

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TREND MICRO, INC.
Petitioner

v.

WEBROOT INC.

Case No. IPR-2023-00655
Patent No. 8,418,250

**PETITION FOR INTER PARTES REVIEW
OF U.S. PATENT NO. 8,418,250**

Filed on behalf of Trend Micro, Inc.
By: Robert M. Hansen
Scott B. Amankwatia
Matthew J. Anderson
The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191

Tel: (703) 391-2900
Fax: (703) 391-2901

TABLE OF CONTENTS

I. Mandatory Notices (37 C.F.R. § 42.8(a)(1))	16
PAYMENT OF FEES.....	18
II. Grounds for Standing (37 C.F.R. § 42.104(a)).....	18
III. Identification of Challenge (37 C.F.R. § 42.104(b)).....	18
A. Citation of Prior Art.....	18
B. Statutory Grounds for Challenge.....	20
IV. The '250 Patent.....	21
A. Claim of Priority of the '250 Patent	21
B. Specification of the '250 Patent	21
C. Summary of the Prosecution History	23
D. Level of Ordinary Skill in the Art	26
E. Claim Construction.....	26
V. Grounds of Rejection.....	27
A. Ground 1: Claims 1-5, 12-17, and 24 are unpatentable under 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin and Mankins.	28
1. Claim 1	28
1[a] A method of classifying a computer object as malware, the method comprising:	28
1[b] at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behavior of the object running on one or more remote computers;	29
1[c] determining in the base computer whether the data about the computer object received from the plural computers indicates that the computer object is malware;	30

1[d]	classifying the computer object as malware when the data indicates that the computer object is malware;.....	31
1[e]	when the determining does not indicate that the computer object is malware, initially classifying the computer object as not malware;....	31
1[f]	automatically generating a mask for the computer object that defines acceptable behavior for the computer object, wherein the mask is generated in accordance with normal behavior of the object determined from said received data;	32
1[g]	running said object on at least one of the remote computers;.....	34
1[h]	automatically monitoring operation of the object on the at least one of the remote computers;	34
1[i]	allowing the computer object to continue to run when behavior of the computer object is permitted by the mask;	35
1[j]	disallowing the computer object to run when the actual monitored behavior of the computer object extends beyond that permitted by the mask; and, ...	35
1[k]	reclassifying the computer object as malware when the actual monitored behavior extends beyond that permitted by the mask.	36
2.	Claim 2	38
	A method according to claim 1, wherein the data about the computer object that is sent from the plural remote computers to the base computer includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.	38

3. Claim 3	40
A method according to claim 1, wherein determining identifies relationships between the object and other objects.....	40
4. Claim 4	41
A method according to claim 3, wherein if at least one other object to which said object is related is classed as malware, then classifying said object as malware.	41
5. Claim 5	42
A method according to claim 3, wherein said other objects include the object or similar objects stored on at least some of the remote computers.	42
6. Claim 12	43
A method according to claim 1, comprising if the monitored behavior of the object running on the remote computer exhibits non-permitted behavior that extends beyond that permitted by the mask, comparing at the base computer the non-permitted behavior with the behavior of other objects and if said behavior is known to be not malicious for said other objects, allowing said behavior for the object and modifying the mask to allow that behavior for that object in future.....	43
7. Claim 13	45
13[a] Apparatus for classifying a computer object as malware, the apparatus comprising:	45
13[b] a base computer constructed and arranged to receive data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behavior of the object running on one or more remote computers;	46
13[c] the base computer being constructed and arranged to determine whether the data about the computer object received from said plural computers indicates that the computer object is malware; and, ...	46

13[d]	the base computer being constructed and arranged to classify the computer object as malware when the base computer determines that the data indicates that the computer object is malware, ...	46
13[e]	wherein the base computer is constructed and arranged to automatically generate a mask for an object that is initially classed not as malware, said mask defining acceptable behavior for the object built in accordance with normal behavior of the object determined from said received data, ...	46
13[f]	wherein operation of the object is automatically monitored when the object is running on at least one of the remote computers, the object being allowed to continue to run when behavior of the object is permitted by the mask, the object not being permitted to run when the actual monitored behavior of the object extends beyond that permitted by the mask, and the object is reclassified as malware when the actual monitored behavior extends beyond that permitted by the mask.....	47
8.	Claim 14	48
	Apparatus according to claim 13, the data includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.....	48
9.	Claim 15	48
	Apparatus according to claim 13, wherein the base computer is constructed and arranged so that the determining identifies relationships between the object and other objects.....	48
10.	Claim 16	49

Apparatus according to claim 15, the base computer is constructed and arranged so that if at least one other object to which said object is related is classed as malware, then said object is classified as malware.49

11. Claim 17 49

Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include the object or similar objects stored on at least sonic [sic] of said remote computers...49

12. Claim 24 50

A method according to claim 13, wherein, if the monitored behavior of the object running on the remote computer exhibits non-permitted behavior that extends beyond that permitted by the mask, the base computer is arranged to compare the non-permitted behavior with the behavior of other objects and if said behavior is known to be not malicious for said other objects the base computer is arranged to allow said behavior for the object and to modify the mask to allow that behavior for that object in future. 50

B. Ground 2: Claims 6 and 18 are unpatentable under 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin, Mankins, and Arnold. 50

1. Claim 6 52

A method according to claim 3, wherein said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object. 52

2. Claim 18 53

Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object. 53

C. Ground 3: Claims 7-11, 19-23, and 25-30 are unpatentable under pre-AIA 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin (EX1003), Mankins (EX1004), Kirby (EX1006), and Carpentier (EX1013). 53

1. Claim 7	55
A method according to claim 1, wherein the data is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on the respective remote computers.	
2. Claim 8	57
A method according to claim 7, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.	
3. Claim 9	57
A method according to claim 7, wherein the key has at least one component that represents data about said object.....	
4. Claim 10	58
A method according to claim 9, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.....	
5. Claim 11	59
A method according to claim 7, wherein the key has at least one component that represents the physical size of the object.....	
6. Claim 19	59
Apparatus according to claim 13, wherein the base computer is constructed and arranged so as to be able to process data that is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on said respective remote computers.....	
7. Claim 20	60

Apparatus according to claim 19, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.	60
8. Claim 21	60
Apparatus according to claim 19, wherein the key has at least one component that represents data about said object.....	60
9. Claim 22	61
Apparatus according to claim 21, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.....	61
10. Claim 23	61
Apparatus according to claim 19, wherein the key has at least one component that represents the physical size of the object.....	61
11. Claim 25	62
25[a] A method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising:	62
25[b] providing from a remote computer to a base computer data about a computer object that is stored on the remote computer;.....	63
25[c] the data including information about the behavior of the object running on one or more remote computers and including one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the	

creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers;	63
25[d] the data being sent in the form of key that is obtained by a hashing process carried out in respect of the object on the remote computer;	64
25[e] automatically generating a mask for the computer object, the mask defining acceptable behavior for the object built in accordance with normal behavior of the object determined from said received data;	64
25[f] executing the computer object at the remote computer;	65
25[g] automatically monitoring operation of the computer object at the remote computer compared to the behavior permitted by the mask;	66
25[h] allowing the object to continue to run when behavior of the computer object is permitted by the mask;	66
25[i] disallowing the object to run when the actual monitored behavior of the object extends beyond that permitted by the mask; and, ...	66
25[j] reclassifying the object as malware when the actual monitored behavior extends beyond that permitted by the mask, wherein the base computer is arranged to receive information about the behavior of the object running on one or more remote computers, wherein the step of generating a mask for that object comprises building the mask with the base computer in accordance with normal behavior of the object determined from said received information.	66
12. Claim 26	67
A method according to claim 25, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.	67

13. Claim 27	68
A method according to claim 25, wherein the key has at least one component that represents data about said object.....	68
14. Claim 28	68
A method according to claim 27, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.....	68
15. Claim 29	69
A method according to claim 25, wherein the key has at least one component that represents the physical size of the object.....	69
16. Claim 30	69
A non-transitory storage medium storing a computer program comprising program instructions for causing a computer to perform the method of claim 25.....	69
D. Ground 4: Claims 1-5, 12-17, and 24 are unpatentable under 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin and Inoue.	70
1. Claim 1	70
1[a] See claim element in Ground 1	70
1[b] See claim element in Ground 1	70
1[c] See claim element in Ground 1	71
1[d] See claim element in Ground 1	71
1[e] See claim element in Ground 1	71

1[f] automatically generating a mask for the computer object that defines acceptable behavior for the computer object, wherein the mask is generated in accordance with normal behavior of the object determined from said received data;	71
1[g] running said object on at least one of the remote computers;.....	73
1[h] automatically monitoring operation of the object on the at least one of the remote computers;	74
1[i] allowing the computer object to continue to run when behavior of the computer object is permitted by the mask;	74
1[j] disallowing the computer object to run when the actual monitored behavior of the computer object extends beyond that permitted by the mask; and, ...	75
1[k] reclassifying the computer object as malware when the actual monitored behavior extends beyond that permitted by the mask.	75
2. Claim 2	77
A method according to claim 1, wherein the data about the computer object that is sent from the plural remote computers to the base computer includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.	77
3. Claim 3	77
A method according to claim 1, wherein determining identifies relationships between the object and other objects.....	77
4. Claim 4	78

A method according to claim 3, wherein if at least one other object to which said object is related is classed as malware, then classifying said object as malware.	78
5. Claim 5	78
A method according to claim 3, wherein said other objects include the object or similar objects stored on at least some of the remote computers.	78
6. Claim 12	79
A method according to claim 1, comprising if the monitored behavior of the object running on the remote computer exhibits non-permitted behavior that extends beyond that permitted by the mask, comparing at the base computer the non-permitted behavior with the behavior of other objects and if said behavior is known to be not malicious for said other objects, allowing said behavior for the object and modifying the mask to allow that behavior for that object in future.....	79
7. Claim 13	80
a. Apparatus for classifying a computer object as malware, the apparatus comprising:	80
b. a base computer constructed and arranged to receive data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behavior of the object running on one or more remote computers;	80
c. the base computer being constructed and arranged to determine whether the data about the computer object received from said plural computers indicates that the computer object is malware; and, ...	80
d. the base computer being constructed and arranged to classify the computer object as malware when the base computer determines that the data indicates that the computer object is malware, ...	80

- e. wherein the base computer is constructed and arranged to automatically generate a mask for an object that is initially classed not as malware, said mask defining acceptable behavior for the object built in accordance with normal behavior of the object determined from said received data, ...81
- f. wherein operation of the object is automatically monitored when the object is running on at least one of the remote computers, the object being allowed to continue to run when behavior of the object is permitted by the mask, the object not being permitted to run when the actual monitored behavior of the object extends beyond that permitted by the mask, and the object is reclassified as malware when the actual monitored behavior extends beyond that permitted by the mask.....81
8. Claim 14 82
- Apparatus according to claim 13, the data includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.....82
9. Claim 15 82
- Apparatus according to claim 13, wherein the base computer is constructed and arranged so that the determining identifies relationships between the object and other objects.....82
10. Claim 16 83
- Apparatus according to claim 15, the base computer is constructed and arranged so that if at least one other object to which said object is related is classed as malware, then said object is classified as malware.83
11. Claim 17 83

Apparatus according to claim 15, wherein the base computer is constructed and arranged wherein the base computer is constructed and arranged so that said other objects include the object or similar objects stored on at least sonic [sic] of said remote computers.....83

12. Claim 24 84

A method according to claim 13, wherein, if the monitored behavior of the object running on the remote computer exhibits non-permitted behavior that extends beyond that permitted by the mask, the base computer is arranged to compare the non-permitted behavior with the behavior of other objects and if said behavior is known to be not malicious for said other objects the base computer is arranged to allow said behavior for the object and to modify the mask to allow that behavior for that object in future. 84

VI. NO BASIS FOR DISCRETIONARY DENIAL.....84

VII. CONCLUSION 87

TABLE OF AUTHORITIES

Cases

<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH</i> , IPR2019-01469, Paper 6 (Feb. 13, 2020)	85
<i>Cellco Partnership v. Huawei Device Co.</i> , IPR2020-01117, Paper 10 (Feb. 3, 2021)	86
<i>General Electric v. Vestas</i> , IPR2018-00928, Paper 9, *16-17 (PTAB Nov. 5, 2018)	27
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966)	27
<i>KSR Int’l Co. v. Teleflex, Inc.</i> , 550 U.S. 398 (2007)	27
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	26
<i>Titanium Metals Corp. of Am. v. Banner</i> , 778 F.2d 775,782 (Fed. Cir. 1985)	39, 58
<i>Western Digital Corporation v. Spex Technologies, Inc.</i> , IPR2018-00084, Paper 14, *11 (PTAB Apr. 25, 2018).....	27

Statutes

35 U.S.C. § 102(a)/(b)(pre-AIA)	18, 19
35 U.S.C. § 102(e)(pre-AIA)	19
35 U.S.C. § 103 (a)(pre-AIA).....	passim

Rules

37 C.F.R. § 42.10(a).....	17
37 C.F.R. § 42.10(b)	17
37 C.F.R. § 42.104(a).....	18
37 C.F.R. § 42.8(a)(1)	16
37 C.F.R. § 42.8(b)(2).....	16
37 C.F.R. § 42.8(b)(3).....	17
37 C.F.R. § 42.100(b)	26

EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 8,418,250 to Morris <i>et al.</i>
1002	Expert Declaration of Dr. Seth Nielson
1003	U.S. Published Patent Application No. 2004/0111632 to Halperin
1004	U.S. Patent No. 7,900,194 to Mankins
1005	Inoue, et al., “Anomaly Intrusion Detection in Dynamic Execution Environments”, New Security Paradigms Workshop ’02, September 23-26, 2002.
1006	U.S. Patent No. 7,694,150 to Kirby
1007	Prosecution History of the ’250 patent
1008	U.S. Patent No. 5,440,723 to Arnold
1009	Fred Cohen, “Computer Viruses Theory and Experiments”
1010	Joseph A. Bank, “Java Security”
1011	<i>Curriculum Vitae</i> of A.L. Seth Nielson, Ph.D.
1012	U.S. Patent Publication 2003/0177394 to Dozortsev
1013	U.S. Published Patent Application No. 2004/0068652 to Carpentier <i>et al.</i>

Trend Micro, Inc. (“Petitioner”) petitions for *inter partes* review of claims 1-30 of U.S. Patent No. 8,418,250 to Morris *et al.*, entitled “METHODS AND APPARATUS FOR DEALING WITH MALWARE” (“the ’250 patent,” EX1001), which according to PTO records, is assigned to Webroot Inc.

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

REAL PARTY IN INTEREST:

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

RELATED MATTERS:

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

The ’250 patent is also at issue in the following district court litigations:

- *Webroot, Inc. and Open Text, Inc. v. Trend Micro Inc.*, Case No. 6:22-cv-00239 (W.D. Tex.), which was originally filed on March 4, 2022.
- *Webroot, Inc. and Open Text, Inc. v. Sophos LTD.*, Case No. 6:22-cv-00240 (W.D. Tex.), which was originally filed on March 4, 2022.

- *Webroot, Inc. and Open Text, Inc. v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.*, Case No. 6:22-cv-00241 (W.D. Tex.), which was originally filed on March 4, 2022.
- *Webroot, Inc. and Open Text, Inc. v. AO Kaspersky Lab*, Case No. 6:22-cv-00243 (W.D. Tex.), which was originally filed on March 4, 2022.

LEAD AND BACKUP COUNSEL:

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

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

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

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

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

SERVICE INFORMATION:

Petitioner provides the following service information for designated counsel:

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

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

rhansen@marburylaw.com;

samankwatia@marburylaw.com;
manderson@marburylaw.com; and
pat-docketing@marburylaw.com.

PAYMENT OF FEES

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

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

Petitioner certifies that the '250 patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

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

A. Citation of Prior Art

Petitioner cites the following prior art references:

U.S. Published Patent Application No. 2004/0111632A1 to Halperin

("Halperin," EX1003) is prior art under pre-AIA § 102(a)/(b)(pre-AIA) because it published June 10, 2004, more than a year before the earliest priority date of the '250 patent. Halperin was not submitted to or considered by the Examiner of the '250 patent.

U.S. Patent No. 7,900,194 to Mankins (“Mankins,” EX1004) is prior art under pre-AIA § 102(e)(pre-AIA) because Mankins issued from a non-provisional filed on March 23, 2005, before the earliest priority date of the ’250 patent, and from a provisional application filed on March 25, 2004. Mankins was not submitted to or considered by the Examiner of the ’250 patent.

U.S. Patent No. 5,440,723 to Arnold (“Arnold,” EX1008) is prior art under pre-AIA § 102(a)/(b)(pre-AIA) because it published August 8, 1995, more than one year before the earliest priority date of the ’250 patent. Arnold was not submitted to or considered by the Examiner of the ’250 patent.

Inoue *et al.*, Anomaly Intrusion Detection in Dynamic Execution

Environments”, New Security Paradigms Workshop ’02, September 23-26, 2002 (“Inoue,” EX1005) is prior art under pre-AIA § 102(a)/(b)(pre-AIA) because it published no later than September 26, 2002, more than a year before the earliest priority date of the ’250 patent. Inoue was not disclosed to or considered by the Examiner of the ’250 patent.

U.S. Patent 7,694,150 to Kirby (“Kirby,” EX1006) is prior art under pre-AIA § 102(e)(pre-AIA) because Kirby issued from a non-provisional filed on June 22, 2004, more than a year before the earliest priority date of the ’250 patent. Kirby was not disclosed to or considered by the Examiner of the ’250 patent.

U.S. Patent Publication 2004/0068652 to Carpentier, et al. (“Carpentier,” EX1013) is prior art under pre-AIA § 102(a)/(b)(pre-AIA)) because it published April 8, 2004, more than a year before the earliest priority date of the ’250 patent. Carpentier was not disclosed to or considered by the Examiner of the ’250 patent.

B. Statutory Grounds for Challenge

Petitioner requests review of claims 1-30 (“Challenged Claims”) on the following grounds:

GROUND 1: Claims 1-5, 12-17, and 24 are unpatentable under pre-AIA 35 U.S.C. § 103(a) as obvious in view of Halperin (EX1003) and Mankins (EX1004).

GROUND 2: Claims 6 and 18 are unpatentable under pre-AIA 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin (EX1003), Mankins (EX1004), and Arnold (EX1008).

GROUND 3: Claims 7-11, 19-23, and 25-30 are unpatentable under pre-AIA 35 U.S.C. § 103(a) as obvious in view of Halperin (EX1003), Mankins (EX1004), Kirby (EX1006), and Carpentier (EX1013).

GROUND 4: Claims 1-5, 12-17, and 24 is unpatentable under pre-AIA 35 U.S.C. § 103(a) as obvious in view of Halperin (EX1003) and Inoue (EX1005).

IV. The '250 Patent

A. Claim of Priority of the '250 Patent

The '250 patent, entitled “Methods and Apparatus for Dealing With Malware,” issued April 9, 2013 from U.S. Application No. 11/477,807, filed June 30, 2006, claiming priority to GB 0513375.6, filed June 30, 2005. EX1001, Cover.¹ Therefore, the '250 patent should be examined under pre-AIA statutes.

B. Specification of the '250 Patent

The '250 patent describes methods and an apparatus for classifying a computer object (e.g., computer executable code) as malware based on data about the computer object received from a plurality of remote computers on which the object or similar objects are stored. EX1001, Front Cover, Abstract. Claimed methods compare computer objects or their effects on different remote computers to determine whether a computer object should be classified as good or malware. *Id.*, 3:9-12.

Claims 1-12 recite computer-implemented methods that include: “at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including

¹ Petitioner does not concede that all '250 patent claims are entitled to a June 30, 2005 priority date because the GB 0513375.6 application does not support several dependent claims.

information about the behavior of the object running on one or more remote computers; determining in the base computer whether the data about the computer object received from the plural computers indicates that the computer object is malware; classifying the computer object as malware when the data indicates that the computer object is malware; when the determining does not indicate that the computer object is malware, initially classifying the computer object as not malware; automatically generating a mask for the computer object that defines acceptable behavior for the computer object, wherein the mask is generated in accordance with normal behavior of the object determined from said received data; running said object on at least one of the remote computers; automatically monitoring operation of the object on the at least one of the remote computers; allowing the computer object to continue to run when behavior of the computer object is permitted by the mask; disallowing the computer object to run when the actual monitored behavior of the computer object extends beyond that permitted by the mask; and, reclassifying the computer object as malware when the actual monitored behavior extends beyond that permitted by the mask.” EX1001, 20:19-50.

Dependent claim 2 recites types of data about the computer object that the plural remote computers send to the base computer. *Id.*, 20:51-62. Dependent claims 3-6 recite further limitations regarding the determining step. *Id.*, 20:63–

21:6. Dependent claims 7-11 recite that data regarding computer objects are sent in the form “keys” obtained by a hashing process applied on the object or objects by remote computers. *Id.*, 21:7-27. Dependent claim 12 limits responses to non-permitted behavior. *Id.*, 21:27-35.

Claims 13-24 are apparatus claims that duplicate the subject matter of method claims 1-12. Claims 25-29 recite methods for remote computers to provide data to the base computer to perform analysis as in claim 1, with claims 26-29 corresponding to claims 8-11. Claim 30 depends from claim 25 reciting a non-transitory medium storing instructions for the method of claim 25.

C. Summary of the Prosecution History

After a Restriction Requirement, prosecution of the 11/477,807 application began with a non-final Office Action on May 4, 2010, continued through two Non-final Office Actions, two Final Office Actions, three sets of claim amendments, three Requests for Continued Examination, and concluded with the Notice of Allowance that issued on December 13, 2012. EX1007, pp.39–525.

In a September 28, 2010 Final Office Action, claims 1-9, 12-21, 24-27, and 30 were rejected as unpatentable over U.S. Patent Publication 2003/0177394 to Dozortsev (EX1012) in view of U.S. Patent Publication 2001/0052014 to Sheymov. Claims 10, 11, 22, 23, 28, and 29 were rejected as unpatentable over

Dozortsev and Sheymov in view of U.S. Patent Publication 2005/0021994 to Barton. EX1007, pp.451-460.

In a response filed January 28, 2011, Applicants amended the independent claims to include, *inter alia*: “automatically monitoring operation of the computer object on the at least one of the remote computers; allowing the computer object to continue to run when behavior of the computer object is permitted by the mask; disallowing the computer object to run when the actual monitored behavior of the computer object extends beyond that permitted by the mask.” EX1007, pp.433 (underlines showing added limitations). Applicants argued that the amended claims differed from Dozortsev because that reference related to “an isolated sandbox” “on a central computer” that was “monitored by a human operator,” all of which were cited as points of differentiation from Dozortsev. EX1007, pp.332-335. Thus, Applicants argued Dozortsev is incapable of “automatically” classifying the object, “while the object is running on the remote computer.” *Id.*

A February 15, 2011 Non-final Office Action rejected all pending claims (1-11, 13-23, 25-30, and 48-51). Claims 1-10, 13-22, 25-28 and 30 were rejected as unpatentable over U.S. Patent Publication 2003/0177394 to Dozortsev in view of U.S. Patent Publication 2002/0099952 to Lambert. Claims 11, 23 and 29 were rejected as unpatentable over Dozortsev and Lambert in view of U.S. Patent Publication 2003/0084323 to Gales. EX1007, pp.307–318.

On June 15, 2011, Applicants responded by amending the independent claims to recite “... at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behavior of the object running on one or more remote computers; ... automatically generating a mask for the computer object that defines acceptable behavior for the computer object, wherein the mask is generated in accordance with normal behavior of the object determined from said received data.” EX1007, pp.295–306 (underlines showing amendments). Applicants argued that Dozortsev and Lambert do not disclose that the data received from remote devices includes information about the behavior of the object and the mask is automatically generated based on normal behavior of the objected determined from the received data including behavior information. *Id.*, pp.284–288.

In a July 25, 2011 Final Office Action, the Examiner agreed in part but replaced Lambert with U.S. Patent Publication 2005/0210035 to Kester. *Id.*, pp.266-272.

On December 23, 2011, Applicants responded by arguing that Kester relied entirely on human administrators to create and update access policies stored in a policy database. Even though behavior data was collected from other computers in Kester, Applicants argued that humans ultimately built the policies based on that

data and that the policies were not automatically generated. Applicants argued that the method recited in the claims automatically generates the mask and uses the mask to automatically monitor operations, asserting: “a human decision is not necessary to generate the mask or monitor operation.” *Id.*, pp.140-144.

Neither the first Notice of Allowance that issued March 21, 2012 nor the second Notice of Allowance that issued December 13, 2012 included a statement of reasons for allowance. EX1007, pp.44,79-80.

D. Level of Ordinary Skill in the Art

As established by the Declaration of Dr. Seth Nielson dated 3/2/2023 (EX1002, ¶81)(CV:EX1011), a person having ordinary skill in the art (PHOSITA) at the time of the earliest claimed priority date for the '250 patent would have had a bachelor's degree in computer science, or a related field, or an equivalent technical degree or equivalent work experience, and an additional one to two years of education or experience in the field of computer security topics such as malware/exploit detection and/or network forensics. More education can supplement practical experience and vice versa. EX1002, ¶81.

E. Claim Construction

In an IPR 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).

In parallel litigation, Webroot argues all claim terms should be given their plain and ordinary meanings. For purposes of this proceeding only, Petitioner proposes the Board adopt Webroot's proposal. *See Western Digital Corporation v. Spex Technologies, Inc.*, IPR2018-00084, Paper 14, *11 (PTAB Apr. 25, 2018) (finding Petition based on claim constructions urged by Patent Owner satisfies the claim construction requirements and Petitioner is not required to express its subjective agreement regarding correctness of the proffered claim construction or take ownership of the construction); *see also General Electric v. Vestas*, IPR2018-00928, Paper 9, *16-17 (PTAB Nov. 5, 2018).²

V. Grounds of Rejection

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

In all grounds Petitioner evaluates the scope and content of the prior art, any differences between the art and the claims, and the knowledge of person of ordinary skill in the art in accordance with *Graham v. John Deere Co.*, 383 U.S. 1 (1966) and *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398 (2007). Petitioner is

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

unaware of any secondary considerations relevant to the '250 patent and a more than adequate written rationale is provided to support the combinations of Halperin, Mankins, Arnold, Kirby, Carpentier, and Inoue under KSR.

A. Ground 1: Claims 1-5, 12-17, and 24 are unpatentable under 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin and Mankins.

A PHOSITA would have been motivated to combine Halperin and Mankins because both are directed to malware containment in which Halperin provides a reactive response while Mankins discloses a proactive approach that together form a strong malware defense. EX1002, ¶107. Further, a PHOSITA would have an expectation of success in making this combination because both references solve the same problem in a complementary manner (based on observing normal behavior) as identified on a per-claim-limitation basis below. EX1002, ¶¶107-109. Further, Mankins offers an improvement to the accuracy-limiting trade off disclosed in Halperin as a choice between false alarms and malware detection. EX1002, ¶¶107-113.

1. Claim 1

1[a] A method of classifying a computer object as malware, the method comprising:

Halperin discloses methods for detecting malicious activity on a computer network, including categorizing behavior of computer processes to identify viruses. EX1002, ¶¶116-123; EX1003, ¶¶[0087]-[0089], Figs.2,7,11. Halperin discloses

containment of suspicious software if it is identified as a virus. *Id.* Accordingly, a PHOSITA would have understood that Halperin discloses classifying a computer object as malware as recited in this preamble. EX1002, ¶¶116-123.

Therefore, Halperin discloses this portion of claim 1. *Id.*

1[b] at a base computer, receiving data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behavior of the object running on one or more remote computers;

Halperin discloses a centralized server (corresponding to “base computer”) that receives data about network and software activity from multiple computers (corresponding to “plural remote computers”) via a network (e.g., Internet). EX1002, ¶¶63(citing-EX1009),126-130; EX1003, ¶¶[0036],[0047],[0087], Figs.5,7. Halperin discloses that infected messages are identified and descriptions of the suspicious behavioral activity from software (corresponding to “data about a computer object from each of plural remote computers”) on these computers are shared across the network including with the centralized server. EX1002, ¶¶126-130; EX1003, ¶¶[0022],[0034],[0047]. The computers use the shared information to correlate parameters that characterize suspicious software to identify, and heighten sensitivity to, similar software and behaviors (corresponding to “a computer object from each of plural remote computers on which the object or similar objects are stored”) on the computers including at the centralized server. EX1002, ¶¶126-130; EX1003, ¶¶[0037]-[0059]. The multiple computers on the

network share data describing suspicious behaviors from software and emails with the centralized server (corresponding to “the data including information about the behavior of the object running on one or more remote computers”). EX1002, ¶¶130-130; EX1003, ¶¶ [0037]-[0059].

Therefore, Halperin discloses this limitation. EX1002, ¶¶130,132.

1[c] determining in the base computer whether the data about the computer object received from the plural computers indicates that the computer object is malware;

Halperin discloses a server analyzing behavior at plural computers to determine if a computer is infected by a virus. EX1002, ¶135; EX1003, ¶¶ [0037]-[0059]. Halperin states:

...server 502 may then correlate the presence of target behavior detected at two or more of computers 500 in order to determine whether the correlated target behavior corresponds to a predefined suspicious behavior pattern of target behavior as an indication that a computer virus may have infected those computers...

EX1003, ¶[0047].

As noted regarding limitation 1[b], Halperin discloses identifying portions of emails and software (corresponding to “computer object”) that are suspicious or malicious. EX1002, ¶¶134-135; EX1003, ¶¶ [0037]-[0059].

Therefore, Halperin discloses this limitation. EX1002, ¶136.

1[d] classifying the computer object as malware when the data indicates that the computer object is malware;

Halperin discloses that, at the server, if the suspicious software meets certain criteria or has certain parameters, the software is quarantined (corresponding to “classifying...that the computer object is malware”). EX1002, ¶¶138-140; EX1003, ¶[0058]. Halperin states:

[S]hould a computer virus send one or more infected messages to valid, non-decoy addresses before sending an infected message to a decoy address, the infected messages to valid, non-decoy addresses that are still held at the server may be "quarantined" at the server and thus prevented...

EX1003, ¶[0058].

Thus, based on data received at the server, infected messages with computer viruses are quarantined, which Halperin describes as an action separate from actions taken for merely suspicious software. EX1002, ¶¶130,139; EX1003, ¶[0058]-[0061].

Therefore, Halperin discloses this limitation. EX1002, ¶¶140-141.

1[e] when the determining does not indicate that the computer object is malware, initially classifying the computer object as not malware;

Halperin discloses identifying suspicious activity and collecting software behavior data before software is identified as a virus or quarantined. EX1002, ¶¶143-145; EX1003, ¶¶[0061],[0090]. Halperin discloses collecting data from multiple computers to analyze patterns in behavior that are used to determine a

probability whether software is malicious. *Id.* Halperin discloses that the system defines the software as not malicious if the probability does not meet a threshold probability. *Id.* Halperin discloses that as more data is collected on the software, the probability or threshold may change so the software may later be defined as malicious. *Id.*

Therefore, Halperin discloses this limitation. EX1002, ¶146.

1[f] automatically generating a mask for the computer object that defines acceptable behavior for the computer object, wherein the mask is generated in accordance with normal behavior of the object determined from said received data;

As described regarding claim 1[b], Halperin discloses a system collecting behavioral information in a learning database. EX1002, ¶¶125-132; EX1003, ¶¶[0022],[0034],[0047]. This behavioral information is analyzed to identify normal and suspicious software behavior (corresponding to “normal behavior of the object determined from said received data”). EX1002, ¶148; EX1003, ¶¶[0071],[0076]. This behavioral analysis is stored and used to filter further information received at the server (corresponding to “automatically generating ...acceptable behavior for the computer object”). EX1002, ¶¶148-149; EX1003, ¶¶[0086]-[0088].

Mankins discloses automatically generating a statistical data structure called a Bloom filter (corresponding to “a mask”) that summarizes system call behavior of software as a series of hashes. EX1002, ¶150; EX1004, 3:15-30,7:25-33,8:8-55. Mankins teaches that the intrusion detection system defines this data structure

object during a training period based on normal behavior baseline (corresponding to “a mask ...that defines acceptable behavior for the computer object”) for system calls (corresponding to “generated in accordance with normal behavior of the object determined from said received data”). EX1002, ¶¶150-154; EX1004, 3:30-45. Mankins teaches:

“Normal' behavior may be defined differently but generally refers to observable and acceptable behavior characteristics expected of networks when not under attack.

EX1004, 1:41-43.

The intrusion detection system then uses a Bloom filter to generate a Bloom filter data object containing data representative of the normal system-call behavior.

EX1004, 3:40-45.

During operation of software on a remote computer of the network, the Bloom filter is used to check system calls to flag abnormal behavior. EX1002, ¶¶150-154; EX1004, 3:45-60,5:60-65.

Both the behavioral analysis disclosed in Halperin and the Bloom filter for behavior disclosed in Mankins are automatically generated. EX1002, ¶¶148,150; EX1004, 8:8-13; EX1003, ¶¶[0061],[0081],[0086]-[0088]. It would have been obvious for a PHOSITA to generate the Bloom filter of normal behavior disclosed in Mankins on the central server of Halperin after receiving information about a software's system calls from the remote computers as disclosed by Halperin.

EX1002, ¶¶148-156. This combination would have been obvious to efficiently implement the distributed intrusion detection system proposed by Mankins. *Id.* A PHOSITA would have expected this combination to be successful because the data structure disclosed in Mankins can be assembled anywhere with the system call information as an input. *Id.* Further, a PHOSITA would have been motivated to use Mankins' Bloom filter because it obviates a risky tradeoff between false alarms and virus detection, which Halperin identifies as an issue. EX1002, ¶¶148-158; EX1004, 8:38-41; EX1003, ¶[0090].

Therefore, Halperin in view of Mankins discloses this limitation. EX1002, ¶159.

1[g] running said object on at least one of the remote computers;

Halperin discloses allowing software to run on remote computers until certain behavioral thresholds are reached. EX1002, ¶¶161-163; EX1003, ¶¶[0069]-[0071]. Mankins also teaches this feature. EX1002, ¶164.

Therefore, Halperin and Mankins disclose this limitation. EX1002, ¶155.

1[h] automatically monitoring operation of the object on the at least one of the remote computers;

Halperin discloses allowing software (corresponding to “the object”) to run on remote computers and monitoring the execution of the software until certain behavioral thresholds are reached, which renders this limitation obvious. EX1002, ¶166; EX1003, ¶¶[0069]-[0071]. Further, Mankins discloses similar compatible

features including automatically monitoring software using the Bloom filter.

EX1004, 8:8-15; EX1002, ¶¶151-153,166-167.

Therefore, Halperin in view of Mankins discloses this limitation. *Id.*

1[i] allowing the computer object to continue to run when behavior of the computer object is permitted by the mask;

Halperin discloses allowing software (corresponding to “the object”) to run on remote computers when certain suspicious behaviors are below a threshold. EX1002, ¶¶170-172; EX1003, ¶¶[0087],[0090]. Halperin discloses that allowing the software to run may occur in a learning phase. *Id.* Together, Halperin and Mankins teach monitoring software as it runs in its normal environment (subject to intrusion) using the Bloom filter to check whether software behavior is normal, and if so, the Bloom filter allows it. EX1002, ¶¶170-172; EX1004, 3:45-67.

Therefore, Halperin in view of Mankins discloses this limitation. EX1002, ¶173.

1[j] disallowing the computer object to run when the actual monitored behavior of the computer object extends beyond that permitted by the mask; and,

Halperin teaches both containment and quarantine of software that has been determined to be malicious based on the received data from computers. EX1002, ¶175; EX1003, ¶¶[0075]-[0076],[0079],[0090]. Halperin discloses that detected behavior for containment includes operations that differ from normal correlated behavior or matches a target behavior profile. EX1002, ¶175; EX1003, ¶¶[0038]-

[0047]. Together, Halperin and Mankins teach issuing an alert or taking other actions when a monitored behavior or system call does not match the Bloom filter data structure. EX1002, ¶176; EX1004, 10:5-15. A PHOSITA would understand the alert from the Bloom filter can be used to trigger other actions including halting the behavior or performing containment operations as disclosed in Halperin. EX1002, ¶¶175-177; EX1003, ¶¶[0038]-[0047].

Therefore, Halperin and Mankins disclose this limitation. EX1002, ¶178.

1[k] reclassifying the computer object as malware when the actual monitored behavior extends beyond that permitted by the mask.

Halperin discloses collecting data on software behavior and changing a threat probability or threshold for an alert (corresponding to “reclassifying the computer object as malware”) as more data is collected on the software so that later the software can be defined as malicious and contained. EX1002, ¶¶132-135,180; EX1003, ¶¶[0061],[0090]. As described regarding limitation 1[e], behavior may initially be classified as not malware despite being suspicious. EX1002, ¶¶132-135,180; EX1003, ¶¶[0047],[0061],[0090]. Halperin discloses that if the suspicious behavior spreads to groups of computers that have few connections or little normal correlated activity, the activity may be identified as malicious (corresponding to “when the actual monitored behavior extends beyond that permitted”). EX1002, ¶180; EX1003, ¶¶[0037],[0063]-[0064],[0078]-[0081].

Halperin states:

Activity that cannot be confidently attributed to a virus when observed on one computer can be clearly identified as such when observed on several computers in a network.

EX1003, ¶[0037].

The present invention may be employed to identify suspicious activity occurring in multiple groups simultaneously. For example, if suspicious behavior is detected at a computing device, and similar suspicious behavior is also detected in various groups to which the computing device belongs, virus containment actions may be taken in each of the groups.

EX1003, ¶[0079].

Thus, Halperin teaches reclassifying suspicious behavior as malicious given various abnormal or unacceptable behaviors (e.g., distant spreading, communication spikes, etc.). EX1002, ¶180; EX1003, ¶¶[0037],[0063]-[0064],[0078]-[0081].

Analogously, Mankins discloses monitoring software as it runs to ensure that its behavior does not stray outside of an authorized set of approved operations, as described regarding limitation 1[j]. EX1002, ¶¶150-151,181-182; EX1004, 3:45-67,10:5-15. Mankins discloses that recorded behavior is formed into a hash-based data structure (corresponding to “a mask”) of acceptable behavior for the software. EX1002, ¶¶150-151,182-183; EX1004, 3:15-45.

A PHOSITA would have found it obvious to combine data structure and comparison methods of Mankins with the behavioral data collection and analysis of Halperin, so that suspicious behavior would be reclassified as malicious when the behavior extends beyond the mask, because the two approaches for gathering information on software behavior are compatible and complementary. EX1002, ¶¶180-184.

Therefore, Halperin and Mankins disclose this limitation. EX1002, ¶185.

For the foregoing reasons, a PHOSITA would have found claim 1 to be obvious in view of Halperin and Mankins. EX1002, ¶114. Accordingly, claim 1 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins.

2. Claim 2

A method according to claim 1, wherein the data about the computer object that is sent from the plural remote computers to the base computer includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

As described regarding claim 1, Halperin discloses collecting information at the server from computers on the network (corresponding to “the data about the computer object that is sent from the plural remote computers to the base

computer”). The collected information includes target behaviors, such as modifying operating system files, the timing of deleting files (corresponding to “creation and modification dates of the object”), and sending files, attachments (corresponding to “executable instructions contained within or constituted by the object”), and messages with suspicious initiations (corresponding to “events initiated by or involving the object when the object is created, configured or runs on the respective remote computers”). EX1002, ¶¶187-188; EX1003, ¶¶[0038]-[0047]. Accordingly, Halperin teaches at least three of the alternative data limitations listed in claim 2, 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.”) .

Therefore, a PHOSITA would have found the claim 2 limitations obvious in view of Halperin. EX1002, ¶189. Accordingly, claim 2 should be found invalid under 35 U.S.C. § 103(a) based on Halperin in view of Mankins.

3. Claim 3

A method according to claim 1, wherein determining identifies relationships between the object and other objects.

Halperin discloses a server analyzing target behavior at plural computers and the interaction between software on the computers to determine whether a computer is infected by a virus. EX1002, ¶192; EX1003, ¶¶[0037]-[0059].

Halperin states:

...server 502 may then correlate the presence of target behavior detected at two or more of computers 500 in order to determine whether the correlated target behavior corresponds to a predefined suspicious behavior pattern of target behavior as an indication that a computer virus may have infected those computers...

EX1003, ¶[0047].

Halperin discloses several suspicious behavior patterns across a network of computers that indicate malware infection. EX1002, ¶¶192-194; EX1003, ¶¶[0037]-[0059]. For example, Halperin discloses that “[a] certain percentage of the computers in the network sending messages not initiated via the message GUI in the last 1 minute’ may indicate malware and that “[a] certain percentage of the computers in the network deleting more than 10 files in the last 1 minute” may indicate malware. EX1002, ¶¶192-200; EX1003, ¶¶[0049]-[0050]. In these examples and the patterns identified, Halperin is disclosing potential malware as an actor (“the object”) and other computers, files, or registries as targets (“other

objects”). EX1002, ¶¶193-204; EX1003, ¶¶[0037]-[0059]. A PHOSITA would have recognized that the correlations and pattern recognition disclosed in Halperin identify relationships as recited in claim 3. EX1002, ¶¶194-203. Further, the relationships identified in Halperin parallel those relationships described in the ’250 patent. EX1002, ¶¶193-204; EX1001, 12:60-13:15.

Therefore, a PHOSITA would have found the claim 3 limitation obvious in view of Halperin. EX1002, ¶205. Accordingly, claim 3 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins.

4. Claim 4

A method according to claim 3, wherein if at least one other object to which said object is related is classed as malware, then classifying said object as malware.

Halperin teaches a server identifying malware based on parameters and characteristics in software or emails that indicate malicious actions (corresponding to “said object is related is classed as malware”). EX1002, ¶¶207-210; EX1003, ¶¶[0038]-[0047],[0056]. Further, Halperin teaches using signatures of a set of one or more software actions (corresponding to “if at least one other object to which said object is related”) that can be used to identify a malicious piece of software and contain it (corresponding to “then classifying said object as malware”). EX1002, ¶¶210-212; EX1003, ¶¶[0047],[0073]. The ’250 patent, likewise, describes using signatures to identify related objects. EX1001, 17:40-60.

Therefore, a PHOSITA would have found the claim 4 limitations obvious in view of Halperin. EX1002, ¶213. Accordingly, claim 4 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins.

5. Claim 5

A method according to claim 3, wherein said other objects include the object or similar objects stored on at least some of the remote computers.

As described regarding claim limitation 1[h] and claim 3, Halperin discloses using behavioral signatures to identify viruses on remote computers and that using similar analysis can allow correlation of suspicious activities between groups. EX1002, ¶¶215-217; EX1003, ¶¶[0047],[0073],[0078]. A PHOSITA would have understood that this means the corresponding malware is “stored” on the remote computers. EX1002, ¶216.

Therefore, a PHOSITA would have found the claim 5 limitations obvious in view of Halperin. EX1002, ¶218. Accordingly, claim 5 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins.

6. Claim 12

A method according to claim 1, comprising if the monitored behavior of the object running on the remote computer exhibits non-permitted behavior that extends beyond that permitted by the mask, comparing at the base computer the non-permitted behavior with the behavior of other objects and if said behavior is known to be not malicious for said other objects, allowing said behavior for the object and modifying the mask to allow that behavior for that object in future.

Halperin teaches that for suspicious behavior (corresponding to “non-permitted behavior that extends beyond that permitted by the mask”) the system may delay a decision that it is malicious until more behavior can be observed (corresponding to “if said behavior is known to be not malicious for said other objects”). EX1002, ¶¶220-222; EX1003, ¶¶[0023]-[0028],[0058]-[0059]. Further, Halperin teaches “[a]ctivity that cannot be confidently attributed to a virus when observed on one computer can be clearly identified as such when observed on several computers in a network.” (corresponding to “modifying the mask to allow that behavior for that object in future.”) EX1002, ¶¶222-224; EX1003, ¶¶[0037]. A PHOSITA would have found it obvious to continuously update the set of normal behaviors taught by Halperin. EX1002, ¶225.

Halperin discloses analyzing, at the server, behavior at plural computers to determine whether a computer is infected by a virus (corresponding to “comparing at the base computer”). EX1002, ¶¶124-128,224-225; EX1003, ¶¶[0037]-[0059]. Halperin further states:

... Live and test set responses to system events, such as actual virus detections and false alarms, may be compared to identify algorithm effectiveness. This may be performed retrospectively once a series of system alerts have been identified as either real virus alerts or false alarms. ...

EX1003, ¶[0075].

As noted regarding claim 4, Halperin discloses a server identifying portions of software or emails as malware based on comparisons to the behavior of known malware. Likewise, alerts and possible detections can be compared with behavior of known malware or test sets to identify false positives (corresponding to “comparing at the base computer the non-permitted behavior with the behavior of other objects”). EX1002, ¶¶126-128, 222-230; EX1003, ¶¶[0037]-[0059].

Analogously, Mankins teaches training the Bloom filter (corresponding to “mask”) at the intrusion detection system to encapsulate the normal behavior of the software. EX1002, ¶226-227; EX1004, 3:15-45. A PHOSITA would have understood that this Bloom filter may be updated based on feedback from a test set as described in Halperin above. EX1002, ¶¶231-232. It would have been obvious to a PHOSITA to continuously update the Bloom filter at the intrusion detection system to cover normal behavior of the software if it is determined to be not malicious as described by Halperin. EX1002, ¶¶231.

Therefore, a PHOSITA would have found the claim 12 limitations obvious in view of Halperin and Mankins. EX1002, ¶220. Accordingly, claim 12 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins.

7. Claim 13

13[a] Apparatus for classifying a computer object as malware, the apparatus comprising:

Halperin discloses a system (corresponding to “apparatus”) including a server that collects data describing software behavior from computers on the network and correlates that data to identify malicious software. EX1003, ¶¶[0021],[0092],FIG.1,8; EX1002, ¶¶117-123,236-237.

Therefore, Halperin discloses this portion of claim 13. EX1002, ¶238.

Claim 13 and its dependent claims repeatedly recite a “base computer constructed and arranged to” perform the same or analogous actions as recited in claim 1. As shown above, Halperin discloses a server, which corresponds to a “base computer.” EX1002, ¶227-128,236-237,240. As shown regarding claim 1, Halperin discloses methods performed by computers including the server, such that a PHOSITA would have recognized that the server of Halperin is “constructed and arranged to” perform the actions recited in claims 1 and 13 and their dependent claims. *Id.*

13[b] a base computer constructed and arranged to receive data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behavior of the object running on one or more remote computers;

As shown regarding analogous limitation 1[b], Halperin discloses limitation 13[b]. EX1002, ¶¶126-132,240-241.

13[c] the base computer being constructed and arranged to determine whether the data about the computer object received from said plural computers indicates that the computer object is malware; and,

As shown regarding analogous claim limitation 1[c], Halperin discloses limitation 13[c]. EX1002, ¶¶134-135,243-244.

13[d] the base computer being constructed and arranged to classify the computer object as malware when the base computer determines that the data indicates that the computer object is malware,

As shown regarding analogous claim limitation 1[d], Halperin discloses limitation 13[d]. EX1002, ¶¶138-140,246-247.

13[e] wherein the base computer is constructed and arranged to automatically generate a mask for an object that is initially classed not as malware, said mask defining acceptable behavior for the object built in accordance with normal behavior of the object determined from said received data,

As shown regarding analogous claim limitations 1[e]-[f], Halperin discloses a server collecting behavioral information in a learning database that is analyzed to identify normal software behavior and suspicious behavior (corresponding to “normal behavior of the object determined from said received data”). EX1002, ¶¶143,249; EX1003, ¶¶[0071],[0076]. The behavioral information is collected

during a training period when the monitored software is not considered malware (corresponding to “automatically generate ...for an object that is initially classes not as malware”). EX1002, ¶¶148-149; EX1003, ¶¶[0071],[0076].

As shown regarding analogous claim limitations 1[e]-[f], Mankins discloses the remainder of the limitations associated with the mask. EX1002, ¶¶150-153,250.

13[f] wherein operation of the object is automatically monitored when the object is running on at least one of the remote computers, the object being allowed to continue to run when behavior of the object is permitted by the mask, the object not being permitted to run when the actual monitored behavior of the object extends beyond that permitted by the mask, and the object is reclassified as malware when the actual monitored behavior extends beyond that permitted by the mask.

As described regarding claim limitations 1[g] and 1[h], Halperin discloses the limitation of “wherein operation of the object is automatically monitored when the object is running on at least one of the remote computers.” EX1002, ¶¶160-163,165-167,252.

As described regarding claim limitations 1[i], 1[j], and 1[k], the remainder of this limitation is disclosed by Halperin and Mankins. EX1002, ¶¶169-172,174-177,179-184,252.

For the foregoing reasons, a PHOSITA would have found claim 13 to be obvious in view of the combination of Halperin and Mankins. *Id.* Accordingly, claim 13 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins.

8. Claim 14

Apparatus according to claim 13, the data includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

As shown regarding analogous limitations in claim 2, Halperin discloses the limitations in claim 14. EX1002, ¶¶187-188,255-256. Accordingly, claim 14 should be found invalid under 35 U.S.C. § 103(a) based on Halperin in view of Mankins.

9. Claim 15

Apparatus according to claim 13, wherein the base computer is constructed and arranged so that the determining identifies relationships between the object and other objects.

As shown regarding analogous limitations in claim 3, Halperin discloses the limitations of claim 15. EX1002, ¶¶192-194,258. Accordingly, claim 15 should be found invalid under 35 U.S.C. § 103(a) based on Halperin in view of Mankins.

10. Claim 16

Apparatus according to claim 15, the base computer is constructed and arranged so that if at least one other object to which said object is related is classed as malware, then said object is classified as malware.

As shown regarding analogous limitations in claim 24, Halperin discloses the limitations of claim 16. EX1002, ¶¶199-203,261. Accordingly, claim 16 should be found invalid under 35 U.S.C. § 103(a) based on Halperin in view of Mankins.

11. Claim 17

Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include the object or similar objects stored on at least sonic [sic] of said remote computers.

As shown regarding analogous limitations in claim 5, Halperin discloses the limitations of claim 17. EX1002, ¶¶206-208,264. Accordingly, claim 17 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins.

12. Claim 24

A method according to claim 13, wherein, if the monitored behavior of the object running on the remote computer exhibits non-permitted behavior that extends beyond that permitted by the mask, the base computer is arranged to compare the non-permitted behavior with the behavior of other objects and if said behavior is known to be not malicious for said other objects the base computer is arranged to allow said behavior for the object and to modify the mask to allow that behavior for that object in future.

Petitioner notes that claim 13 from which claim 24 depends is an apparatus claim, not a method claim. As shown regarding analogous limitations in claim 12, Halperin and Mankins teach the limitations of claim 24. EX1002, ¶¶215-222,267. Accordingly, claim 24 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins.

B. Ground 2: Claims 6 and 18 are unpatentable under 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin, Mankins, and Arnold.

A PHOSITA would have been motivated to combine the teachings of Halperin-Mankins with those of Arnold because all three publications solve the same problem of identifying and isolating previously unknown malware in a computer network. EX1002, ¶¶270-272. Halperin notes that many anti-virus programs use code signatures to identify viruses, which limits their ability to identify new or modified viruses. EX1003, ¶¶[0005]-[0007]; EX1002, ¶¶273-283. Halperin addresses this problem by collecting and correlating network events to establish behavior patterns and isolate anomalies. EX1003, ¶¶[0037]-[0057]; EX1002, ¶263.

Likewise, Arnold uses network information to detect anomalies and isolate suspect software. EX1008, 5:1-26; EX1002, ¶280. The systems of both Halperin and Arnold operate to generate identifiers for the anomalies; Halperin generates behavior profiles and Arnold generates N-gram sequences of invariant portions of the previously unknown malware. EX1002, ¶¶272-283; EX1003, ¶[0038]; EX1008, 4:60-67. Accordingly, Arnold, Halperin and Mankins disclose compatible systems that a PHOSITA would have had a reasonable expectation of success in combining. EX1002, ¶284.

A PHOSITA would have been motivated to modify the system of Halperin to better isolate worms detected using methods of Halperin so possibly-benign software launching the worm may be isolated. EX1002, ¶285-287; EX1008, 21:3-65. Halperin discloses that loading of a new software application into RAM may be detected as an event for possible anomalous behavior. EX1002, ¶285; EX1003, ¶[0043]. Many worms, though, rely on separate launching scripts that may require isolation to contain the worm. EX1002, ¶¶285-291; EX1004, 21:20-40. To detect worms, a PHOSITA would recognize that correlation and additional analysis would be needed to flag anomalous loading of software applications into RAM. EX1002, ¶285-291. Arnold solves this problem by providing processes for capturing and containing unknown viruses and worms, including identifying the launching script of the worm. EX1002, ¶285-291; EX1008, 21:10-30. Accordingly,

a PHOSITA would have found it obvious and advantageous to also correlate the loading of software into RAM with other behavior and events as described in Arnold so as to better contain worms. EX1002, ¶¶270-292.

1. Claim 6

A method according to claim 3, wherein said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object.

Halperin teaches capturing events that involve loading a software application (corresponding to “child object”) into memory to launch the application. EX1002, ¶295; EX1003, ¶[0043]. Arnold discloses that launching a software application is a key event in the spread of worms, which are a type of malware. EX1002, ¶296; EX1008, 21:10-25. Arnold, like Halperin, discloses that detecting activity anomalies is a primary step to identify malware, enabling the originating software and the entire process family tree to be isolated for signature identification. EX1002, ¶¶296-302; EX1008, 21:17-41. Arnold explains that “the method proceeds to Step B, in which that process, all parents or children of that process, and perhaps all other processes in memory, are scanned for known worms.” Thus, Arnold describes identifying family (parent-child) relationships between objects or process tree relationships between objects. EX1002, ¶¶303-306; EX1008, 21:30-42. A PHOSITA would have recognized that combining Halperin, Mankins and Arnold enables better tracking and monitoring of start-up scripts and worm

spawning so that worms can be more effectively identified and contained. EX1002, ¶305-306.

Thus, Halperin and Arnold disclose the claim 6 limitations. EX1002, ¶295-307. Accordingly, claim 6 should be found invalid under 35 U.S.C. § 103(a) based on Halperin, Mankins, and Arnold.

2. Claim 18

Apparatus according to claim 15, wherein the base computer is constructed and arranged so that said other objects include other objects that are parent objects or child objects or otherwise process-related objects to said object.

As shown regarding analogous limitations in claim 6, Halperin teaches capturing events that involve loading a software application (corresponding to “child object”) into memory to launch the application. EX1002, ¶295-307,309; EX1003, ¶[0043]. As shown regarding claim 6, Halperin-Mankins in combination with Arnold teaches the claim 18 limitations. Accordingly, claim 18 should be found invalid under 35 U.S.C. § 103(a) based on Halperin, Mankins, and Arnold.

C. Ground 3: Claims 7-11, 19-23, and 25-30 are unpatentable under pre-AIA 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin (EX1003), Mankins (EX1004), Kirby (EX1006), and Carpentier (EX1013).

A PHOSITA would have been motivated to combine the teachings of Halperin and Mankins as described regarding Ground 1 and particularly claim

limitation 1[f]. EX1002, ¶¶107-113,152-157.

A PHOSITA would have been further motivated to combine the teachings of Halperin and Mankins with teachings of Kirby at least because Kirby is similarly directed to network security and malware containment, and provides the Halperin-Mankins combination with a more efficient method of transmitting data and comparing objects. EX1002, ¶¶313-315,321; EX1006, 4:10-16,7:41-62. The application data, process execution data, and file information that is collected and analyzed by the Halperin-Mankins combination would require transmission from the network computers to the server. EX1002, ¶313; EX1003, ¶[0073]. A PHOSITA would have been motivated to implement the data transmissions of Halperin-Mankins using the hashing methods of Kirby. EX1002, ¶¶321-323; EX1003, ¶¶[0047],[0073]; EX1004, 8:15-42; EX1006, 3:35-58. Further, a PHOSITA applying Kirby to Halperin-Mankins to efficiently transmit data and compare objects would have looked to Carpentier for further guidance as Carpentier taught how to use hashes and signatures to communication data over networks generally. EX1002, ¶¶323-325. Thus, a PHOSITA would have understood from Carpentier the broad applications of data hashings/signatures, and therefore been motivated to implement the descriptor files and metadata hashing as taught in Carpentier in the Halperin-Mankins-Kirby system to efficiently provide much of the data used by Halperin regarding network computers (e.g., file

locations, time stamps, etc.). EX1002, ¶¶313-326; EX1013, ¶¶[0030]-[0034],[0060]. The results from the application of hashes of Kirby-Carpentier to the system of Halperin-Mankins would have been predictable to a PHOSITA since both Halperin and Mankins use hashing (checksumming is a form of hashing) to characterize the behavior information and it was a common approach before the earliest priority date of the '250 patent. EX1002, ¶¶313-326; EX1004, 8:15-42; EX1003, ¶[0073].

1. Claim 7

A method according to claim 1, wherein the data is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on the respective remote computers.

Halperin teaches using a checksum (corresponding to a “key”) to characterize attachments for future identification and teaches sharing these checksums with computers on the network for comparison (and, therefore, checksumming on those computers) (corresponding to “carried out in respect of the objects on the respective remote computers”). EX1003, ¶[0073]; EX1002, ¶¶328-329. Additionally, Mankins teaches using hashing in the Bloom filter. EX1004, 8:15-42.

Likewise, Kirby teaches generating hashes of incoming suspect objects (e.g., emails, files, attachments) and storing these hashes in a signature database (corresponding to “the data is sent in the form of key that is obtained by a hashing

process”). EX1002, ¶¶330-331; EX1006, 4:10-16,7:41-62. These hashes are used to check or compare future objects. *Id.* Further, Carpentier teaches that the contents of assets (i.e., files) are hashed for a unique signature to identify the contents should it appear elsewhere. EX1002, ¶¶332-336; EX1013, ¶¶ [0013],[0033]. Carpentier also teaches generating a descriptor file for an asset that includes its location on a computer and hashing the descriptor file for an identifier. EX1002, ¶¶334-336; EX1006, ¶[0031].

A PHOSITA would have had an expectation of success in combining the hashing process of Kirby-Carpentier with the system of Halperin-Mankins at least because Halperin also teaches checksums for file identification across a network. EX1002, ¶¶337; EX1003, ¶[0073]. Also, Mankins Bloom filter uses hashing. EX1004, 8:15-42. A PHOSITA would have been motivated to add the hashing of Kirby and Carpentier as described above in the motivation to combine. EX1002, ¶¶313-326.

Therefore, a PHOSITA would have found the limitations recited in claim 7 obvious in view of the Halperin-Mankins- Kirby-Carpentier combination. EX1002, ¶¶328-339. Accordingly, claim 7 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

2. Claim 8

A method according to claim 7, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

Halperin teaches generating checksums of attachments that include software (corresponding to “the key has at least one component that represents executable instructions contained within or constituted by the object”). EX1003, ¶[0073]. Further, Kirby and Carpentier both teach hashing files including software contained in the files. EX1002, ¶342; EX1006, 4:45-55 (signature of a virus); EX1013, ¶[0035].

Therefore, a PHOSITA would have found the claim 8 limitations obvious in view of Halperin and Kirby-Carpentier. EX1002, ¶341-343. Accordingly, claim 8 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

3. Claim 9

A method according to claim 7, wherein the key has at least one component that represents data about said object.

Halperin teaches generating checksums of attachments that include software. EX1003, ¶[0073]. Carpentier also teaches hashing file contents and descriptors to ensure data integrity. EX1002, ¶¶346-347; EX1006, ¶¶[0031]-[0034]. Therefore, Halperin and Mankins in further view of Kirby and Carpentier teaches the claim 9 limitation.

Therefore, a PHOSITA would have found the claim 9 limitation obvious in view of Halperin and Kirby-Carpentier. EX1002, ¶¶345-347. Accordingly, claim 9 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

4. Claim 10

A method according to claim 9, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

Carpentier teaches hashing file descriptions in order to easily and efficiently characterize files (corresponding to “said data about said object”) on a computer. EX1002, ¶¶350-351; EX1006, ¶¶[0031]-[0034]. Attributes disclosed in Carpentier as being hashed to identify the file include: file size, file name, time/date stamp, and other information. EX1002, ¶¶350-351; EX1006, [0031], Fig. 5. Accordingly, Carpentier teaches hashing at least two of the alternative features listed in claim 10, disclosure of any one of which is sufficient to render the claim obvious. *See e.g., Titanium Metals Corp.*, 778 F.2d at 782; EX1002, ¶¶349-352.

Therefore, a PHOSITA would have found claim 10 limitations obvious in view of Kirby-Carpentier. *Id.* Accordingly, claim 10 should be found invalid

under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

5. Claim 11

A method according to claim 7, wherein the key has at least one component that represents the physical size of the object.

Carpentier teaches hashing file descriptors in order to easily and efficiently characterize files (corresponding to “the key has at least one component that represents”) on a computer. EX1002, ¶¶354-355; EX1006, ¶¶[0031]-[0034]. Descriptors disclosed in Carpentier as being hashed include: file size (corresponding to “represents the physical size of the object”) and file location. EX1002, ¶¶354-356; EX1006, ¶[0031], Fig.-5.

Therefore, a PHOSITA would have found the claim 11 limitations obvious in view of Kirby-Carpentier. EX1002, ¶356. Accordingly, claim 11 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

6. Claim 19

Apparatus according to claim 13, wherein the base computer is constructed and arranged so as to be able to process data that is sent in the form of key that is obtained by a hashing process carried out in respect of the objects on said respective remote computers.

As shown regarding analogous limitations in claim 7, the combination of Halperin, Mankins, and Kirby-Carpentier teaches the limitations of claim 19, and

Halperin and Mankins disclose all limitations of claim 13 from which claim 19 depends. EX1002, ¶¶328-338,358-359. Accordingly, claim 19 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

7. Claim 20

Apparatus according to claim 19, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

As shown regarding analogous limitations in claim 8, the combination of Halperin, Mankins, and Kirby-Carpentier discloses the limitations of claim 20, as well as all limitations of claims 13 and 19, from which claim 20 depends. EX1002, ¶¶341-343,362-363. Accordingly, claim 20 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

8. Claim 21

Apparatus according to claim 19, wherein the key has at least one component that represents data about said object.

As shown regarding analogous limitations in claim 9, the combination of Halperin, Mankins, and Kirby-Carpentier teaches the limitations of claim 21. As shown, Halperin and Mankins disclose all limitations of claims 13 and 19, from which claim 21 depends. EX1002, ¶¶345-347,365-366. Accordingly, claim 21

should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

9. Claim 22

Apparatus according to claim 21, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

As shown regarding analogous limitations in claim 10, Kirby-Carpentier teaches the limitations of claim 22, and the Halperin-Mankins-Kirby-Carpentier combination discloses all limitations of claims 13, 19, and 21, from which claim 22 depends. EX1002, ¶¶349-352, 368-369. Accordingly, claim 22 should be found invalid under 35 U.S.C. § 103(a) based on Halperin, Mankins, Kirby, and Carpentier.

10. Claim 23

Apparatus according to claim 19, wherein the key has at least one component that represents the physical size of the object.

As shown regarding analogous limitations in claim 11, Kirby-Carpentier teaches the limitations of claim 23. As shown, Halperin and Mankins discloses all limitations of claims 13 and 19, from which claim 23 depends. EX1002, ¶¶354-

356,371-372. Accordingly, claim 23 should be found invalid under 35 U.S.C.

§ 103(a) based on Halperin, Mankins, Kirby, and Carpentier.

11. Claim 25

25[a] A method of providing data about a computer object from a remote computer to a base computer so that a comparison can be made at the base computer with similar data received from other remote computers, the method comprising:

Halperin discloses methods for detecting malicious activity on a computer network. EX1002, ¶¶118-123,231-232,235,375; EX1003, ¶¶[0007]-[0008],[0023], Fig.2,7. Further, Halperin discloses receiving data at a server from computers on a network (corresponding to “providing data about a computer object from a remote computer to a base computer”) for categorizing and comparing behaviors of computer processes on those computers. EX1002, ¶118-123,375; EX1003, ¶¶[0047],[0071],[0075],[0087]-[0089], Fig.11. Computers across the network use the shared information to correlate parameters that characterize suspicious software to identify similar software and behaviors (corresponding to “a comparison can be made at the base computer with similar data received from other remote computers”) on the computers including at the centralized server. EX1002, ¶¶126-131,375; EX1003, ¶¶[0037]-[0059].

Therefore, Halperin discloses limitation 25[a]. EX1002, ¶376.

25[b] providing from a remote computer to a base computer data about a computer object that is stored on the remote computer;

Halperin discloses a centralized server (corresponding to “base computer”) that receives data about network and software activity from multiple computers (corresponding to “a remote computer”) on the network. EX1002, ¶¶131,378; EX1003, ¶¶[0047],[0087]. Halperin discloses that infected messages are identified and descriptions of the suspicious behavioral activity from software (corresponding to “data about a computer object that is stored on the remote computer”) on these computers are shared across the network including with the centralized server. EX1002, ¶¶126,378; EX1003, ¶¶[0022],[0034],[0047]-[0059].

Therefore, Halperin discloses limitation 25[b]. EX1002, ¶379.

25[c] the data including information about the behavior of the object running on one or more remote computers and including one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers;

As shown regarding analogous limitations in claim limitation 1[g], Halperin discloses running and monitoring the object on remote computers. EX1002, ¶¶160,381. As shown regarding analogous limitations in claim 2, Halperin discloses at least two of the alternatives listed in limitation 25[c]. EX1002, ¶¶187-188,381.

Therefore, Halperin discloses limitation 25[c]. EX1002, ¶382.

25[d] the data being sent in the form of key that is obtained by a hashing process carried out in respect of the object on the remote computer;

Halperin teaches using a checksum to characterize files. EX1003, ¶[0073].

As shown regarding analogous limitations in claim 7, Kirby-Carpentier teaches the remainder of this limitation. EX1002, ¶¶328-338,382.

A PHOSITA would have had an expectation of success in combining the hashing process of Kirby-Carpentier with the system of Halperin-Mankins at least because Halperin and Mankins also teach checksums for file identification across a network. EX1002, ¶313-326. A PHOSITA would have been motivated to add the hashing of Kirby-Carpentier as described above in the motivation to combine. EX1002, ¶¶313-326.

Therefore, the combination of Halperin, Mankins, Kirby, and Carpentier discloses limitation 25[d]. EX1002, ¶385.

25[e] automatically generating a mask for the computer object, the mask defining acceptable behavior for the object built in accordance with normal behavior of the object determined from said received data;

As shown regarding analogous limitations in claim limitation 1[f], the combination of Halperin and Mankins discloses this limitation. EX1002, ¶¶148-156,387.

Both the behavioral analysis disclosed in Halperin and the Bloom filter for behavior disclosed in Mankins are automatically generated. EX1002, ¶¶148,150; EX1004, 8:8-13; EX1003, ¶¶[0061],[0081],[0086]-[0088]. As described above in Ground 1, it would have been obvious to a PHOSITA to generate the Bloom filter of normal behavior disclosed in Mankins on the central server of Halperin after receiving information about a software's system calls from the remote computers as disclosed by Halperin. EX1002, ¶¶152-154. In particular, this combination would have been obvious to efficiently implement the distributed intrusion detection system proposed by Mankins. *Id.* A PHOSITA would have an expectation of success in this combination because the data structure disclosed in Mankins can be assembled anywhere with the system call information as an input. *Id.* Further, a PHOSITA would have been motivated to use Mankins' Bloom filter because it obviates a risky tradeoff between false alarms and virus detection as disclosed to be an issue in Halperin. EX1002, ¶¶155-157,387; EX1004, 8:38-41; EX1003, ¶[0090].

Therefore, Halperin in view of Mankins discloses limitation 25[e]. EX1002, ¶388.

25[f] executing the computer object at the remote computer;

As shown regarding analogous limitations in claim element 1[g], Halperin discloses limitation 25[f]. EX1002, ¶¶160,390-391.

25[g] automatically monitoring operation of the computer object at the remote computer compared to the behavior permitted by the mask;

As shown regarding analogous limitations in claim elements 1[h] and 1[j], the combination of Halperin and Mankins discloses limitation 25[g]. EX1002, ¶¶165,173-176,393-394.

25[h] allowing the object to continue to run when behavior of the computer object is permitted by the mask;

As shown regarding analogous limitations in claim element 1[i], the combination of Halperin and Mankins discloses limitation 25[h]. EX1002, ¶¶169-171,396-397.

25[i] disallowing the object to run when the actual monitored behavior of the object extends beyond that permitted by the mask; and,

As shown regarding analogous limitations in claim element 1[j], the combination of Halperin and Mankins discloses limitation 25[i]. EX1002, ¶¶175-177,399-400.

25[j] reclassifying the object as malware when the actual monitored behavior extends beyond that permitted by the mask, wherein the base computer is arranged to receive information about the behavior of the object running on one or more remote computers, wherein the step of generating a mask for that object comprises building the mask with the base computer in accordance with normal behavior of the object determined from said received information.

As shown regarding analogous limitations in claim elements 1[b], 1[f], and 1[k], the combination of Halperin and Mankins discloses limitation 25[j]. EX1002, ¶¶402-403.

Further, as described regarding claim elements 25[a], 25[b], and 25[e], Halperin teaches collecting behavioral information in a learning database that is analyzed to identify normal software behavior and suspicious behavior (corresponding to “building... in accordance with normal behavior of the object determined from said received information”). EX1002, ¶¶402-404; EX1003, ¶¶[0071],[0076].

Therefore, Halperin and Mankins discloses this limitation. EX1002, ¶403-404.

For the foregoing reasons, a PHOSITA would have found claim 25 to be obvious in view of Halperin, Mankins, Kirby, and Carpentier. EX1002, ¶¶375-404. Accordingly, claim 25 should be found invalid under 35 U.S.C. § 103(a) based on Halperin, Mankins, Kirby, and Carpentier.

12. Claim 26

A method according to claim 25, wherein the key has at least one component that represents executable instructions contained within or constituted by the object.

As shown regarding analogous elements in claim 8, Halperin and Mankins teaches this limitation. EX1002, ¶¶314-315,406. Accordingly, claim 26 should be found invalid under 35 U.S.C. § 103(a) based on Halperin, Mankins, Kirby, and Carpentier.

13. Claim 27

A method according to claim 25, wherein the key has at least one component that represents data about said object.

As shown regarding analogous element in claim 9, Halperin and Mankins teaches this limitation. EX1002, ¶¶318-319,409. Accordingly, claim 27 should be found invalid under 35 U.S.C. § 103(a) based on Halperin, Mankins, Kirby, and Carpentier.

14. Claim 28

A method according to claim 27, wherein said data about said object includes at least one of: the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

As shown regarding analogous limitations in claim 10, Carpentier teaches the limitations of claim 28. EX1002, ¶¶322-326,412. Accordingly, claim 28 should be found invalid under 35 U.S.C. § 103(a) based on Halperin, Mankins, Kirby, and Carpentier.

15. Claim 29

A method according to claim 25, wherein the key has at least one component that represents the physical size of the object.

As shown regarding analogous element in claim 11, Kirby-Carpentier teaches the limitations of claim 29. EX1002, ¶¶329-330,415. Accordingly, claim 29 should be found invalid under 35 U.S.C. § 103(a) based on Halperin, Mankins, Kirby, and Carpentier.

16. Claim 30

A non-transitory storage medium storing a computer program comprising program instructions for causing a computer to perform the method of claim 25.

Halperin teaches hardware and software and that the methods disclosed in Halperin may be embodied as hardware or software. EX1002, ¶418; EX1003, ¶¶0092]. A POSITA would have known that software is stored in a non-transitory storage medium, such as hard drives. EX1002, ¶418. A POSITA would have further known that software is equivalent to a “computer program” and that computer programs, by definition, contain “program instructions.” EX1002, ¶418.

Therefore, a PHOSITA would have found the limitations recited in claim 30 obvious in view of the Halperin-Mankins-Kirby-Carpentier combination. EX1002, ¶419. Accordingly, claim 30 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Mankins in view of Kirby and Carpentier.

D. Ground 4: Claims 1-5, 12-17, and 24 are unpatentable under 35 U.S.C. § 103(a)(pre-AIA) as obvious in view of Halperin and Inoue.

A PHOSITA would have been motivated to combine teachings of Halperin and Inoue because both references are directed to containing malware, in which Halperin provides reactive, post-infection solutions and Inoue provides proactive, largely pre-infection solutions for containment. EX1002, ¶¶421-424. The disclosures of Halperin and Inoue provide different but compatible capabilities that when combined would yield a more complete defense-in-depth solution for an enterprise network. *Id.* A PHOSITA would have been motivated to avoid (or reduce) tradeoffs between false positives in malware detection and effectiveness disclosed in Halperin by adding the proactive measures to reduce false positives disclosed in Inoue. EX1002, ¶¶425-427. Further, a PHOSITA would have had an expectation of success in combining the teachings of Inoue with Halperin.

EX1002, ¶427.

1. Claim 1

1[a] See claim element in Ground 1

As shown in Ground 1 Halperin discloses limitation 1[a]. EX1002, ¶¶430-437.

1[b] See claim element in Ground 1

As shown in Ground 1 Halperin discloses limitation 1[b]. EX1002, ¶¶439-446.

1[c] See claim element in Ground 1

As shown in Ground 1 Halperin discloses limitation 1[c]. EX1002, ¶¶448-450.

1[d] See claim element in Ground 1

As shown in Ground 1 Halperin discloses limitation 1[d]. EX1002, ¶¶452-455.

1[e] See claim element in Ground 1

As shown in Ground 1 Halperin discloses limitation 1[e]. EX1002, ¶¶457-459.

1[f] automatically generating a mask for the computer object that defines acceptable behavior for the computer object, wherein the mask is generated in accordance with normal behavior of the object determined from said received data;

As shown in Ground 1, Halperin discloses certain compatible features of limitation 1[f]. EX1002, ¶¶143-149,461-462.

In a complementary manner, Inoue discloses a dynamic sandbox that provides a Java virtual machine (JVM) with monitoring capabilities that records the behavior of software running on the JVM. EX1002, ¶¶463-464; EX1005, pp.3. Inoue discloses that recorded behavior is formed into a profile (corresponding to “a mask”) of acceptable behavior of the software. EX1002, ¶¶468-469; EX1005, pp.3. The profile is used to filter behavior and identify malicious infections. *Id.*

As described regarding the '250 patent's prosecution history, Applicants differentiated the '250 patent's claims from a sandbox by alleging that the sandbox

was “an isolated sandbox” “on a central computer” that was “monitored by a human operator.” EX1007, pp.334; EX1002, ¶¶463-467. In contrast, the sandbox described in Inoue may be trained on a central computer as described above (or any computer), and then deployed to operate in sandboxing mode based on the trained profile to automatically monitor behavior of the program operating on the JVM. EX1005, pp.1,6; EX1002, ¶¶72(citing-EX1010),465-466. In sandboxing mode, the JVM and trained profile disclosed in Inoue may be provided on a remote computer so that software executing in that environment is automatically monitored at the remote computer as a form of application intrusion detection system. *Id.* The disclosure of Inoue differentiates the described dynamic sandboxing of its disclosure from “the standard sandbox” and offers dynamic sandboxing as an alternative to user-defined policies. EX1005, pp.3,6; EX1002, ¶¶463-469. Therefore, a PHOSITA would have understood that the “sandbox” of Inoue operates using a mask and not as a “standard sandbox” requiring human monitoring. *Id.*

A PHOSITA would have found it obvious to combine the profiles and modified JVM of Inoue with the behavioral data collection and analysis of Halperin because the two approaches to gathering information are compatible and complementary. EX1002, ¶¶463-473. A PHOSITA would have found it obvious to execute software in a JVM as described in Inoue on the centralized server of

Halperin to collect operating parameters and validate behavior of the program.

EX1002, ¶¶463-473. Further, a PHOSITA would have found it obvious to improve the profile (i.e., “mask”) built on the centralized computer by the JVM of Inoue by including additional data from the behavior of the software on network computers that is collected by the system in Halperin. EX1002, ¶¶463-473.

A PHOSITA would have been motivated to modify analysis of data collected from software across the network and on the server as taught by Halperin with the JVM collection and profile building of Inoue at least because dynamic analysis of the sandbox of Inoue reduces the tradeoff between false alarms and allowed malicious activity. EX1002, ¶¶468-473; EX1005, pp.3.

Therefore, Halperin and Inoue discloses limitation 1[f]. EX1002, ¶474.

1[g] running said object on at least one of the remote computers;

As shown in Ground 1, Halperin discloses allowing software to run on remote computers until certain behavioral thresholds are reached. EX1002, ¶476; EX1003, ¶¶[0069]-[0071]. Likewise, Inoue teaches executing software in the JVM sandbox (sandboxing mode) on various computers after building the profile so as to characterize acceptable behavior on those computers. EX1002, ¶¶476-478; EX1005, pp.4-5.

Therefore, Halperin and Inoue discloses limitation 1[g]. EX1002, ¶479.

1[h] automatically monitoring operation of the object on the at least one of the remote computers;

As shown in Ground 1 regarding element 1[h], Halperin discloses automatic monitoring. EX1002, ¶¶481-482. Further, Inoue discloses a dynamic sandbox that provides a JVM with monitoring capabilities that records the behavior of software running on the JVM. EX1002, ¶¶463-468, 481-482; EX1005, pp.3.

Therefore, Halperin and Inoue discloses limitation 1[h]. EX1002, ¶483.

1[i] allowing the computer object to continue to run when behavior of the computer object is permitted by the mask;

As shown in Ground 1 regarding element 1[i], Halperin discloses allowing software (corresponding to “the object”) to run on remote computers when certain suspicious behaviors are below a threshold. EX1002, ¶¶485; EX1003, ¶¶[0087],[0090].

Inoue teaches that when a program is executed on the JVM in sandboxing mode, the program is allowed to execute functions (corresponding to “allowing the computer object to continue to run”) until one deviates from the pre-established profile (corresponding to “when behavior of the computer object is permitted by the mask”). EX1002, ¶486; EX1005, pp.4-5.

Therefore, Halperin and Inoue discloses limitation 1[i]. EX1002, ¶487.

1[j] disallowing the computer object to run when the actual monitored behavior of the computer object extends beyond that permitted by the mask; and,

As shown in Ground 1 regarding element 1[j], Halperin teaches both containment and quarantine of software that has been determined to be malicious based on the received data from computers. EX1002, ¶489; EX1003, ¶¶[0075]-[0076],[0079],[0090].

Inoue discloses that, after training a profile for the JVM sandbox (corresponding to “mask”), the profiled program can be executed in the sandbox in sandboxing mode and any malicious code will be blocked by the sandbox based on the profile of behavior. EX1002, ¶490; EX1005, pp.4-5.

Therefore, Halperin and Inoue discloses limitation 1[j]. EX1002, ¶491.

1[k] reclassifying the computer object as malware when the actual monitored behavior extends beyond that permitted by the mask.

As shown in Ground 1 regarding element 1[e], Halperin discloses that software behavior may initially be classified as not malware despite being suspicious. Further, as shown in Ground 1 regarding element 1[k], Halperin discloses that if the suspicious behavior spreads to groups of computers which have few connections or normal correlated activity, then the activity may be identified as malicious. EX1002, ¶¶493-495; EX1003, ¶¶[0037],[0063]-[0064],[0078]-[0081].

Likewise, as described regarding limitation 1[j], Inoue discloses a dynamic sandbox JVM with monitoring capabilities that records the behavior of software running on the JVM. EX1002, ¶¶463-468, 497-498; EX1005, pp.3. Inoue discloses that the recorded behavior is formed into a profile (corresponding to “a mask”) of acceptable behavior of the software such that unacceptable behavior is prevented by the sandbox. *Id.*

A PHOSITA would have found it obvious to combine the profiles and modified JVM of Inoue with the behavioral data collection and analysis of Halperin because the two approaches to gathering information are compatible and complementary. EX1002, ¶¶421-427, 493-499.

Therefore, Halperin and Inoue disclose limitation 1[k]. EX1002, ¶500.

For the foregoing reasons, a PHOSITA would have found claim 1 to be obvious in view of the combination of Halperin and Inoue. EX1002, ¶428.

Accordingly, claim 1 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

2. Claim 2

A method according to claim 1, wherein the data about the computer object that is sent from the plural remote computers to the base computer includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

As shown, Halperin discloses the limitations of claim 2, and the Halperin-Inoue combination discloses all elements of claim 1, from which claim 2 depends.

EX1002, ¶¶502-504. Accordingly, claim 2 should be found invalid under 35

U.S.C. § 103(a) based on Halperin and Inoue.

3. Claim 3

A method according to claim 1, wherein determining identifies relationships between the object and other objects.

As shown, Halperin discloses the limitations of claim 3, and the Halperin-Inoue combination discloses all elements of claim 1, from which claim 3 depends.

EX1002, ¶¶507-511. Accordingly, claim 3 should be found invalid under 35

U.S.C. § 103(a) based on Halperin and Inoue.

4. Claim 4

A method according to claim 3, wherein if at least one other object to which said object is related is classed as malware, then classifying said object as malware.

As shown, Halperin discloses the limitations of claim 4, and the Halperin-Inoue combination discloses all elements of claims 1 and 3 from which claim 4 depends. EX1002, ¶¶513-518. Accordingly, claim 4 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

5. Claim 5

A method according to claim 3, wherein said other objects include the object or similar objects stored on at least some of the remote computers.

As shown, Halperin discloses the limitations of claim 5, and the Halperin-Inoue combination discloses all elements of claims 1 and 3, from which claim 5 depends. EX1002, ¶¶520-522. Accordingly, claim 5 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

6. Claim 12

A method according to claim 1, comprising if the monitored behavior of the object running on the remote computer exhibits non-permitted behavior that extends beyond that permitted by the mask, comparing at the base computer the non-permitted behavior with the behavior of other objects and if said behavior is known to be not malicious for said other objects, allowing said behavior for the object and modifying the mask to allow that behavior for that object in future.

As shown in Ground 1, Halperin discloses various limitations of claim 12.

In a compatible manner Inoue teaches the remaining limitations. Thus, Halperin and Inoue discloses all limitations of claim 12. EX1002, ¶¶524-529.

Inoue discloses that, after training a profile for the JVM sandbox (corresponding to “mask”), the profiled program can be executed in the sandbox in sandboxing mode and any malicious code will be blocked by the sandbox. EX1002, ¶¶463-468, 530-532; EX1005, pp.4-5. Inoue discloses that the sandbox may operate in training mode and sandboxing mode at the same time for dynamic sandboxing (learning from normal behavior). EX1005, pp.1,4. A PHOSITA would have understood that a JVM operated as a dynamic sandbox could dynamically build a profile on the server of Halperin based on updated information from multiple objects received from remote computers in the network of Halperin (corresponding to “comparing at the base computer the non-permitted behavior with the behavior of other objects... modifying the mask to allow that behavior for that object in future”) as described above. EX1002, ¶¶530-533.

Therefore, a PHOSITA would have found claim 12 limitations obvious in view of Halperin and Inoue. EX1002, ¶¶533. Accordingly, claim 12 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

7. Claim 13

a. Apparatus for classifying a computer object as malware, the apparatus comprising:

As shown in Ground 1, Halperin discloses limitation 13[a]. EX1002, ¶¶536-538.

b. a base computer constructed and arranged to receive data about a computer object from each of plural remote computers on which the object or similar objects are stored, the data including information about the behavior of the object running on one or more remote computers;

As shown in Ground 1, Halperin discloses limitation 13[b]. EX1002, ¶541.

c. the base computer being constructed and arranged to determine whether the data about the computer object received from said plural computers indicates that the computer object is malware; and,

As shown in Ground 1, Halperin discloses limitation 13[c]. EX1002, ¶544.

d. the base computer being constructed and arranged to classify the computer object as malware when the base computer determines that the data indicates that the computer object is malware,

As shown in Ground 1, Halperin discloses limitation 13[d]. EX1002, ¶547.

- e. wherein the base computer is constructed and arranged to automatically generate a mask for an object that is initially classed not as malware, said mask defining acceptable behavior for the object built in accordance with normal behavior of the object determined from said received data,**

As shown in Ground 1 for limitations 13[e]-13[f], Halperin discloses automatically identifying normal software behavior and suspicious behavior.

EX1002, ¶¶236,528; EX1003, ¶¶[0071],[0076]. Likewise, as shown in Ground 4 for limitations 1[e]-1[f], the remaining elements of this limitation are taught by Inoue. EX1002, ¶¶457-473,550.

- f. wherein operation of the object is automatically monitored when the object is running on at least one of the remote computers, the object being allowed to continue to run when behavior of the object is permitted by the mask, the object not being permitted to run when the actual monitored behavior of the object extends beyond that permitted by the mask, and the object is reclassified as malware when the actual monitored behavior extends beyond that permitted by the mask.**

As shown in Ground 4 for limitations 1[g]-1[k], the combination of Halperin and Inoue discloses this limitation. EX1002, ¶¶553.

For the foregoing reasons, a PHOSITA would have found claim 13 obvious in view of Halperin and Inoue. EX1002, ¶¶554. Accordingly, claim 13 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

8. Claim 14

Apparatus according to claim 13, the data includes one or more of: executable instructions contained within or constituted by the object; the size of the object; the current name of the object; the physical and folder location of the object on disk; the original name of the object; the creation and modification dates of the object; vendor, product and version and any other information stored within the object; the object header or header held by the remote computer; and, events initiated by or involving the object when the object is created, configured or runs on the respective remote computers.

As shown in Ground 4 for analogous claim 2, Halperin discloses the limitations of this claim. EX1002, ¶557. Accordingly, claim 14 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

9. Claim 15

Apparatus according to claim 13, wherein the base computer is constructed and arranged so that the determining identifies relationships between the object and other objects.

As shown in Ground 4 for analogous claim 3, Halperin discloses the limitations of this claim. EX1002, ¶560. Accordingly, claim 15 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

10. Claim 16

Apparatus according to claim 15, the base computer is constructed and arranged so that if at least one other object to which said object is related is classed as malware, then said object is classified as malware.

As shown in Ground 4 for analogous claim 4, Halperin discloses the limitations of this claim. EX1002, ¶563. Accordingly, claim 16 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

11. Claim 17

Apparatus according to claim 15, wherein the base computer is constructed and arranged wherein the base computer is constructed and arranged so that said other objects include the object or similar objects stored on at least sonic [sic] of said remote computers.

As shown in Ground 4 for analogous claim 5, Halperin discloses this claim. EX1002, ¶566. Accordingly, claim 17 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

12. Claim 24

A method according to claim 13, wherein, if the monitored behavior of the object running on the remote computer exhibits non-permitted behavior that extends beyond that permitted by the mask, the base computer is arranged to compare the non-permitted behavior with the behavior of other objects and if said behavior is known to be not malicious for said other objects the base computer is arranged to allow said behavior for the object and to modify the mask to allow that behavior for that object in future.³

As shown in Ground 4 for analogous claim 12, the combination of Halperin and Inoue discloses these limitations. EX1002, ¶569. Accordingly, claim 24 should be found invalid under 35 U.S.C. § 103(a) based on Halperin and Inoue.

VI. NO BASIS FOR DISCRETIONARY DENIAL

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

There is parallel court litigation, so *Fintiv* factors are addressed preliminarily. First, no stay has been entered in the parallel litigation. Second, trial is currently set for August 19, 2024, so there is time to complete inter parte review. Third, it is early in the parallel litigation as fact discovery has not started. Fourth, plaintiffs have identified the Halperin, Mankins, Inoue, and Kirby prior art in Defendant's invalidity contentions filed in the parallel litigation. Finally, Petitioner and Webroot are parties in the parallel litigation as identified in the Mandatory Notices.

³ Method claim 24 depends from *apparatus* claim 23.

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

Halperin, Mankins, Arnold, Kirby, Carpentier and Inoue were not cited, individually or in combination, in any of the Office communications. *See* EX1006. Regarding *Becton* factors (a) (material differences between asserted art and prior art) and (b) (cumulative nature of the asserted art and prior art), Halperin, Mankins, Arnold, Kirby, Carpentier, and Inoue are materially different from the combination of references asserted in Office communications because Halperin, Mankins, Arnold, Kirby, Carpentier, and Inoue disclose the independent claim limitations that the data received from remote devices includes information about the behavior of the object and the mask is generated based on normal behavior of the object determined from the received data including behavior information. Thus, individually Halperin, Mankins, Arnold, Kirby, Carpentier and Inoue are not merely cumulative of references the Office relied. Regarding *Becton*

factor (d) (extent of overlap), the arguments in this Petition do not overlap with arguments made during examination.

Even if the Board finds that the “same or substantially the same” art or arguments were presented to the Office, discretionary denial under § 325(d) is still not proper because the Office made a “material error” in not considering the Halperin, Mankins, Arnold, Kirby, Carpentier and Inoue references.

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

Regarding *Becton* factor (e) (examiner error) and (f) (other facts), this petition specifically points out and explains how Halperin, Mankins, Arnold, Kirby, Carpentier and Inoue disclose all limitations of the claims.

VII. CONCLUSION

Inter partes review of claims 1-30 of U.S. Patent No. 8,418,250 is requested based on the Grounds described above.

Respectfully submitted,

The Marbury Law Group, PLLC

A handwritten signature in dark ink, appearing to read "Robert M. Hansen", is written over a horizontal line.

Robert M. Hansen, Reg. No. 43,656
Scott B. Amankwatia, Reg. No. 78,226
Matthew J. Anderson, Reg. No. 78,257

Attorneys for Petitioner

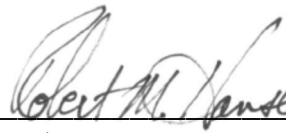
Date: March 2, 2023

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

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

The undersigned hereby certifies that the above-captioned **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 8,418,250**, Petitioner's power of attorney, and all associated exhibits were served in their entireties on March 2, 2023, on the following party via FedEx for overnight delivery to:

**Merchant & Gould P.C.
150 South Fifth Street, Suite 2200
Minneapolis, MN 55402**



Robert M. Hansen, Reg. No. 43,656

Attorney for Petitioner

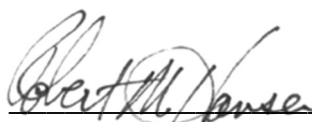
Date: March 2, 2023

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

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION

Pursuant to 37 C.F.R. § 42.24 *et seq.*, the undersigned certifies that this Petition complies with the 14,000-word type-volume limitation. The Petition contains **13,988** words, excluding the table of contents, table of authorities, exhibit list, mandatory disclosures, signature block, certificates and attached exhibits of the Petition.

The Marbury Law Group, PLLC

A handwritten signature in dark ink, appearing to read "Robert M. Hansen", is written over a horizontal line.

Robert M. Hansen, Reg. 43,656

ATTORNEY FOR PETITIONER

Date: March 2, 2023

THE MARBURY LAW GROUP, ^{PLLC}
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: 703-391-2900