertain system code operations, modifications or code behaviors, also known as code genes (414, 418 and 420), may be identified 208

in the executing code based on code genes (414; 418; 420) that are stored in a code gene database 424.” EX1005, [0084]. For example, Wright identifies among monitored user interactions “attaching a peripheral device, ... a change in a firewall setting, a bridging of two previously unconnected networks” all of which involve a change to the state of the computing system. EX1005, [0010]; EX1003, ¶208. Wright further discloses that a code gene “may be at least one of a system modification and a behavior of process.” EX1005, [0014].

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 4 limitations as well as claim 1, and thus have found claim 4 obvious in view of Wright and Daswani. EX1003, ¶211. Therefore, claim 4 should be found invalid under 35 U.S.C. §103.

5. Claim 5

- a) The method of claim 1, wherein a first action of the at least two performed actions is performed at a first time, and the second action of the at least two performed actions is performed at a second time.***

Wright discloses that one of the two “actions”—where each action can be user behavior (“user action”), the code process executed during a usage session (“application action”), and the system code operations (“system action”)—“is performed at a first time” and another one of the actions “is performed at a second time.

Wright discloses that these different “actions” can be performed at different times because behavioral genes and code genes can progressively take place at different times. EX1005, [0087]. A PHOSITA would have understood tracking behavior over time means that one action can be performed at a time before a second action. EX1003, ¶215. Specifically, Wright states that:

A plurality of malicious user and code behavior indications may be collected in this fashion by comparing a user behavior or a behavior of an executing code with a predetermined behavior. Behavior and code genes may continue to be identified progressively over the entire course of a code execution so that at any time during execution, the behavior monitor facility 204 may match an observed phenotype with one stored in a behavior-code phenotype database.

EX1005, [0087] (emphasis added).

Additionally, Wright discloses the limitations of claim 5 with the following example, “a first usage session in which an Internet browser is used, and a second usage session in which a file is opened that was downloaded to the users hard drive during the first usage session.” EX1005, [0075].

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 5 limitations as well as claim 1, and thus have found claim 2 obvious in view of Wright and Daswani. EX1003, ¶217. Therefore, claim 5 should be found invalid under 35 U.S.C. §103.

6. Claim 6

- a) The method of claim 5, wherein the first action is a user action, the second action is a system action, and the second time occurs after the first time.***

As shown regarding claim 5, Wright discloses that these different “actions” can be performed at different times because behavioral genes and code genes can progressively take place at different times. EX1005, [0087].

As shown regarding claim 4, A PHOSITA would have known that “logging in” is a user action that can trigger system actions, such as terminating execution of /bin/login (the login program), asking for a password, encrypting an entered password, verifying the password against the encrypted password stored in the password file (/etc/passwd), and if correct, replacing the login application with the user’s shell, which then waits for the first command. EX1003, ¶¶66, 220.

For this reason, a PHOSITA would understand that Wright and Daswani disclose the claim 6 limitations as well as claims 1 and 5, and thus have found claim 6 obvious in view of Wright and Daswani. *Id.*, ¶221. Therefore, claim 6 should be found invalid under 35 U.S.C. §103.

7. Claim 7

- a) The method of claim 5, wherein the first action is a user action, the second action is an application action, and the second time occurs after the first time.***

As shown regarding claim 5, Wright discloses that different “actions,” which include user behavior, can be performed and collected at different times. EX1005, [0084]-[0087]; EX1003, ¶224.

Wright explains that user interactions, such as clicking on a link, also cause code actions, such as downloading content from a server or forwarding a URL.

In embodiments, the user interaction with a computer may include, but is not limited to, executing a link, executing an Internet link, executing an intranet link, downloading a content item from a server, forwarding an URL, forwarding an email, opening an application, closing an application, updating an application, patching an application, bookmarking a website, placing an VoIP call, adding an item to a shopping cart, entering a financial datum in a field, making a purchase, using a webcam, creating an application macro, an application shortcut, entering a novel keystroke combination, remotely accessing a computer, Submitting a password, attaching a peripheral device, a temporal measure (e.g., time spent on a page, or time between a first and second user action), a change in a network behavior, a change in a firewall setting, a bridging of two previously unconnected networks, or some other type of user interaction.

EX1005, [0010].

In embodiments, indication of a code operation may include, but is not limited to, an automatic code behavior, such as a system startup, login, scheduled job process, system service, or some other type of code operation. The gene may be at least one of: disables OS tools in a registry, disables a firewall in a registry, adds itself to a fire wall authorized applications list in a registry, injects code into an application, copies itself to a system folder, sets a run key to itself in a registry, sets a second run key to itself in a registry in a different location, and opens a hidden file with write access. The gene may be at least one of a system modification and a behavior of process. The phenotype may be a combination of more than one gene.

EX1005, [0013]-[0014].

A PHOSITA would have recognized that these user interactions described in Wright trigger one or more code actions (corresponding to “application actions”). EX1003, ¶225.

Accordingly, a PHOSITA would understand that Wright and Daswani disclose the claim 7 limitations as well as claims 1 and 5, and thus have found claim 7 obvious in view of Wright and Daswani. EX1003, ¶227. Therefore, claim 7 should be found invalid under 35 U.S.C. §103.

8. Claim 8

- a) The method of claim 5, wherein the first action is an application action, the second action is a system action, and the second time occurs after the first time.***

As shown regarding claim 5, Wright discloses that different “actions,” which include code behavior, can be performed and collected at different times. EX1005, [0084]-[0087]; EX1003, ¶230.

As shown regarding limitation 1[g], a PHOSITA would have known that system startup, logging in, starting a scheduled process, and system services (which are “application actions”) would trigger “system actions”; in which “system actions” at least include: system file creation, system folder modification, registry modification or additions, application setting modifications, driver install or uninstall, network activity, execution from temporary system files. EX1003, ¶231.

Accordingly, a PHOSITA would understand that Wright and Daswani disclose the claim 8 limitations as well as claims 1 and 5, and thus have found claim 8 obvious in view of Wright and Daswani. EX1003, ¶233. Therefore, claim 8 should be found invalid under 35 U.S.C. §103.

9. **Claim 9**

- a) ***The method of claim 1, wherein the action sequence includes three performed actions, a first action of the three performed actions is a user action, a second action of the three performed actions is an application action, a third action of the three performed actions is a system action, the second action is performed temporally after the first action, and the third action is performed temporally after the second action.***

As shown regarding claim 5, Wright discloses that user actions trigger code operations (“application actions”) and code operations (i.e., startup, logging in, starting a scheduled process, and system services) trigger “system actions” (i.e. system file creation, system folder modification, registry modification or additions, application setting modifications, driver install or uninstall, network activity, execution from temporary system files). EX1005, [0013]-[0014]; EX1003, ¶¶234-238.

Accordingly, a PHOSITA would understand that Wright and Daswani disclose the claim 9 limitations as well as claim 1, and thus have found claim 9 obvious in view of Wright and Daswani. EX1003, ¶239. Therefore, claim 9 should be found invalid under 35 U.S.C. §103.

10. Claim 10

- a) The method of claim 1, wherein the runtime risk is identified based on matching rules and probabilistic weights and scores of performed actions on the device.***

Wright and Daswani disclose “the runtime risk is identified based on matching rules and probabilistic weights and scores of performed actions on the device.” Wright describes combining (“matching”) user and code behaviors in the phenotype database to enable more detailed analysis of a suspicious user behavior or a suspicious code. Specifically, Wright states:

Behavior-code phenotypes may capture a combination or a series of user and code behaviors that may be ranked to create increasing levels of confidence that the runtime object being monitored is executing a behavior pattern comparable to a known family of malware. The predetermined collections of behaviors, or phenotypes, may be created based on Suspicious or risky user behavior (e.g. downloading a file from a website, running an executable, opening a signed file with an unverifiable digital signature, or other potentially suspicious or risky behavior) and scaled analysis of families of malware in runtime, such as in a threat research facility 132 of a threat management facility 100. Including greater combinations of user and code behaviors in the phenotype database 432 may enable more detailed analysis of a Suspicious user behavior or a Suspicious code.

EX1005, [0089] (emphasis added).

Wright discloses comparing (“matching”) detected phenotypes and existed phenotypes to reduce the probability of bad behaviors. Specifically, Wright states:

[A]nalysis of a suspicious code downloaded from a Suspicious site by a user may be moved from “Accessing the registry in this location is bad” to “This process is exhibiting the same behavior as the family of malware Conficker” based on a comparison of a detected phenotype with an existing phenotype. This more detailed analysis may eliminate many unwanted detections as the probability of a legitimate application attempting to execute a series of bad behaviors is significantly lower. Equally, this more specific detection of the family of malware may enable clean up to be called that can look for other known components of the threat, thus enabling a more comprehensive clean up rather than a simple process killing. Analyzing the user behavior information in connection with the code behavior may provide additional accuracy for determining whether a monitored set of behaviors is either likely legitimate or likely illegitimate. For example, if software code is associated with a user behavior, such as having downloaded that content from the Internet, a remedial action 220 may be taken since the combination, or phenotype, of the code in conjunction with the user’s downloading behavior may cause more concern with the code and its potential harm to the computer network.

EX1005, [0089] (emphasis added).

Daswani discloses “probabilistic weights and scores of performed actions.” Daswani provides further detail to the probability analysis of Wright because Daswani discloses probabilistic scoring to determine an overall likelihood that an application is malicious. EX1003, ¶244. Daswani discloses hard and soft signals that represent indications of how much a system call or action is likely to contain

malware, a significant indicator of risk. *Id.* at ¶245. Each “signal” models behavior and represents rules and probabilistic weights. *Id.* Specifically, Daswani states:

Behavioral analysis of a mobile application is performed to determine whether the application is malicious. During analysis, various user interactions are simulated in an emulated environment to activate many possible resulting behaviors of an application. The behaviors are classified as hard or soft signals.

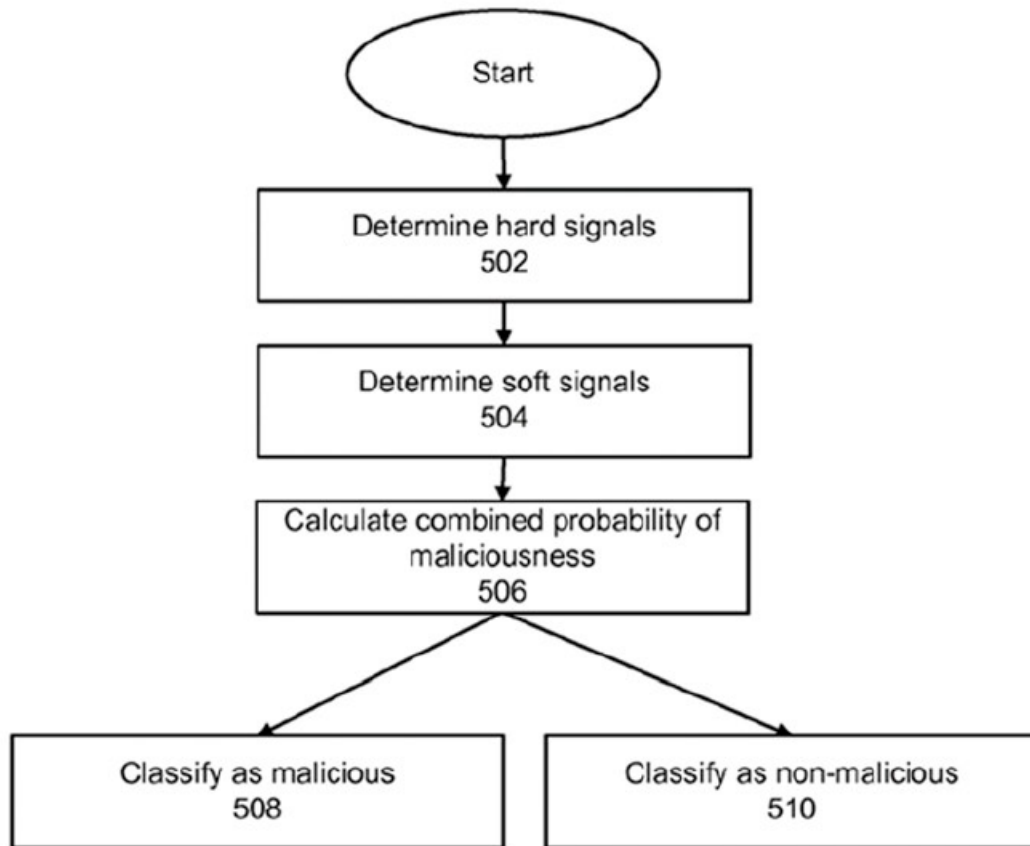
EX1006, Abstract.

In one embodiment, some system calls or actions simulated in the emulated environment may be classified as soft signals or hard signals. In one embodiment, soft signals comprise system calls or actions which have an associated probability of the application containing malware. Hard signals comprise system calls or actions that are by definition malicious actions, or associated with a probability of the application containing malware, the probability exceeding a hard signal threshold.

EX1006, 6:47-55.

Signal combination module 306 combines multiple behaviors signals to determine an overall likelihood that an application is malicious.

EX1006, Fig. 5, 6:45-47.



EX1005, Fig. 5; EX1003, ¶245.

Daswani's signal combination module outputs a determination for an application, malicious or non-malicious, based on a computed score. Daswani refers to this score as the "probability of maliciousness." Daswani states:

Likewise if one or more independent Soft signals are observed, each of the soft signals has their own probability of indicating that an application is malicious. According to one embodiment, the probability of maliciousness, X , that an application is malicious when a single independent soft signal is observed is shown in equation (1) below, where P is the overall probability that any application contains malware without specific information about the application, D is the detection

rate or probability that the soft signal is observed for a malicious application, and F is the false positive rate or probability that the soft signal is observed for a legitimate.

EX1006, 7:9-20.

$$X = (D * P) / (D * P + F(1 - P)) \quad (1)$$

Id., Equation 1. As noted above, Wright’s probability of maliciousness, *X*, is a number or other value (e.g., a rating) that may indicate the risk or threat of the application or device to the computing system. A PHOSITA would thus recognize or understand Wright’s calculated probability of maliciousness to be a “behavior score” as described in the ’517 Patent. EX1003, ¶170.

A PHOSITA would have been motivated to combine Daswani’s soft and hard signals analysis with the method disclosed in Wright as described above. EX1003, ¶248.

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 10 limitations as well as claim 1, and thus have found claim 10 obvious in view of Wright and Daswani. *Id.*, ¶249. Therefore, claim 10 should be found invalid under 35 U.S.C. §103.

11. Claim 11

- a) The method of claim 1, wherein the runtime risk is identified based on time proximity of the at least two performed actions of the action sequence and natural affinity inferred between the at least two performed actions.**

Wright discloses that the runtime risk is identified based on temporal measure (corresponding to “time proximity”) of at least two of the user behavior, code process, or system code operations (“actions” and “action sequences”) and “natural affinity inferred between the at least two performed actions.” EX 1003, ¶251.

Wright discloses the behavior monitoring facility may guard against threats due to runtime “by analyzing the behavior of a user 400 and the behavior at runtime for executing software code 202.” EX1005, [0081]. Wright further discloses that behavior of the user can be analyzed by temporal measure between a first user action and a second user action. EX1005, [0010]. Wright states that, “the user interaction with a computer may include, but is not limited to... a temporal measure (e.g., time spent on a page, or time between a first and second user action).” *Id.* A PHOSITA would have understood this temporal measure based on a time between a first user action and second user action to be a “time proximity of the at least two performed actions of the action sequence.” EX1003, ¶253.

Further, Wright discloses that the two actions that define the “temporal measure” may have a “natural affinity” (i.e., “time spent on a page”) that is

“inferred” as the actions can be related. *Id.*, ¶254. A PHOSITA would have understood that the time spent on a webpage is commonly measured by the time between two user actions; a first action executing a link to the page (when the user is “on” the page), and second action executing a link to another page (when the user is “off” the page). *Id.*

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 11 limitations as well as claim 1, and thus have found claim 11 obvious in view of Wright and Daswani. EX1003, ¶255. Therefore, claim 11 should be found invalid under 35 U.S.C. §103.

12. Claim 12

- a) 12[a] The method of claim 1, further comprising: assessing, by the processing device, that the at least two performed actions of the action sequence describe a forensic chain of events based on a plurality of factors,***

Wright discloses the behavior monitoring facility (“processing device”) assessing at least two performed user behaviors (“actions”) that “describe a forensic chain of events based on a plurality of factors.” EX1003, ¶258.

Wright discloses “the at least two performed actions of the action sequence describe a forensic chain of events based on a plurality of factors” because Wright disclose applying scientific methods to identify malware by phenotypes, which are specific combinations of actions. EX1003, ¶¶259-260. Wright states:

A combination of one or more behavioral genes may be combined with one or more code genes and the combination compared to a predetermined collection of user behavior-code operation indications (“phenotypes”). The phenotypes may comprise a grouping of specific behavioral and code gene combinations that are typically present in a type of malicious usage session or operation with a computer. EX1005, [0008].

Wright discloses that a phenotype has at least two performed actions where one of the actions is a user “gene” and another action is a code “gene.” EX1003, ¶260. Thus, a PHOSITA would have recognized “phenotypes,” which are used to identify malicious usage, are a “forensic chain.” *Id.*

For these reasons, Wright and Daswani disclose the 12[a] limitations. EX1003, ¶261.

b) 12[b] wherein the plurality of factors includes at least one of: ordered sequences, unordered sequences, proximity analysis, outliers, probabilistic weights and scores, and probable action inferences based on an application's competency.

Wright discloses that user behaviors that describe the forensic chain can be based on at least “probable action inferences”, “ordered sequences”, and “unordered sequences.”

Wright discloses that a phenotype (corresponding to “forensic chain of actions”) includes an “ordered sequence,” an “unordered sequence”, or “proximity analysis.” EX1003, ¶264. The phenotype includes at least one user behavior gene,

and one code gene, and a PHOSITA would have known that this is either an ordered sequence of actions or an unordered sequence of actions. *Id.* at ¶265.

As shown regarding Claim 11, Wright discloses that user actions could include a measurement of time between two actions, and a PHOSITA would have recognized this as “proximity analysis.” *Id.*

Accordingly, Wright and Daswani disclose at least one of the alternative limitations listed in 12[b], any one of which is sufficient to render the entire limitation obvious. *See e.g., Titanium Metals Corp*, 778 F.2d at 782 (Fed. Cir. 1985).

Thus, Wright and Daswani disclose the 12[b] limitations. EX1003, ¶266.

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 12 limitations as well as claim 1, and thus have found claim 12 obvious in view of Wright and Daswani. *Id.* Therefore, claim 12 should be found invalid under 35 U.S.C. §103.

13. Claim 13

a) 13[a] *A system for assessing runtime risk for an application program that executes on a device, comprising:*

Limitation 13[a] recites limitations equivalent to the limitations recited in limitation 1[a] except reciting “system.” Wright discloses “behavior monitoring systems.” EX1005, claim 1; EX1003, ¶269. Wright further discloses that its systems can be run on a machine that executes computer software, program codes, and/or

instructions on a processor. EX1005, 26:61-27:4, EX1003, ¶¶270-272. As shown regarding limitation 1[a], Wright and Daswani disclose the rest of the limitations recited in limitation 13[a]. EX1005, claim 1; EX1003, ¶273.

b) 13[b] a rules database storing a plurality of rules, wherein each rule identifies an action sequence;

Limitation 13[b] recites limitations equivalent to the limitations recited in limitation 1[b]. As shown regarding limitation 1[b], Wright and Daswani disclose the 13[b] limitations. EX1005, [0007]-[0008]. EX1003, ¶275.

c) 13[c] a policy database storing a plurality of assessment policies, wherein each assessment policy includes at least one rule of the plurality of rules; and

Limitation 13[c] recites limitations equivalent to the limitations recited in limitation 1[c]. As shown regarding limitation 1[c], Wright and Daswani disclose the 13[c] limitations. EX1005, [0007]-[0008]; EX1003, ¶277.

d) 13[d] a runtime monitor including a processing device

Limitation 13[d] recites limitations equivalent to the limitations recited in limitation 1[e]. As shown regarding limitation 1[e], Wright and Daswani disclose the 13[d] limitations. EX1003, ¶279.

- e) 13[e] identifying, using at least one assessment policy, a runtime risk for an application program that executes on a device, wherein the identified runtime risk indicates a risk or threat of the identified action sequence of the application, and***

Limitation 13[e] recites limitations equivalent to the limitations recited in limitation 1[d]. As shown regarding limitation 1[d], Wright and Daswani disclose the 13[e] limitations. EX1003, ¶281.

- f) 13[f] identifying a behavior score for the application program that executes on the device based on the identified runtime risk, wherein,***

Limitation 13[f] recites limitations equivalent to the limitations recited in limitation 1[e]. As shown regarding limitation 1[e], Wright and Daswani disclose the element 13[f] limitations. EX1003, ¶283.

- g) 13[g] the action sequence is a sequence of at least two performed actions, and***

Limitation 13[g] recites limitations equivalent to the limitations recited in limitation 1[f]. As shown regarding limitation 1[f], Wright and Daswani disclose the 13[g] limitations. EX1003, ¶285.

- h) 13[h] each performed action is at least one of: a user action, an application action, and a system action.***

Limitation 13[h] recites limitations equivalent to the limitations recited in limitation 1[g]. As shown regarding limitation 1[g], Wright and Daswani disclose the 13[h] limitations. EX1003, ¶287.

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 13 limitations, and thus have found claim 13 obvious in view of Wright and Daswani. EX1003, ¶288. Therefore, claim 13 should be found invalid under 35 U.S.C. §103.

14. Claim 14

- a) *The system of claim 13, wherein the user action is any form of explicit input from a user of a computing system.***

Claim 14 recites limitations equivalent to the limitations recited in claim 2. As shown in the analysis of claim 2 and claim 13, Wright and Daswani disclose the claim 14 limitations as well as claim 13. EX1003, ¶290.

Accordingly, a PHOSITA would have found claim 14 obvious in view of Wright and Daswani. *Id.* Therefore, claim 14 should be found invalid under 35 U.S.C. §103.

15. Claim 15

- a) *The system of claim 13, wherein the application action is any activity performed by the application initiated programmatically by a task in execution of a computing system.***

Claim 15 recites limitations equivalent to the limitations recited in claim 3. As shown regarding claim 3 and claim 13, Wright and Daswani disclose the claim 15 limitations as well as claim 13. EX1003, ¶292. Accordingly, a PHOSITA would have found claim 15 obvious in view of Wright and Daswani. *Id.* Therefore, claim 15 should be found invalid under 35 U.S.C. §103.

16. Claim 16

- a) The system of claim 13, wherein the system action is any operation performed by a computing system on behalf of, or as a consequence of, a user action or application action that changes the state of the computing system.**

Claim 16 recites limitations equivalent to the limitations recited in claim 4. As shown regarding claim 4, Wright and Daswani disclose the claim 16 limitations as well as claim 13. Accordingly, a PHOSITA would have found claim 16 obvious in view of Wright and Daswani. EX1003, ¶294. Therefore, claim 16 should be found invalid under 35 U.S.C. §103.

17. Claim 17

- a) The system of claim 13, wherein a first action of the at least two performed actions is performed at a first time, and the second action of the at least two performed actions is performed at a second time.**

Claim 17 recites limitations equivalent to the limitations recited in claim 5. As shown regarding claims 5 and 13, Wright and Daswani disclose the claim 17 limitations as well as claim 13. EX1003, ¶296. Accordingly, a PHOSITA would have found claim 17 obvious in view of Wright and Daswani. *Id.* Therefore, claim 17 should be found invalid under 35 U.S.C. §103.

18. Claim 18

- a) *The system of claim 17, wherein the first action is a user action, the second action is a system action, and the second time occurs after the first time.***

Claim 18 recites limitations equivalent to the limitations recited in claim 6. As shown regarding claims 6 and claim 17, Wright and Daswani disclose the claim 18 limitations as well as claim 17. EX1003, ¶298. Accordingly, a PHOSITA would have found claim 18 obvious in view of Wright and Daswani. *Id.* Therefore, claim 18 should be found invalid under 35 U.S.C. §103.

19. Claim 19

- a) *The system of claim 17, wherein the first action is a user action, the second action is an application action, and the second time occurs after the first time.***

Claim 19 recites limitations equivalent to the limitations recited in claim 7. As regarding claim 7, Wright and Daswani disclose the claim 19 limitations as well as claim 17. EX1003, ¶300. Accordingly, a PHOSITA would have found claim 19 obvious in view of Wright and Daswani. *Id.* Therefore, claim 19 should be found invalid under 35 U.S.C. §103.

20. Claim 20

- a) *The system of claim 17, wherein the first action is an application action, the second action is a system action, and the second time occurs after the first time.***

Claim 20 recites limitations equivalent to the limitations recited in claim 8. As shown regarding claims 8 and 17, Wright and Daswani disclose the claim 20

limitations as well as claim 17. EX1003, ¶302. Accordingly, a PHOSITA would have found claim 20 obvious in view of Wright and Daswani. *Id.* Therefore, claim 20 should be found invalid under 35 U.S.C. §103.

21. Claim 21

- a) The system of claim 17, wherein the action sequence includes three performed actions, a first action of the three performed actions is a user action, a second action of the three performed actions is an application action, a third action of the three performed actions is a system action, the second action is performed temporally after the first action, and the third action is performed temporally after the second action.***

Claim 21 recites limitations equivalent to the limitations recited in claim 9. As shown regarding claim 9, Wright and Daswani disclose the claim 21 limitations as well as claim 17. EX1003, ¶304. Accordingly, a PHOSITA would have found claim 21 obvious in view of Wright and Daswani. *Id.* Therefore, claim 21 should be found invalid under 35 U.S.C. §103.

22. Claim 22

- a) The system of claim 13, wherein the runtime risk is identified based on matching rules and probabilistic weights and scores of performed actions on the device.***

Claim 22 recites limitations equivalent to the limitations recited in claim 10. As shown regarding claims 10 and 13, Wright and Daswani disclose the claim 22 limitations as well as claim 13. EX1003, ¶306. Accordingly, a PHOSITA would

have found claim 22 obvious in view of Wright and Daswani. *Id.* Therefore, claim 22 should be found invalid under 35 U.S.C. §103.

23. Claim 23

- a) The system of claim 13, wherein the runtime risk is identified based on time proximity of the at least two performed actions of the action sequence and natural affinity inferred between the at least two performed actions.*

Claim 23 recites limitations equivalent to the limitations recited in claim 11. As shown regarding claim 11, Wright and Daswani disclose the claim 23 limitations as well as claim 13. EX1003, ¶308. Accordingly, a PHOSITA would have found claim 23 obvious in view of Wright and Daswani. *Id.* Therefore, claim 23 should be found invalid under 35 U.S.C. §103.

24. Claim 24

- a) 24[a] The system of claim 13, wherein the processing device assesses that the at least two performed actions of the action sequence describe a forensic chain of events based on a plurality of factors, and*

Limitation 24[a] recites limitations equivalent to the limitations recited in claim limitation12[a]. As shown regarding limitation 12[a], the combination of Wright and Daswani discloses the 24[a] limitations. EX 1003, ¶311.

- b) 24[b] the plurality of factors includes at least one of: ordered sequences, unordered sequences, proximity analysis, outliers, probabilistic weights and scores, and probable action inferences based on an application's competency.**

Limitation 24[b] recites limitations equivalent to the limitations recited in claim limitation 12[b]. As shown regarding limitation 12[b], Wright and Daswani disclose at least one of the alternative limitations of claim 24[b], any one of which is sufficient to render the entire limitation obvious. EX1003, ¶313. *See e.g., Titanium Metals Corp.*, 778 F.2d at 782 (Fed. Cir. 1985).

For these reasons, a PHOSITA would understand that Wright and Daswani disclose the claim 24 limitations as well as claim 13, and therefore a PHOSITA would have found claim 24 obvious in view of Wright and Daswani. EX1003, ¶314. Therefore, claim 24 should be found invalid under 35 U.S.C. §103.

VI. NO BASIS FOR DISCRETIONARY DENIAL

There is no basis to discretionarily deny institution. Under *Fintiv*, the Board may consider parallel litigation, including an early trial date, in determining whether to institute under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd., v. Intri-Plex Technologies, Inc.*, IPR2018-00752, Paper 8 at 19–20 (PTAB Sept. 12, 2018) (precedential); *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 2–3, 6 (PTAB Mar. 20, 2020) (precedential). As detailed below, there is no operative trial date set between the parties.

Beginning on August 3, 2022, pursuant to 28 U.S.C. §1407(a), the United States Judicial Panel on Multidistrict Litigation (“MDL Panel”) centralized several litigations from various districts involving Patent Owner Taasera in the Eastern District of Texas (“EDTX”), including the litigation involving the parties to this proceeding.³ EX1010. The centralized proceedings were assigned to Judge Gilstrap for pretrial management only and must be statutorily remanded to the originating court before trial. 28 U.S.C. §1407(a) (“Each action so transferred shall be remanded by the panel at or before the conclusion of such pretrial proceedings to the district from which it was transferred unless it shall have been previously terminated”). While Judge Gilstrap has set trial for EDTX cases on April 15, 2024, no trial has been set in the originating district. EX1011, 1 (noting the “**trial date applies only to cases originally filed in [EDTX]**”). Therefore, trial must ultimately be set by the eventual trial court pending that court’s trial calendar.

Once back at the originating trial court, Petitioner intends to file a motion to transfer the case to California, a motion to stay pending that transfer determination, and a motion to stay pending IPR at the transferee forum, leaving a trial date even more uncertain. Ex. 1012, 3 (inviting “Motions to Transfer upon remand by the panel...to its originating court”).

³ *Taasera Licensing LLC v. CrowdStrike, Inc., CrowdStrike Holdings, Inc.*, No. 2:22-CV-00468-JRG.

Further, Petitioner stipulates that if instituted, Petitioner will not seek resolution in the District Court of any ground of invalidity for the '517 Patent that utilizes *Wright* or *Daswani*—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 '517 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.

Therefore, with an unknown trial date and any overlap mitigated by stipulation, denial under § 314(a) is not warranted.

Nor is discretionary denial under §325(d) warranted. *Wright* and *Daswani* were not cited nor considered during prosecution and are materially different than the references asserted in Office communications because they disclose the “wherein the identified runtime risk indicates a risk or threat of the identified action sequence of the application”, and “a runtime monitor including a processing device” limitations not purportedly found in the prior art at issue during prosecution. *Amperex Tech. Ltd. v. Maxell, Ltd.*, IPR2021-01442, Paper 16, (PTAB Apr. 12, 2022). Moreover, the Office materially erred by not considering *Wright* and

Daswani because *Wright* and *Daswani* disclose all limitations of the claims as described above.

VII. CONCLUSION

Inter partes review of claims 1-24 of U.S. Patent No. 8,850,517 is requested based on the grounds described above.

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

A. REAL PARTY IN INTEREST:

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

B. RELATED MATTERS:

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

The '517 patent is at issue in the following matters:

- *In re: Taasera Licensing LLC, Patent Litigation*, 2:22-md-03042-JRG (E.D. Tex.), which was consolidated by the Judicial Panel on Multidistrict Litigation on August 3, 2022;
- *Taasera Licensing LLC v. Palo Alto Networks, Inc.*, Case No. 2:23-cv-00113 (E.D. Tex.);
- *Taasera Licensing LLC v. Trend Micro Inc.*, Case No. 2:21-cv-00441-JRG (E.D. Tex.);
- *Trend Micro Inc. v. Taasera Licensing LLC*, Case No. 2:22-cv-00303-JRG (E.D. Tex.);

- *Palo Alto Networks, Inc. v. Taasera Licensing LLC*, Case No. 2:22-cv-00314-JRG (E.D. Tex.);
- *Taasera Licensing LLC v. Musarubra US LLC, d/b/a Trellix*, Case No. 2:22-cv-00427-JRG (E.D. Tex.);
- *Taasera Licensing LLC v. CrowdStrike, Inc., et al.*, Case No. 2:22-cv-00468-JRG (E.D. Tex.);
- *Palo Alto Networks, Inc. v. Taasera Licensing LLC, Inter Partes Review* No. IPR2023-00704; and
- *Trend Micro, Inc. v. Taasera Licensing LLC, Inter Partes Review* No. IPR2023-00798.

C. LEAD AND BACKUP COUNSEL:

Pursuant to 37 C.F.R. § 42.8(b)(3), Petitioner identifies lead and back-up counsel as follows:

Lead Counsel	Back-Up Counsel
Adam P. Seitz (Reg. No. 52,206) Adam.Seitz@eriseip.com PTAB@eriseip.com Postal and Hand-Delivery Address: ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601	Paul R. Hart (Reg. No. 59,646) Paul.Hart@eriseip.com Postal and Hand-Delivery Address: ERISE IP, P.A. 717 17th St., Suite 1400 Denver, Colorado 80202 Telephone: (913) 777-5600 Fax: (913) 777-5601
	Hunter A. Horton, <i>pro hac vice</i> Hunter.Horton@eriseip.com

	Postal and Hand-Delivery Address: ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601
--	---

D. SERVICE INFORMATION:

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

E. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-6159.

Date: October 19, 2023

Respectfully submitted,
ERISE IP, P.A.

BY: /s/ Adam P. Seitz
Adam P. Seitz, Reg. No. 52,206
Paul R. Hart, Reg. No. 59,646
Hunter A. Horton, *pro hac vice*

COUNSEL FOR PETITIONER

CERTIFICATION OF WORD COUNT

The undersigned certifies pursuant to 37 C.F.R. § 42.24 that the foregoing Petition for *Inter Partes* Review, excluding any table of contents, mandatory notices under 37 C.F.R. § 42.8, certificates of service or word count, or appendix of exhibits, contains 13,997 words according to the word-processing program used to prepare this document (Microsoft Word).

Date: October 19, 2023

Respectfully submitted,
ERISE IP, P.A.

BY: /s/ Adam P. Seitz

Adam P. Seitz, Reg. No. 52,206

Paul R. Hart, Reg. No. 59,646

Hunter A. Horton, *pro hac vice*

COUNSEL FOR PETITIONER

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

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105, the undersigned certifies that on October 19, 2023, a complete and entire copy of this Petition for *Inter Partes* Review including exhibits was provided via Federal Express to the Patent Owner by serving the correspondence address of record for the '517 Patent as listed on PAIR:

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

Further, a courtesy copy of this Petition for *Inter Partes* Review was sent via electronic mail to Petitioner's and Patent Owner's counsel in IPR2023-00798:

Robert M. Hansen (rhansen@marburylaw.com)
Scott B. Amankwatia (samankwatia@marburylaw.com)
Matthew Anderson (manderson@marburylaw.com)
pat-docketing@marburylaw.com
Peter Lambrianakos (plambrianakos@fabricantllp.com)
Vincent J. Rubino, III (vrubino@fabricantllp.com)
Alfred R. Fabricant (ffabricant@fabricantllp.com)
Joseph M. Mercadante (jmercadante@fabricantllp.com)
PTAB@fabricantllp.com

Date: October 19, 2023

Respectfully submitted,
ERISE IP, P.A.

BY: /s/ Adam P. Seitz

Adam P. Seitz, Reg. No. 52,206

Paul R. Hart, Reg. No. 59,646

Hunter A. Horton, *pro hac vice*

COUNSEL FOR PETITIONER