

UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE PATENT TRIAL AND APPEAL BOARD

Sophos Ltd. and Sophos Inc.,
Petitioners

v.

Webroot Inc.,
Patent Owner.

Case No. IPR2023-00491

U.S. Patent No. 9,413,721

**Declaration of Dr. Richard Newman
In Support of Petitioners in the
Petition for *Inter Partes* Review of U.S. Patent No. 9,413,721**

Mail Stop **PATENT BOARD**
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

*Declaration of Dr. Richard Newman
In Support of Petitioners in the
Petition for Inter Partes Review of U.S. Patent No. 9,413,721*

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. BASIS FOR OPINIONS	2
A. Qualifications	2
B. Preparation for This Declaration	3
III. LEGAL STANDARDS	4
A. A Person of Ordinary Skill in the Art	4
B. Legal Standard for Prior Art.....	6
C. Claim Construction.....	6
D. Legal Standard for Obviousness	7
IV. TECHNOLOGY BACKGROUND.....	10
A. Overview of Malware.....	10
B. Overview of Malware-Detection Software	11
V. THE '721 PATENT	19
A. Overview of the '721 Patent.....	19
B. Overview of the Challenged Claims	22
VI. PRIOR ART CONSIDERED	24
A. Identification of the Primary Prior-Art References.....	25
B. General Overview of the Primary Prior-Art References.....	26
1. Liu	26
2. Tuvell	27
3. Kester	28

*Declaration of Dr. Richard Newman
In Support of Petitioners in the
Petition for Inter Partes Review of U.S. Patent No. 9,413,721*

TABLE OF CONTENTS

(continued)

	<u>Page</u>
C. Motivation to Combine and Reasonable Expectation of Success	28
VII. ANALYSIS OF THE CHALLENGED CLAIMS AGAINST THE PRIOR ART.....	31
A. Claim Construction.....	31
B. Application of the Prior Art to the Claims	31
C. Ground 1: Liu in View of Tuvell and Kester Renders Obvious Claims 1, 8, 9, 14, and 15.....	32
1. Independent Claim 1	32
2. Dependent Claim 8.....	63
3. Independent Claim 9	69
4. Dependent Claim 14	71
5. Dependent Claim 15	75
VIII. SECONDARY CONSIDERATIONS	76
IX. RIGHT TO SUPPLEMENT	76
X. CONCLUSION.....	78

I. INTRODUCTION

1. I, Dr. Richard Newman, have been retained by Gibson, Dunn & Crutcher LLP on behalf of Sophos Ltd. (“Petitioner”) as an independent expert in this Petition for *inter partes* review (“IPR”) by the U.S. Patent and Trademark Office. I have been asked to provide analysis and expert opinions on U.S. Patent No. 9,413,721 (“the ’721 Patent”), and, in particular, the validity of claims 1, 8, 9, 14, and 15 (“the challenged claims”).

2. In my view, the challenged claims of the ’721 Patent are invalid over the prior art I discuss in this Declaration. My opinion is that the prior art discloses the invention claimed in the Challenged Claims, so the Challenged Claims are invalid as obvious over the prior art.

3. I am being paid for my work on this case at my standard consulting rate and being reimbursed for my expenses. My compensation is not contingent upon the results of my study, the substance of my testimony, or the outcome of this case.

4. Based upon the facts and information available to me to date, I detail my opinions relevant to this petition for IPR. I am prepared to provide expert testimony on the opinions I express if asked about those issues by the Board or others.

5. In this Declaration I discuss my own work, teachings, and knowledge of the state of the art in the relevant time period, and I cite technical literature (like handbooks and textbooks) to demonstrate the state of the art in the relevant period.

6. Throughout this declaration, I refer to specific line numbers or page numbers of the relevant prior art or other technical documents. Those citations might not be exhaustive—there may be more evidence to support a particular proposition than I cite here in this Declaration.

II. BASIS FOR OPINIONS

A. Qualifications

7. I recently retired from the faculty of the University of Florida's Computer and Information Science and Engineering Department. I completed my Ph.D. in Computer Science from The University of Rochester Computer Science Department in Rochester, NY, in 1986, after I earned my M.S. there in 1983. I earned my BA in Mathematics from New College in Sarasota, FL, and attended Eckerd College in St. Petersburg, FL.

8. I introduced the graduate-level course on Computer and Network Security at the University of Florida. I have also taught graduate-level courses on networks, operating systems, distributed systems, and computer and information security; and special-topics courses in cryptography, cryptographic protocols,

anonymity, the theory of security, and cybersecurity. I also chaired the university-wide Networking and Telecommunications Standing Committee in the mid-1990s, which considered practical issues in network provision, management, and security for the UF campus-wide network and telecommunications facilities.

9. My publications include topics in security models, access control, steganography, covert channels, traffic analysis, and cryptographic protocols. I am co-inventor of seventeen patents, most of which disclose shared-networking functions. I am a member of the Institute of Electrical and Electronics Engineers (IEEE) and ACM (Association for Computing Machinery).

10. I have technical expertise in computer and network security; cryptography and cryptographic protocols; distributed systems; network protocols, topologies, and architecture; and powerline communications. My full qualifications and experience are set forth in my Curriculum Vitae, which is attached as Exhibit 1004.

B. Preparation for This Declaration

11. My findings contained in this declaration are based on my education, research, experience, and background in the fields of network computing and cybersecurity, as well as my review of prior-art references and other documents disclosed in this declaration. In forming my opinions, I have studied and considered

the '721 Patent, the file history of the '721 Patent, the prior-art references cited in this Declaration, and the other references I cite or refer to in this report.

12. I understand that this declaration will be submitted in support of a Petition requesting IPR of the '721 Patent.

III. Legal Standards

13. Counsel for the Petitioners have told me about the following legal standards for this case, which I have adhered to and applied throughout my analysis.

A. A Person of Ordinary Skill in the Art

14. Counsel for the Petitioners have told me that a person of ordinary skill in the art, or a “POSITA,” is a hypothetical person from whose viewpoint the prior art is analyzed (without the benefit of hindsight). A POSITA uses conventional wisdom in the art but does not try to innovate, either by extraordinary insights or by systematic research.

15. Counsel for the Petitioners have also told me that the hypothetical POSITA has knowledge of all prior-art references that are sufficiently related to one another and to the pertinent art, and to know of all prior art in the fields reasonably pertinent to the particular problem that the claimed invention purports to address.

16. In evaluating the prior-art references and other material, I have used the perspective of a POSITA in the field to which the patent is related as of the time of the patent’s priority date. I understand that a POSITA is presumed to be aware of

all pertinent prior art and the conventional wisdom in the art and is a person of ordinary creativity. I have applied this standard in my analysis throughout my declaration.

17. The '721 Patent is entitled "Methods and Apparatus for Dealing With Malware." Counsel for the Petitioners have told me that the 2010 timeframe is the earliest claimed effective filing date of the '721 Patent, and a POSITA at that time would have had at least a Bachelor's degree in Computer Science, Computer Engineering, Electrical Engineering, or an equivalent discipline, and at least two years' worth of experience developing computer networking or network architecture, or a Master's degree in a related field. Or, instead of the education requirements, a POSITA could have four or more years of work experience in computer networking and in the development of client-server systems. A POSITA would have been familiar with fields including operating-system design, and distributed systems.

18. I have met the criteria described above since long before 2010, so I would have been a POSITA with regard to the '721 Patent at the time of the earliest priority date Patent Owner claims for the '721 Patent.

B. Legal Standard for Prior Art

19. Counsel for the Petitioners have told me that a patent, published patent application, or other publication, must first qualify as prior art before it can be used to invalidate a patent claim.

20. Counsel have told me that a U.S. or foreign patent qualifies as prior art to a patent claim if the date of application, issuance, or publication of the prior-art patent is before the purported date of invention of the patent claim.

21. Counsel have also told me that a printed publication (like a technical publication, a magazine article, or newsgroup post) qualifies as prior art to a patent claim if the date of publication is before the purported date of invention of the patent claim. Counsel have also told me that a U.S. patent qualifies as prior art to a patent claim if the application for the prior-art patent was filed in the United States before the purported date of invention of the patent claim. Finally, Counsel have told me that, if a U.S. Patent incorporates another patent or document by reference, it is considered to be part of that same reference.

C. Claim Construction

22. Counsel have told me that in an IPR the Board uses the same standard for construing the challenged claims as during claim construction in district court civil actions. Counsel have also told me that, in the district court context and in this IPR, the words of a claim are generally given their ordinary and customary

meaning—that is, the meaning a term would have to a person of ordinary skill in the relevant art at the time of the invention. The time of invention is generally the time the application was filed, although in this case Counsel have told me that Patent Owner has asserted an invention date of May 2010. Counsel have asked me to use May 2010 for my analysis in this Declaration, so I did. But I do not necessarily agree that May 2010 is the correct priority date, and I do not mean to convey by using that date that I think it is the correct date.

23. Counsel have also told me that the specification informs the proper interpretation of a claim. This is because a POSITA is assumed to have read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification. Counsel have also told me that the prosecution history and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art are also considered in ascertaining the meaning of claim terms.

D. Legal Standard for Obviousness

24. Counsel have told me that an invention may be obvious, even if it is not identically disclosed or described in a single prior-art reference, as long as the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious to a POSITA at the time the invention was made.

25. Counsel have also told me that the Supreme Court has told the PTO to be cautious granting a patent based on the combination of elements found in prior art, because a patent for a design that only combines old elements but does not actually change their respective functions removes an already-known design from the public and gives it exclusively to the patentee. The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.

26. Counsel have also told me that when a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a POSITA can implement a predictable variation, the invention is probably obvious. For the same reason, if a technique has been used to improve one device, and a POSITA would recognize that the technique would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond the POSITA's skill. It is my understanding, in other words, that a court has to determine whether the improvement is more than the predictable use of prior-art elements according to their established functions. Counsel have told me that that this standard is typical and it is not confined to any particular formalistic conception.

27. Counsel have also told me that an obviousness analysis should consider the scope and content of the prior art, differences between the prior art and

the claims of the challenged patent claims, and the level of ordinary skill in the art. Secondary considerations of obviousness or non-obviousness, where supplied, must also be considered. Secondary considerations can include commercial success enjoyed by devices practicing the patented invention, industry praise for the patented invention, copying by others, and the existence of a long-felt but unsatisfied need for the invention. I am also informed by Counsel that there must be a nexus between these factors and the claims themselves for these secondary considerations to have an impact on the analysis of obviousness.

28. Counsel have told me that there generally must be a motivation to combine the references in order to find obviousness. This motivation or reason to combine can arise from the references themselves, changing market dynamics, or from simple common sense.

29. I am further informed by Counsel that a single reference can render a patent claim obvious, even if that reference does not explicitly or inherently disclose every element of the claim.

30. To determine whether there was an apparent reason to combine the known elements in the way a patent claims, it will often be necessary to look to interrelated teachings of multiple patents; to the effects of demands known to the design community or present in the marketplace; and to a POSITA's background knowledge.

IV. TECHNOLOGY BACKGROUND

31. In this section I offer some background on malware and malware-detection software, for the purpose of contextualizing my opinions on the validity of the claims of the '721 Patent. All of the concepts discussed in this section were well-known and widely used well prior to the May 2010 priority date I have assumed applies to the '721 Patent.

A. Overview of Malware

32. Malware, or malicious software, is software that targets information or other resources on a computer or network in order to modify, steal, or destroy the data or otherwise misuse the resources. Malware enters a computer system or network and acts on it in harmful ways. Malware may come in the form of a “virus” that inserts its own code into existing programs and may destroy, encrypt, modify, or exfiltrate data and or programs; slow system resources; or log keystrokes, screenshots, and or other user actions; and to reproduce by infecting other files and spread to other computers to repeat the process. Malware may also exist as a “worm” that can spread without attaching itself to existing files, and can delete, encrypt, or steal data or otherwise misuse resources. Worms generally do not require user input to reproduce, and they typically spread to other computers on a network. “Trojan horses” are malware programs that appear harmless or legitimate so that an end user will unwittingly grant them access to a computer or system. For example, a Trojan

horse software may appear as a regular operating-system update from Microsoft, but once installed, it may damage, disrupt, steal, infect, etc., the data or programs on a computer or network.

33. Malware enters a computer system from an external source, usually the Internet or some external storage device (e.g., external hard drive or flash drive). Malware may be hidden inside software bundles or a storage device's firmware, or it may be disguised as a recognizable program.

B. Overview of Malware-Detection Software

34. Malware detection usually involves analyzing a signature of a suspect object. A signature is a characteristic of the malware (or other object), which identifies that object with good specificity and that uses much less space than the full object itself. The signature may be extrinsic metadata (e.g., size, name, date of creation or modification, etc.) or intrinsic (certain portions of the file or code, or all of it, or a pattern somewhere within the object), or a digest derived mathematically by applying a function to some or all of the object and or its metadata. The purpose of a signature is to identify a particular object (e.g., malware, or alternatively, legitimate software from a reputable source) with high reliability. Malware-detection-software organizations compile the signatures for known malware and store them in a database. That way, the software can generate signatures for objects introduced into computer systems and compare the generated signatures to those in

the database. If the signature is found in the database, the object is classified as malware and isolated or otherwise handled.

35. This traditional method is effective against known malware, but it has drawbacks: It relies on an up-to-date, exhaustive database. Brand-new malware can compromise a computer or network before the software provider is able to update the database with the signature for the new malware. And if the signature is not in the database, the new malware will not be detected and may “infect” the computer or network. So the reliability of this traditional method depends in large part on up-to-date databases.

36. Some malware creators may also hide the attributes of their malicious code that malware-detection software could use to create its signature, allowing malware to evade detection and infect the computer or network (e.g., polymorphic malware, which changes its appearance but not its functionality when it spreads its infection). So even if a particular malware’s signature is already known and stored in the database, a variant of it that “looks” different but functions in essentially the same way may be able to evade detection and infect its target.

37. Malware-detection software may also use behavior analysis. Behavior analysis assesses a suspect object’s actual or attempted behavior (i.e., what it has done or tries to do within the computer or system if it is allowed entry). Behavior-analyzing software might allow a process to run on a user’s system, or it

may run the object in a particular environment and in either case assess whether the object's activity is malicious or benign. The particular environment may be a "sandbox" in which the process runs with restricted access privileges, perhaps on the real system, or the environment may be a simulated or emulated environment in which the suspect object can do no harm to the real system. The advantage of behavior-analysis detection is that it works for objects that have not been previously detected because it analyzes the actions an object takes or tries to take, rather than statically analyzing the object's characteristics (e.g., code or metadata).

38. Behavior-based analysis takes two general forms. One form specifies suspicious or known bad behaviors often taken by malware (e.g., changing the registry contents, modifying executable files, sending numerous emails without user input, sending emails with certain types of attachments, etc.). Often a single isolated action is insufficient to indicate malware reliably, since some of the suspicious actions may be performed by legitimate programs as well, so the system may be set to diagnose an object as malware only when it performs multiple suspicious actions. Some types of malware may be positively identified by specific sequences of actions (e.g., deleting or encrypting files) while previously unknown malware might only be identified as malicious but not associated with a known malware instance.

39. The second form of behavior analysis attempts to define “normal” or “acceptable” behavior, and requires that an uninfected system be monitored and its behaviors analyzed to arrive at a sufficiently complete collection of behaviors that benign processes exhibit. When an unknown program (or a modified version of a known program) executes, either it or the system as a whole may be monitored for behavior to determine whether or not the attempted or actual behavior is “normal.” This approach may rely upon statistical analysis, specific sequences of actions, or other means to detect anomalous behavior and to flag it. Anomaly detection thus may include acceptability parameters. If anomaly-detection software with acceptability parameters discovers that an object has performed or has attempted to perform an activity or activities outside those parameters, it may classify that object as malware. In order to avoid excessive false positives (i.e., identifying a legitimate process as malware), it is often necessary to allow for some anomalous behavior to occur without raising an alert, and so some means of setting acceptable levels of anomalous behavior must be employed in such systems.

40. The notion of “taint” was also well-known by the time the ’721 Patent was filed. Here, objects that have been written by an object believed to be malicious are considered to have lost integrity and possibly be malicious themselves. Likewise, previously benign objects that read from a malicious object lose integrity and become tainted. These concepts of information flow-based security were

formalized in the mid-1970s by the Biba model for integrity and the Bell-LaPadula model for confidentiality. *See* EX-1014; EX-1015. A POSITA working in the area of information security would have been well-familiarized with these famous models.

41. The '721 Patent incorporates by reference, though does not claim priority to, U.S. Patent Publication No. 2007/0016953 (the "'953 Publication") (Jan. 18, 2007). EX-1001, 2:15-19. The '953 Publication itself claims priority to June 30, 2005. Therefore, the '953 Publication illustrates the state of the art in 2005.

42. The '953 Publication describes the use of signatures to determine whether a file included malware. EX-1009, [0003] ("A typical anti-malware product, such as anti-virus scanning software, scans objects or the results of an algorithm applied to the object or part thereof to look for signatures in the object that are known to be indicative of the presence of a virus. Generally, the method of dealing with malware is that when new types of malware are released, for example via the Internet, these are eventually detected. Once new items of malware have been detected, then the service providers in the field generate signatures that attempt to deal with these and these signatures are then released as updates to their anti-malware programs.").

43. The '953 Publication also describes signatures that attempt to uniquely identify files, or parts of files, to servers: "[i]n the case of these prior art

systems, the request for authorisation to run the file that is sent by a local computer to the central computer may comprise sending a checksum or ‘signature’ or ‘key’ that uniquely represents the file.” EX-1009, [0007].

44. The ’953 Publication also describes the use of databases categorizing known files to determine whether a file might be malware: “WO-A-2004/097602 describes a system that analyses computer files received or generated by a local computer and compares these with a database of known files to determine whether a particular file is known and if so whether it has been known about long enough that it can be regarded as ‘safe’.” *Id.*, [0005].

45. The ’953 Publication also describes networked malware-detection systems in which local computers send data on unknown files to a central computer, which would determine whether a file was safe (if it had seen the file previously) or receive the entire file to determine whether it was safe (if it had not seen the file previously): “If the file is recognised at the central computer, then the central computer can send permission for the local computer to run the file if the file is known to be safe or send a ‘deny’ command if the file is known to be malicious. However, in each of these prior art proposals, if the file is not known at the central computer, then the whole file is sent to the central computer where it can be analysed to determine whether it should be regarded as safe or malware.” *Id.*, [0007].

46. The '953 Publication also describes keeping a count of the number of times a file has been seen: “A similar community-based anti-malware system is disclosed in WO-A-02/33525. In this system, in the case that a local computer is seeking clearance to run a file that is not known by the central computer to be safe or malware, some limited audit information *about the prevalence of the file on other local computers* can be sent to a human system administrator associated with the local computer that is seeking permission to run the file. The human system administrator can therefore make a better informed though still ‘manual’ decision as to whether or not the file is safe to run.” *Id.*, [0008] (emphasis added).

47. Likewise, “[i]n the system of US-A-2004/0073810, a metafile containing data about an attachment or other transmitted file is sent to a central computer. The data about that file is analysed to determine a likelihood of the transmitted file being malware. *A specific example given is that if the transmitted file has been transmitted via at least a certain number of servers, then it should be treated as malware.*” *Id.*, [0009] (emphasis added).

48. As the '953 Publication shows, networked malware-detection systems (including databases, signatures, counts, and comparing data from files not previously seen to data from files previously seen) were well-known at the time of the '721 Patent.

49. U.S. Patent No. 7,392,543 (“Szor”), issued June 24, 2008, is another example illustrating the state of the art at the time. Szor describes a system in which “host computer systems 104 automatically detect malicious code” and “host computer systems 104 also automatically generate and send extracted malicious code packets . . . to a local analysis center computer system,” including “the malicious code signature of the malicious code undertaking the attack.” EX-1008, 2:38-44, 2:51-52. Szor also describes sending “an extracted malicious code packet or malicious code signature of the malicious code undertaking the attack to a global analysis center,” for the purposes of “prevent[ing] the further spread of the malicious code globally” by “delivering signature updates” to client computers. *Id.* at 2:59-64.

50. One of Petitioners’ own patents, for which Petitioners applied in 2008, U.S. Patent No. 7,966,650 (“Manring”), EX-1010, describes the state of the art pertaining to many of the ’721 Patent’s claimed inventions: keeping a count of the number of times an object has been seen (*see* EX-1010, 8:14-21); observing the behavior of computer objects and collecting information about that behavior (*see id.*, 7:59-64, 8:21-28); updating a database and then distributing updated information (*see id.*, 7:27-31); and storing information about objects in databases (*see id.*, 8:61-9:1, 12:18-23).

V. THE '721 Patent

A. Overview of the '721 Patent

51. The '721 Patent correctly notes that anti-malware products (e.g., “virus scanning software”) generally assess a suspect object, or “the results of an algorithm applied” to a suspect object, by comparing its characteristics to those of known malware. EX-1001, 1:37-40. According to the '721 Patent, these products can identify malware that has been previously observed, but they are unable to detect novel malware until “service providers in the field generate signatures that attempt to deal with these [new items of malware] and these signatures are then released as updates to their anti-malware programs.” *Id.*, 1:43-47. “Thus, users are at risk from new malware for a certain period of time which might be up to a week or even more.” *Id.*, 1:53-55. Cloud computing can increase malware-identification speed to the extent that “protection is provided by signatures that are stored in the cloud, i.e., in a central server to which the remote computers are connected.” *Id.*, 1:56-60.

52. The '721 Patent incorporates by reference U.S. Patent Publication 2007/0016953 (dated January 18, 2007), in which the same applicants explained “a cloud-based approach . . . where the central server receives information about objects and their behavior on remote computers throughout the community and builds a picture of objects and their behavior seen throughout the community.” EX-1001, 2:15-25. Even then, though, the incorporated cloud-based method “involves

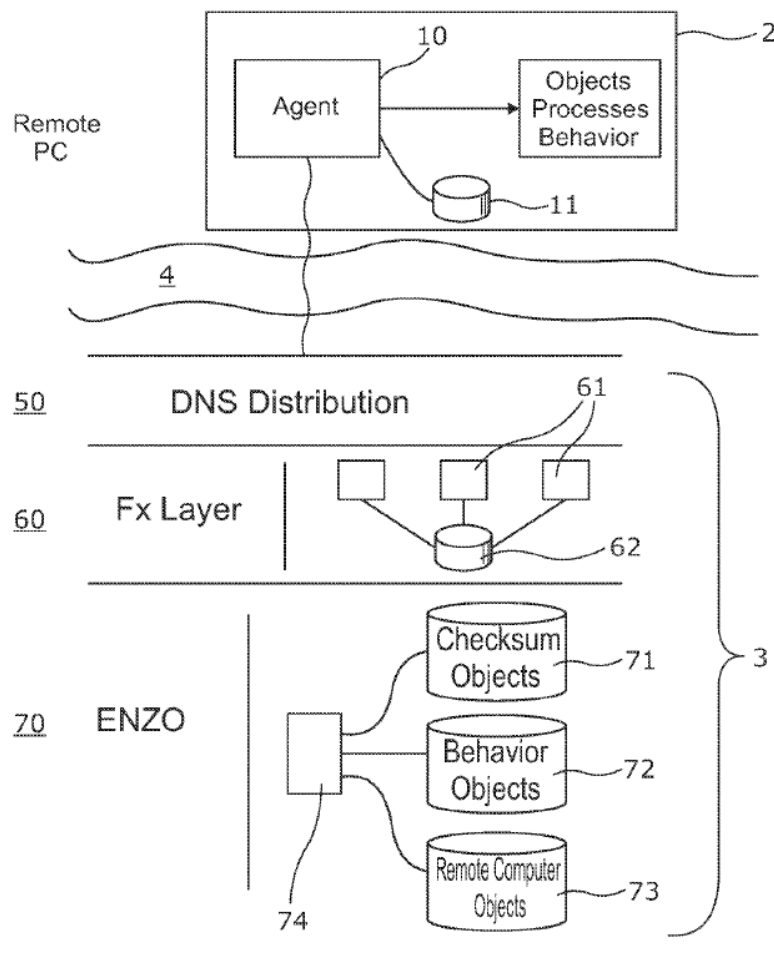
communicating, storing and managing vast amounts of data at the central server, which is a challenging problem in itself.” *Id.*, 2:29-31.

53. At the ’721 Patent’s purported invention date in 2010, networked malware detection was not novel—in fact, it was common. For one example, an earlier patent, No. 7,392,543 (“Szor”) taught a method by which “host computer systems 104 automatically detect malicious code” and “host computer systems 104 also automatically generate and send extracted malicious code packets . . . to a local analysis center computer system,” including “the malicious code signature of the malicious code undertaking the attack.” EX-1008, 2:38-44. Szor’s method also provided for the sending of “an extracted malicious code packet or malicious code signature of the malicious code undertaking the attack to a global analysis center,” for the purposes of “prevent[ing] the further spread of the malicious code globally” by “delivering signature updates” to client computers. EX-1008, 2:59-64.

54. The ’721 Patent claims that it innovates by using a distributed computer architecture that tracks new objects, analyzes information about the new objects, and updates the distributed components to account for the new information. The ’721 Patent teaches a “base computer,” EX-1001, 2:49, paired with a “plural” network of “remote computers,” *id.*, 2:50. The remote computers send data about new objects to the central base computer and the base computer applies “rules and

heuristics” to the information it receives in order to determine “whether or not the object is malware.” *Id.*, 2:49-57.

55. In the '721 Patent, a base computer includes “a first layer . . . referred to as the ‘FX layer,’” *id.*, 9:15-20, that makes a decision on the basis of “information received from the remote computer,” *id.*, 9:31-33, as to whether the object is safe, unsafe, or unknown. Then, “[s]itting behind the FX layer is a second layer (called the ‘ENZO layer’ . . .)” *id.*, 9:47-48, that maintains “a master record . . . of all data received from all remote computers,” *id.*, 9:47-51; see '721 Patent Fig. 3:



56. The Challenged Claims describe a system that manages threat data (i.e., data regarding suspect objects) in layers: It includes a “first remote computer,” a “first computer object,” a “first threat server,” a “first database,” a “second database,” a “database associated with a central server,” and a “second threat server.” This arrangement of layered threat assessment and threat-data management was not novel at the time of the ’721 Patent; in fact, it was common, and many such systems were able to track a count of how often an object was seen, as well.

57. In my opinion the Challenged Claims of the ’721 Patent would have been obvious over the prior-art references Liu, Tuvell, and Kester.

B. Overview of the Challenged Claims

58. I am informed by counsel that Petitioners are challenging two independent claims of the ’721 Patent (Claims 1 and 9) and three dependent claims (Claims 8, 14, and 15). I have analyzed each of these claims reproduced below.

59. For the reasons I detail below, it is my opinion that each of these claims is invalid as obvious over the prior art discussed below.

Independent Claim 1
[1P]: A method of classifying a computer object as malware, the method comprising:
[1A]: receiving, at a first threat server, details of a first computer object from a first remote computer, wherein the details of the first computer object include data uniquely identifying the first computer object;
[1B]: determining, by the first threat server, whether the first computer object has been previously seen by comparing the data uniquely identifying the first computer object to a

plurality of data uniquely identifying plural computer objects in a first database associated with the first threat server;
[1C]: receiving additional information about the first computer object from the first remote computer when the first computer object has not been previously seen;
[1D]: storing the details of the first computer object and the received additional information about the first computer object in a second database associated with the first threat server when the first computer object has not been previously seen;
[1E]: providing contents of the second database to at least one database associated with a central server, wherein the contents comprise a signature of the first computer object, behavior information about the first computer object, and information about the first remote computer;
[1F]: increasing a count associated with a number of times that the first computer object has been seen, and providing the increased count associated with the number of times that the first computer object has been seen to the central server; and
[1G]: receiving, at a second threat server, at least a portion of the contents of the at least one database associated with the central server, wherein the at least a portion of the contents of the at least one database associated with the central server include a subset of the details of the first computer object stored in the second database.
Dependent Claim 8
[8A]: receiving, at the second threat server, details of a second computer object from a second remote computer, wherein the details of the second computer object include data uniquely identifying the second computer object;
[8B]: determining, by the second threat server, whether the second computer object has been previously seen by comparing the data uniquely identifying the second computer object to a plurality of data uniquely identifying plural computer objects in a third database associated with the second threat server;
[8C]: determining that the second computer object has been seen before;
[8D]: increasing a count associated with a number of times that the second computer object has been seen and providing the increased count associated with the number of times that the second computer object has been seen to the at least one central server; and
[8E]: receiving, at the first threat server, a count associated with the number of times that the second computer object has been seen.
Independent Claim 9
[9P]: A system for classifying a computer object as malware, the system comprising:
[9A]: a first threat server arranged to receive details of a computer object from a first remote computer, wherein the details of the first computer object include data uniquely identifying the first computer object;
[9B]: wherein the first threat server is further arranged to receive the details of the computer object from the first remote computer and determine whether the first computer object has been

previously seen by comparing the data uniquely identifying the first computer object to a plurality of data uniquely identifying plural computer objects in a first database associated with the first threat server;
[9C]: wherein the first threat server is further arranged to receive additional information about the first computer object from the first remote computer when the first computer object has not been previously seen;
[9D]: store the details of the first computer object and the received additional information about the first computer object in a second database associated with the first threat server when the first computer object has not been previously seen;
[9E]: provide contents of the second database to at least one database associated with a central server wherein the contents comprise a signature of the first computer object, behavior information about the first computer object, and information about the first remote computer;
[9F]: and increase a count associated with a number of times that the first computer object has been seen; the central server arranged to receive the increased count associated with the number of times that the first computer object has been seen; and
[9G]: a second threat server arranged to receive at least a portion of the contents of the at least one database associated with the central server, wherein the at least a portion of the contents of the at least one database associated with the central server include a subset of the details of the first computer object stored in the second database.
Dependent Claim 14
[14P]: The system according to claim 9, wherein the central server comprises:
[14A]: a) an object database storing object signatures and metadata about objects;
[14B]: b) a behavior database storing object behavior information; and
[14C]: c) a computer-object database storing information about what objects are present on what remote computers.
Dependent Claim 15
[15]: The system according to claim 14, wherein the first and second threat servers and the central server are implemented using cloud computing.

VI. PRIOR ART CONSIDERED

60. I provide below prior-art references with exemplary citations identifying the relevant features related to the claim language of the Challenged Claims. I have also described combinations of these prior-art references that, when

combined, render the Challenged Claims of the '721 Patent obvious. In my opinion the Challenged Claims of the '721 Patent are rendered obvious in light of the prior art specifically discussed below.

61. The citations presented in the body of this declaration correspond to the teachings disclosed in exemplary items of prior art that identify: (1) the problem(s) confronted by one of ordinary skill in the art; (2) an express suggestion to make one or more of the combinations; (3) an implicit suggestion to make one or more of the combinations; or (4) the knowledge of those skilled in the art as to the applicable field of technology at the relevant timeframe.

A. Identification of the Primary Prior-Art References

62. It is my opinion that a person of ordinary skill would consider at least the prior-art references below:

Prior-Art Reference	Priority Date / Date of Publication	Exhibit No.
U.S. Patent Pub. No. 2006/0075504 to Liu ("Liu")	Apr. 6, 2006	1005
U.S. Patent Pub. No. 2008/0086773 to Tuvell ("Tuvell")	Apr. 10, 2008	1006
U.S. Patent Pub. No. 2005/0223001 to Kester ("Kester")	Oct. 6, 2005	1007

In addition, a POSITA would have been familiar with standard problems, techniques, methods, architectures, and systems of the time. These would have included the following books and reports:

Secondary Source	Exhibit No.
HENRY F. KORTH & ABRAHAM SILBERSCHATZ, DATABASE SYSTEM CONCEPTS, McGraw-Hill, New York (6th ed. 2006).	1013 (excerpt)
K. J. Biba, <i>Integrity Considerations for Secure Computer Systems</i> , MTR-3153, The MITRE Corporation (Apr. 1977).	1014
D. Elliot Bell & Leonard J. LaPadula, <i>Secure Computer Systems: Mathematical Foundations</i> , MTR-2547, The MITRE Corporation (Mar. 1973).	1015

B. General Overview of the Primary Prior-Art References

63. The following sections provide a brief description of the anticipating prior-art systems that invalidate the '721 Patent. This discussion is not meant to be exhaustive, and my full element-by-element analysis is provided in more detail below (see VII. ANALYSIS OF THE CHALLENGED CLAIMS AGAINST THE PRIOR ART).

64. As mentioned before, it is my opinion that the references summarized below disclose the features of the Challenged Claims of the '721 Patent.

1. Liu

65. Liu's "threat protection networks" distribute "updated threat definition data . . . using a secure peer-to-peer network," EX-1005, [0005]. Liu

discloses “at least one client computer” and “a server that stores threat definition data and is connected to the network.” *Id.*, [0006]. In the Liu process “clients generate threat signatures and compare them [with] threat definition data. In the event that a potential threat is unknown, a client can refer the potential threat to a server.” *Id.*, [0005]. “[T]he server compares the potential threat signature against its own copy of the threat definition data. When the server has no information concerning the nature of the potential threat, then the server can refer the potential threat to an expert system.” *Id.*, [0005]. An “expert system” makes determinations regarding new threats, and the server can store and update threat-definition data on the basis of those determinations. *Id.*, [0009]. The server can send the updated information to client computers distributed across a “secure peer-to-peer network,” *id.*, [0011], and the system can identify potential threats on the basis of a unique signature generated from the “header” or “body” of a suspect file. *Id.*, [0021]-[0023].

2. Tuvell

66. Tuvell discloses “a network management system [that] monitors malware within a mobile network.” EX-1006, Abstract. Tuvell’s “collection agents” determine whether an executable file on a mobile device may be malware. *Id.*, [0089]. The collection agents can also “notif[y] the sample collection center 1112 of the number of times it has seen the executable.” *Id.*

3. Kester

67. Kester also describes a method for protecting computers from malware. The method in Kester includes storing information identifying malware in a database associated with a computer, EX-1007, [0042], and providing the contents stored in that database to a central database factory, *id.*, [0054].

C. Motivation to Combine and Reasonable Expectation of Success

68. A POSITA would have known to combine the disclosures of Liu, Tuvell, and Kester, because a POSITA would have known that doing so would facilitate Liu's implementation and lead to greater success in identifying (and therefore in stopping) malware.

69. Liu, Tuvell, and Kester all apply to the same field: threat detection that can identify threats and distribute the results throughout the network. In Liu, "threat protection networks" share "threat definition data" along "a secure peer-to-peer network." EX-1005, [0005]. Tuvell's "network management system" "monitors" and "analy[zes] malware," EX-1006, Abstract, and Kester discloses a system "adapt[ed] . . . over the internet" that "protects computers from malicious software programs" by "identifying" them and "determining whether the malicious software program[s] . . . ha[ve] been previously analyzed." EX-1007, Claim 1. A POSITA would have understood that the processes described in Liu, Tuvell, and

Kester are all in the same field of endeavor, and are analogous to the claimed invention of the '721 Patent.

70. A POSITA would also have known that the disclosures of Liu, Tuvell, and Kester supplement and elaborate on one another. For instance, a POSITA would have recognized that it would be useful to consider detailed storage methodologies for the results of a malware-identification and malware-classification process. Liu and Kester would have been natural places to look—Liu teaches storing data related to the malware-detection process, and Kester describes a method for storing the artifacts and results of a malware-detection process.

71. A POSITA would also have known to combine Liu, Tuvell, and Kester because a POSITA would have known that doing so would predictably yield beneficial results, for instance: implementation over different types of architecture, scalability, availability, efficiency, reliability, and the likely outcome of identifying (and thus protecting systems from) more malware than any individual prior-art system alone.

72. Liu's "threat protection networks" include "at least one client computer" and a network server that "stores threat definition data." EX-1005, [0005]-[0006]. In Liu, "threat signatures" generated on the client computer are compared to "threat definition data," and the system can refer any unknown threat to a server. *Id.*, [0005]. The server is then able to update the threat-definition data

it stores based on whatever determinations its “expert system” makes as to whether an object is malware. *Id.*, [0009]. And it can distribute its updated information to client computers at distributed network locations. *Id.*, [0011].

73. In Tuvell, the system disclosed “collect[s] data from distributed locations,” EX-1006, [0011], and is able to count “the number of times it has seen the executable,” *id.*, [0089]; *see id.*, Fig. 11. Because modifying Liu in view of Tuvell would clarify the scale of the possible distribution of clients within a network and increase the odds that the system would successfully detect malware, and because a POSITA would have been familiar with the use of Simple Network Management Protocol (SNMP) and its Management Information Bases (MIBs) used to manage network assets, in part by counting events and items, a POSITA would have known to modify Liu in view of Tuvell.

74. Kester’s method discloses storing malware-related data in databases. In Kester, an event may be “logged in the logging database 206.” EX-1007, [0042]. For instance, “the application name, time of day, and the hash associated with the application.” *Id.*; *see id.*, Claim 1. Kester also determines whether or not an application has been classified/categorized, and treats uncategorized applications differently, including storing “collection data” about them in special databases. *Id.*, [0049], [0052]. Because modifying Liu in view of Tuvell and Kester would further elaborate the method of storage for threat-definition data, and offered possible

benefits of efficiency and performance, a POSITA would have known to modify Liu and Tuvell in view of Kester.

75. Therefore, a POSITA would have known to use counting and distribution (as Tuvell teaches) and database storage (as Kester teaches) when implementing Liu's malware-protection system.

VII. ANALYSIS OF THE CHALLENGED CLAIMS AGAINST THE PRIOR ART

A. Claim Construction

76. I apply the plain and ordinary meaning for each term below, as they would have been understood by a POSITA at the priority date of the patent.

77. Where I address the preamble of a claim, I am not opining regarding whether the preamble imposes limitations on the claim's scope as a matter of claim construction. Instead I am opining that, to the extent the preamble is limiting, it is disclosed for the reasons I describe below.

B. Application of the Prior Art to the Claims

78. Based upon my review of the prior art, it is my opinion that the subject matter of each of the Challenged Claims would have been obvious to one of ordinary skill in the art in view of the known prior-art references.

79. I am informed by Counsel that each of the prior-art references analyzed in this declaration meets the legal requirements for prior art to the '721 Patent. I have performed my analysis with this understanding.

80. Based upon my analysis, it is my opinion that the Challenged Claims of the '721 Patent are rendered obvious by the following prior art:

Ground	Claims	Opinions
1	1, 8, 9, 14, and 15	§ 103 by Liu in view of Tuvell and Kester

C. Ground 1: Liu in View of Tuvell and Kester Renders Obvious Claims 1, 8, 9, 14, and 15

81. In my opinion, Claims 1, 8, 9, 14, and 15 of the '721 Patent are obvious over Liu in view of Tuvell and Kester.

1. Independent Claim 1

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

82. In my opinion, to the extent that the preamble of the '721 Patent is claim-limiting, Liu teaches [1P]. The '721 Patent's claimed "computer object" refers to malware, which it defines as follows: "The term 'malware' is used herein to refer generally to any executable computer file or, more generally 'object', that is or contains malicious code, and thus includes viruses, Trojans, worms, spyware, adware, etc. and the like." EX-1001, 1:33-36. The '721 Patent calls its base server a "threat server" and hence equates malware to threats. EX-1001, Claim 1.

Analogously, Liu discloses “threats,” which, as Liu explains, include “computer virus[es],” “spam,” “spyware,” “adware,” “stealware,” “hijacking,” and “automated hacking.” EX-1005, [0003], [0065]. For example, Liu correctly notes that “Computer viruses are programs that typically seek to . . . modify and/or damage the [h]ost computer system,” *id.*, [0003], and that there may be “check sums generated using a file associated with the potential threat,” *id.*, [0008]. So Liu also equates viruses and other malware with threats.

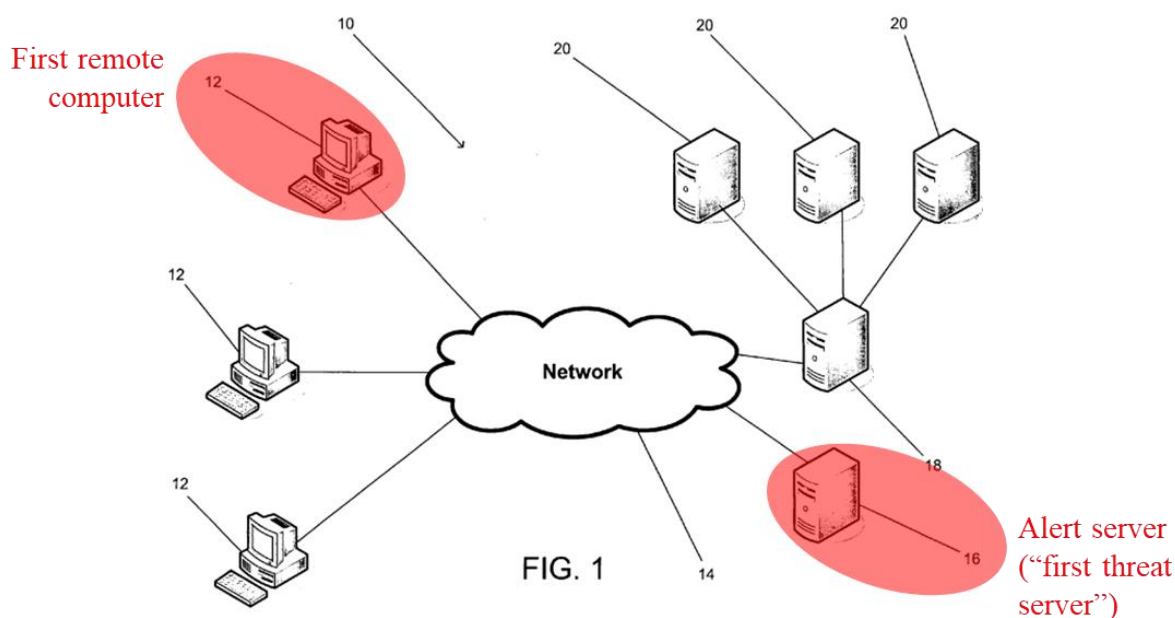
83. Liu also discloses “classifying a computer object as malware.” For instance, in Liu a “threat score” can be generated, and the system in Liu can use the threat score to determine “whether the potential threat is an actual threat.” EX-1005, [0091] (“The threat score can also enable a determination (148) of whether a potential threat is in fact safe.”), [0093] (“In many embodiments, a determination that a previously unknown potential threat is an actual threat is indicated by a threat score exceeding a predetermined threshold. In addition, a determination that the potential threat is safe is indicated by a threat score below a second threshold.”), [0094] (“an expert system that classifies potential threats automatically using a threat score”).

[1A]: receiving, at a first threat server, details of a first computer object from a first remote computer, wherein the details of the first computer object include data uniquely identifying the first computer object;

84. Liu discloses [1A].

85. In the '721 Patent, a network of computers (PCs, cell phones, internet-connected appliances, to name a few examples) connect with a hub computer via an Internet network. The connected devices are all able to load some sort of computer object, which is any packet of code that uses the computer to operate. EX-1001, 3:63-4:5 (“Plural local or ‘remote’ computers 2 are connected via the Internet 1 to a ‘central’ or ‘base’ computer 3,” and that remote computers may be “a personal computer, a server of any type, a PDA, mobile phone, an interactive television, or any other device capable of loading and operating computer objects” where “[a]n object in this sense may be a computer file, part of a file or a sub-program, macro, web page, or any other piece of code to be operated by or on the computer, or any other event whether executed, emulated, simulated or interpreted.”)

86. **“first remote computer”**: In Liu, “[e]mbodiments of threat protection network ... use expert systems to determine the nature of potential threats to a remote computer.” EX-1005, Abstract. Liu also notes that “[n]etworks such as the Internet can enable rapid communication of information between computers. Unfortunately, the capability of computers to communicate is often used to victimize computer systems and/or their users.” EX-1005, [0003]. In Liu’s Fig. 1 (see below), the Liu computer 12 is an example of the claimed “first remote computer.”



87. A POSITA would have known that Liu’s computer 12 is any computer that uses the Internet to communicate with other computers and that is therefore potentially vulnerable to threats. For example, Liu discloses that “[c]lient software is installed on each of the computers 12 and, during operation, the computers execute the client software as a background process.” EX-1005, [0067], [0067]. And because Liu also explains that the client software may include web-browser or email plugins, *id.*, a POSITA would have understood that Liu’s computer could at least be any computer that had a web browser or email program, (e.g., a personal computer, PDA, or mobile phone), and is the claimed “first remote computer,” just as is claimed in the ’721 Patent.

88. **“first threat server”**: In the ’721 Patent a “threat server” may be “any type of suitable server.” EX-1001, 9:16-23. Liu discloses an alert server 16 (which is an example of the claimed “first threat server”), EX-1005, [0066]-[0068], and “the functions of the alert server can be performed by one or more servers, *id.*, [0101]. A POSITA would have understood that Liu’s alert server could be comprised of any server(s) and is the claimed “first threat server.”

89. **“receiving, at a first threat server, details of a first computer object from a first remote computer”**: In Liu, the system can receive details of a potential threat from computer 12, at an alert server. And “[w]hen the client software [on computer 12] encounters an unknown potential threat, the client can contact the alert server 16 for information concerning the potential threat. In the event that no information is known about the potential threat, then the client can refer the potential threat to the alert server 16.” EX-1005, [0067]; *see id.*, [0068] (“The alert server 16 can possess information concerning the nature of a potential threat and provide the information concerning the nature of the threat directly to the computer 12 that referred the potential threat to the alert server.”).

90. Moreover, in Liu, contacting the alert server for information about the potential threat includes sending a signature of the potential threat to the alert server. EX-1005, [0080] (“The process 80 commences with the determination (82) of a potential threat. The signature of the potential threat is sent (84) to an alert

server in order to obtain additional information about the threat.”); *see id.*, Fig. 4. Liu’s signature of the potential threat is an example of the claimed “details of a first computer object.” *See* EX-1005, [0083] (“The process 110 commences with the receipt (112) of a threat signature of a potential threat from a client.”). Therefore, a POSITA would have understood that Liu discloses “receiving, at a first threat server, details of a first computer object from a first remote computer.”

91. **“wherein the details of the first computer object include data uniquely identifying the first computer object”**: In Liu, “[a] signature is one or more pieces of information that attempt to uniquely identify the source.” EX-1005, [0072]; *see id.*, [0004] (“Often, threat signatures are used to identify threats. A threat signature is a characteristic of a threat that is unique and, therefore, distinguishes the threat from other potentially benign files or computer programs.”). Therefore, in my view a POSITA would have known that the “signature” in Liu uniquely identifying a potential threat, which the alert server receives from the client computer, is the “data uniquely identifying the first computer object” claimed by ’721.

[1B]: determining, by the first threat server, whether the first computer object has been previously seen by comparing the data uniquely identifying the first computer object to a plurality of data uniquely identifying plural computer objects in a first database associated with the first threat server;

92. Liu discloses [1B].

93. **“determining, by the first threat server, whether the first computer object has been previously seen”**: Liu teaches that the alert server “compares (114) the threat signature to the threat definition data.” EX-1005, [0083]. Also, Liu states that “the client can inspect the return code to ascertain (96) whether the potential threat has been reported by another client.” EX-1005, [0082]. There are three potential outcomes of this comparison in Liu: (1) the server sends a return code indicating the nature of the potential threat, if the threat-definition data already lists its threat signature, *id.*; (2) if the object’s threat signature is not already listed in the threat-definition data, then the system determines whether another client has contacted the alert server to report the threat signature. If so, a return code indicating the prior report and the object’s still-unknown nature can be sent, *id.*, [0084]; (3) if the server has not received any earlier reports of the threat signature, then a return code including a request for a copy of the potential threat can be sent, *id.*, [0085] (“In the event that the server has not received any previous reports relating to the threat signature, the server can send (122) a return code that includes a request for a copy of the potential threat.”); see *id.*, [0005] (“In the event that a potential threat is unknown, a client can refer the potential threat to a server. In several embodiments, the server compares the potential threat signature against its own copy of the threat definition data.”). A POSITA would therefore have understood that understood that Liu’s comparison method is the claimed “determining, by the first threat server,

whether the first computer object has been previously,” because Liu’s alert-server comparison generates different results depending on whether the threat signature has been seen before.

94. **“by comparing the data uniquely identifying the first computer object to a plurality of data uniquely identifying plural computer objects in a first database associated with the first threat server”**: Liu describes comparing an object’s threat signature to the threat-definition data (as noted above). And as explained in [1A], the threat signature is an example of the claimed “data uniquely identifying the first computer object.”

95. Liu explains, moreover, that threat-definition data includes the threat signatures of multiple (that is, “a plurality of data” concerning “plural”) potential threats. EX-1005, [0083] (“The server compares (114) the threat signature to the threat definition data. If the threat signature is listed in the threat definition data . . . then the server sends (116) a return code that indicates the nature of the potential threat.”).

96. Therefore, in Liu, the threat-definition data is the claimed “plurality of data uniquely identifying plural computer objects in a first database associated with the first threat server.”

97. Liu discloses “a server that stores threat definition data.” EX-1005, Abstract. A POSITA would have known that the server storing threat-definition data

would use a database to store it. First, Liu describes certain characteristics of the threat-definition-data storage that make it clear that Liu is referring to databases. For instance, in Liu the threat-definition data is searchable. *See, e.g.*, EX-1005, [0083] (describing determining whether a threat signature can be found in the threat definition data). The threat-definition data in Liu is also updateable. *See, e.g.*, EX-1005, [0086] (describing a method of updating threat-definition data: adding information to the previous version of the data).

98. A POSITA would have known that a data store that could be searched and updated was a database. Databases were commonly known and widely used when the '721 Patent was allegedly invented. Therefore, a POSITA would have known that the threat-definition data would have been stored in a database, which is the claimed “plurality of data uniquely identifying plural computer objects in a first database.” Moreover, it would have been obvious to any POSITA to store such searchable data in a first database.

99. In fact, the '721 Patent does not elaborate significantly on the issue of database implementation—it assumes the reader’s knowledge regarding database implementation. A POSITA would have known that the threat-definition data would have been stored in a database, which is the claimed “plurality of data uniquely identifying plural computer objects in a first database.” A POSITA would at least have found it obvious to store searchable data in a database.

100. Liu also discloses that the database storing the threat-definition data is located on the alert server. EX-1005, [0006] (“One embodiment of the invention includes at least one client computer connected to the network, a server that stores threat definition data and is connected to the network, an expert system in communication with the server.”), EX-1005, [0025] (“In a further additional embodiment again, threat definition data stored on the server, a list stored on the server and verification information stored on the server.”). Therefore, Liu teaches the claimed “first database associated with the first threat server.”

101. Kester also discloses storing the threat-definition data in a database. In Kester, “If the requested application is found in the hash/policy table 204, the event is logged in the logging database 206. Information that is logged in the logging database 206 can include, for example, the application name, time of day, and the hash associated with the application. The logging database 206 can also include additional data associated with the application requested. For example, a request frequency or a time of execution for the application requested can be included in the logging database 206.” EX-1007, [0042]. Therefore, Liu in view of Kester likewise discloses the storage of threat-definition data in a database.

[1C]: receiving additional information about the first computer object from the first remote computer when the first computer object has not been previously seen;

102. Liu discloses [1C].

103. In Liu, if the alert server has not yet seen the potential threat, as noted above, the server can request a copy of the potential threat. EX-1005, [0085] (“In the event that the server has not received any previous reports relating to the threat signature, the server can send (122) a return code that includes a request for a copy of the potential threat.”); *see also id.*, Fig. 4; *see supra* [1B]. The computer then sends a copy of the potential threat to the alert server. EX-1005, [0086] (“When a copy of the potential threat is received (124) by the server”); *see also id.*, Fig. 5, [0082] (“In many embodiments, the copy of the potential threat is provided to the alert server, which then forwards the copy to the threat identification system.”).

104. The copy of the threat sent to the alert server is one example of the claimed “additional information about the first computer object.” A POSITA would have understood that the Liu’s process of requesting, sending, and receiving a copy of the potential threat at the alert server and threat-identification system is the “receiving additional information about the first computer object from the first remote computer when the first computer object has not been previously seen” as claimed by ’721.

[1D]: storing the details of the first computer object and the received additional information about the first computer object in a second database associated

with the first threat server when the first computer object has not been previously seen;

105. Liu discloses [1D].

106. **“storing the details of the first computer object”**: As demonstrated above in [1A], the threat signature of a potential threat in Liu is “details of the first computer object.” “In many embodiments, the copy of the potential threat is provided to the alert server, which then forwards the copy to the threat identification system.” EX-1005, [0082]. “The threat identification system generates a report concerning the potential threat, which is received (126) by the server, and the report is used by the server to create a new version of the threat definition data.” *Id.*, [0086]. “The server can then make (128) the new version of the threat definition data available for download.” *Id.*

107. Moreover, a POSITA would have known that Liu’s “mak[ing] the new version of the threat definition data available for download,” *id.*, and Liu’s “process that can be used by a server to respond to a request for information concerning a potential threat,” *id.*, [0083], disclose storing details of the potential threat. The process described by Liu includes receiving the threat signature at the server, and then, as discussed, “[t]he server compares (114) the threat signature.” EX-1005, [0083]. A POSITA would have known that this process (receiving the

threat signature and then comparing it) would require storing the potential threat, so that it could be compared.

108. **“storing . . . the received additional information about the first computer object”**: As explained above (*see* [1C] *supra*), Liu’s copy of the potential threat is an example of the “received additional information about the first computer object” claimed in the ’721 Patent. The process in Liu “can be used by a server to respond to a request for information concerning a potential threat made by a client.” EX-1005, [0083]. A POSITA would have known that the Liu process of receiving additional information discloses storing a copy of the potential threat. In Liu, as noted above, if the server has not seen the threat signature before, then “the server can send (122) a return code that includes a request for a copy of the potential threat.” EX-1005, [0085]. And “[w]hen a copy of the potential threat is received (124) by the server, the server forwards the copy to the threat identification system,” which then “generates a report . . . used by the server to create a new version of the threat definition data.” *Id.*, [0086]. And “[t]he server can then make (128) the new version of the threat definition data available for download.” *Id.* A POSITA would have known that this process of receiving and forwarding would involve storing the copy of the potential threat, which is the claimed “storing . . . the received additional information about the first computer object.”

109. **“storing . . . in a second database associated with the first threat server”**: In Liu, “the server forwards the copy to the threat identification system. The threat identification system generates a report concerning the potential threat.” EX-1005, [0086]. The report generated includes content based on the object’s actions. *Id.*, [0087]-[0091]. The object-behavior report is sent to the alert server, and the alert server “create[s] a new version of the threat definition data.” *Id.*, [0086]. And “the new version of the threat definition data is created by taking the previous version and adding the threat signature to either the white list or black list depending upon the outcome of the report provided by the threat identification system.” EX-1005, [0086].

110. Whether or not data concerning unknown threats and data concerning known threats or benign objects should be stored in different databases is a matter of database design and implementation (see, for example, EX-1013, 826-30 for a discussion of various reasons for database partitioning). A single database storing information about all three types of object can easily be decomposed into three or more separate databases storing data about a subset of the objects. Likewise, an implementation that uses two databases can be transformed trivially into an implementation that uses just one by merging the databases appropriately. Moreover, Kester notes that “[a]lternatively, the uncategorized application database 108 can be a separate database from the application inventory

database 103.” EX-1007, [0049]. The “uncategorized application database” is a database at the “Application Server Module,” i.e., the first threat server in Kester, holding the “received additional information about the first computer object” claimed in the ’721 Patent.

111. The process in Liu can identify which potential threat’s signature is associated with the report produced by the threat identification system, and a POSITA would have known that therefore Liu’s system would require a method of connecting the signature and the report. Because the system generates the report based on a copy of the threat, a POSITA would have known that the system could link the signature of the threat with the copy of the potential threat. In fact, it would be necessary to provide the system with the ability to recover the potential threat object in order to perform further examination or testing so that it could be appropriately classified. A POSITA would have known about the option to store the additional data in the database holding unknown objects, as a matter of convenience, and to use the signature as a search key, as was done with the other databases.

112. As explained above, a POSITA would have known that databases were common and widespread methods of storing and connecting information at the time of the ’721 Patent’s alleged invention. The question of whether or not to hold a copy of the unknown threat object in a database at the alert server is also a matter of distributed system design. For the sake of reliability, this would often be the case.

And in Liu, the “threat signature may be listed in the threat definition data,” either “indicated as safe or as a threat.” EX-1005, [0083]. Thus, because the threat-definition data includes signatures that have been classified either as a threat or as safe, a POSITA would also have known that the potential threat’s signature and the copy of the potential threat would not need to be stored in the same database as the threat-definition data. Therefore, a POSITA would have known that the data that has not yet been classified, along with the copy of the threat, would be stored in a different and second database.

113. Kester also discloses the storage of yet-to-be-classified data in a second database. In Kester, “if the malicious software program is not identified in the first database, adding an identifier indicative of the malicious software to a second database,” EX-1007, Claim 1. Kester also discloses that “determining whether the malicious software program associated with the identifier [i.e., the signature] has been previously analyzed,” and “for each identifier that was not previously analyzed, associating the identifier with a digital fingerprint” and “adding the digital fingerprint to a third database.” *Id.* at Claim 1.

114. In the Kester method, the “uncategorized application database 108,” which stores data related to yet-to-be-classified objects, “can be a *separate database* from the application inventory database 103,” EX-1007, [0049] (emphasis added). A POSITA would have understood that by definition a *separate* database for yet-to-

be-classified data is a *second* database for yet-to-be-classified data. Moreover, the method of Kester may include many databases, over and above two. For example, “if the malicious software program is not identified in the third database, adding an identifier indicative of the malicious software program to a fourth database,” *id.*, Claim 9.

115. A POSITA would have known that Kester’s disclosure of a second, third, and fourth database further disclosed that unclassified object data could be stored in a “different” database. This also reinforces the discussion above concerning database design choices, which were well-known at the time. A POSITA would have known that modifying Liu to explicitly include storage in a second database was a known technique to yield the predictable result of dividing storage and retrieval responsibilities with regard to yet-to-be-classified data between multiple databases. Predictable advantages of doing so include improved performance and less duplicated data. EX-1013, 826-30.

116. **“when the first computer object has not been previously seen”:**
As demonstrated above in [1C], in Liu the copy of the potential threat is sent and received only when the system has not previously seen the computer object. Liu discloses that “[i]n the event that the server has not received any previous reports relating to the threat signature, the server can send (122) a return code that includes a request for a copy of the potential threat,” and “[w]hen a copy of the potential

threat is received (124) by the server, the server forwards the copy to the threat identification system. The threat identification system generates a report concerning the potential threat, which is received (126) by the server, and the report is used by the server to create a new version of the threat definition data.” EX-1005, [0085]-[0086]. And as demonstrated above, the creation of a report by the threat-identification system happens when the alert server has not seen the computer object before. Therefore, a POSITA would have known that all of the preceding steps of [1D] occur in the case that the alert server has not previously seen the computer object.

[1E]: providing contents of the second database to at least one database associated with a central server, wherein the contents comprise a signature of the first computer object, behavior information about the first computer object, and information about the first remote computer;

117. Liu and Kester disclose [1E].

118. **“wherein the contents comprise a signature of the first computer object”**: Liu discloses that the second database contains “a signature of the first computer object”: As demonstrated in [1D] above, in Liu the second database stores both a signature of the potential threat and a copy of the potential threat. *See also* [1C] *supra* (quoting EX-1005, [0085] (“In the event that the server has not received

any previous reports relating to the threat signature, the server can send (122) a return code that includes a request for a copy of the potential threat.”).

119. **“wherein the contents comprise . . . behavior information about the first computer object”**: Liu also discloses that the second database contains “behavior information about the first computer object.” As shown in [1D], the report created by the threat-identification system includes the “behavior information about the first computer object.” And a POSITA would have known that the report generated would have to be associated or connected in some way with the threat signature so that the updated threat-definition data could be created, and therefore a POSITA would have known that the report and the signature would be stored in the same database. Thus, a POSITA would have known that the report would be stored in the second database.

120. Additionally, Liu discloses that potential threats are run on the test computers over a variety of configurations (e.g., “different operating systems” and “different versions of the same operating system”), EX-1005, [0088], and a POSITA would have known that the process of exposing different configurations to a threat would have required communicating those configurations in the threat-definition data and threat report. A POSITA would have known that those configurations were information about the remote computer. Therefore, a POSITA would have known

that the threat definition data, stored in the second database, includes the claimed “information about the first remote computer.”

121. **“wherein the contents comprise . . . information about the first remote computer”**: Liu in view of Kester also discloses “information about the first remote computer.” In Liu, “[e]xamples of suspicious events include the execution of a new program or a change in the settings of the operating system, such as an attempt to add a new program to the start up folder of a computer running the Windows® operating system manufactured by Microsoft Corporation of Redmond, Wash.” EX-1005, [0076]. Event information like execution of a new program or a change in the settings of an operating system on the first remote computer would, by definition, include information about the first remote computer. Furthermore, “[t]he set of suspicious events can include any type of activity that is characteristic of any type of threat,” *id.*, and “the server can provide other types of information relating to the potential threat,” *id.*, [0080]. A POSITA would have understood that “other types of information relating to the potential threat” would also necessarily contain information about the computer housing the threat.

122. Moreover, Tuvell discloses a “receiver component” that “receives malware data,” which “includes a reference to particular devices infected or affected by the malware.” EX-1006, [0025]; *see also id.*, [0035] “[M]alware data” may “include device specific information such as current device hardware, software

and/or an identifier for the infected [] device.” *Id.* And Kester notes that the system tracks the remote computers on which an application has been found, and all the applications on each computer through the application inventory DB at each remote workstation, each Application Server Module, and at the Application Database Factory. EX-1007, [0049], [0056]. A POSITA would understand that Liu, Tuvell, and Kester disclose the claimed “information about the first remote computer.”

123. Additionally, a POSITA would have been motivated to combine Liu, Tuvell, and Kester in this way. A POSITA would have known that it was customary to store information about the device on which a threat was found, because doing so facilitates root-cause analysis, system awareness, and the ability to remedy a problem—especially in a network environment with multiple devices. A POSITA would have known, therefore, that combining Liu and Tuvell to disclose the storage of details of the devices on which the object is found would provide context that would make it easier to identify and solve problems.

124. As demonstrated above in [1D], “[t]he threat identification system generates a report concerning the potential threat, which is received (126) by the [alert] server.” EX-1005, [0086]. And, as further demonstrated, the report must be linked to the potential threat’s signature and to the copy of the potential threat, so that the system can know which potential threat’s signature is associated with the report. And a POSITA would have known that the report would itself need to be

linked to the threat signature and the copy of the potential threat in order for the system to create the new version of the threat-definition data. Therefore, a POSITA would have known that the report would be stored in the same database as the signature and copy of the potential threat. And because the signature and copy of the potential threat are stored in a second database, a POSITA would therefore have understood that the report is stored in the second database as well. As noted above, choices regarding how many databases to use to store data concerning an object is a matter of database design, with well-known trade-offs. It makes sense to collect all of the information concerning an object in a database holding information about that type of object, especially if this does not lead to unnecessary duplication of data, which in this case it would not.

125. The report includes the “behavior information about the first computer object” claimed by the ’721 Patent. The system in Liu generates the report by exposing test computers to the potential threat, after which “the behavior of the computers is observed.” EX-1005, [0087]-[0092]; *see also id.*, Fig. 6. The threat report provided by the Threat Identification Server includes information revealed by the object’s behavior. *Id.*, [0095]. Liu also notes that “a single client may be harmed,” and that this can be “communicated to other clients enabling them to protect against similar exposure and harm.” *Id.*, [0075]. In other words, Liu discloses that the harmful actions—the *behavior*—of a computer object on one client

is communicated to other client, which logically occurs through the mechanisms already disclosed in Liu (i.e., the threat definition data and report). Finally, Liu discloses that object signatures can include the source, such as an email address or a website, *id.*, [0078]. This source information constitutes another form of behavior, in that the object was sent, via email, from this address, for instance. For these reasons a POSITA would have understood that Liu discloses the claimed “behavior information about the first computer object.”

126. **“providing contents of the second database to at least one database associated with a central server”**: The ’721 Patent describes the central server as follows: “[T]he central computer and the community database are presented *for convenience* as single entities.” EX-1001 8:35-39 (emphasis added). But, the ’721 Patent explains, “the base computer 3 can be comprised of multiple computers and servers, etc. and the community database 7 can be made up of multiple databases and storage distributed around this central system.” *Id.* And “[t]he ENZO layer 70 has one or more servers 74 (referred to as ‘central servers’ or ‘ENZO servers’ in this disclosure) connected to one or more ‘master databases which aggregate all the information received from all web servers 61 from all remote computers 2.” *Id.*, 9:53-57. A POSITA would have known, therefore, that the plain and ordinary meaning of the central server claimed by the ’721 Patent is any server having aggregated information from remote computers.

127. In Liu, a tiered system distributes updates in a secure peer-to-peer network. EX-1005, [0070] (“In many embodiments, the alert server 16 and the computers 12 form a secure peer-to-peer network that facilitates dissemination of the update. . . . In the event that the information is not available from a peer computer, the alert server 16 can provide the information directly to the computer 12. In several embodiments, some of the functions described above in relation to the alert server are performed by one or more update servers.”).

128. Likewise, in Liu “the functions of the alert server can be performed by one or more servers” and “the threat protection network can include numerous mirrored servers.” EX-1005, [0101]. A POSITA would have known that the fact that the threat-protection network can include multiple mirrored servers would mean that other servers contain the same information as that stored on the first alert server. Therefore, Liu’s mirrored alert server is functionally the “central server” claimed by the ’721 Patent. For example, the ’721 Patent’s “central server” (or “ENZO server”) keeps an updated master list of the information about all of the objects seen on the remote computers, EX-1001, 12:42-50, and sends this information to the threat servers, *id.*, 12:50-54.

129. In any case, it would have been obvious to a POSITA to create a system with Liu’s functions in a centralized architecture (for instance, a “hub-and-spoke” design”), instead of a pure peer-to-peer architecture. The degree of

centralization in any system design is not a choice about functionality (i.e., about whether or how a system will perform its core functions), but rather about practicality: An operator will choose a degree of centralization on the basis of cost, the need for redundancy, efficiency, scalability, and so on, based on the operator's specific needs. Thus, a POSITA would have known that the functionality in Liu of mirrored alert servers could have been implemented in a centralized system, as claimed by the '721 Patent.

130. A POSITA would therefore also have known that Liu's disclosure of "mirrored" servers means that a mirrored server would include information about all of the objects seen on the computers and could send that information to other alert servers. And a POSITA would have understood that the "mirrored" server and the alert servers would store the same information, including the information in the first alert server's second database. A POSITA would also have understood that each alert server would have its own databases because each alert server is mirrored. Moreover, a POSITA would have understood distributed systems design choices include whether coordination between the distributed services is performed through a centralized server (the more traditional approach) or an hierarchical approach (also traditional for scaling, performance, and ease of coordination reasons), or by a peer-to-peer system is an implementation choice. The trade-offs in scalability,

availability, reliability, consistency, etc. presented by such choices were well-known at the time.

131. Thus, a POSITA would have understood that the mirrored alert server in Liu has the same functionality as the claimed central server, even though Liu refers to both servers as alert servers. Therefore, a POSITA would have understood that the mirrored server would store information contained in a database associated with it, because the first server would have a database associated with it.

132. In addition, Kester discloses “providing contents of the second database to at least one database associated with a central server.” Kester discloses sending data from the Application Inventory Database (and the perhaps separate Uncategorized Application Database (a second database located at the Application Server Module) to be incorporated into the Master Database located at the Application Database Factory (the central server), EX-1007, [0053], [0060], [0073]. A POSITA would have understood that the combination of Liu and Kester discloses the claimed “providing contents of the second database to at least one database associated with a central server.”

[1F]: increasing a count associated with a number of times that the first computer object has been seen, and providing the increased count associated with the number of times that the first computer object has been seen to the central server; and

133. Liu and Tuvell disclose and render obvious [1F].

134. **“increasing a count associated with a number of times that the first computer object has been seen”**: In Liu, the server can provide in a return code “whether the potential threat has previously been reported by another client.” EX-1005, [0082]. One way to provide such a return code is for the system in Liu to keep a count of the number of times it has seen a computer object (and increase that count when it sees an object). A POSITA would have been familiar with Simple Network Management Protocol (SNMP) and its use of counters held in the Management Information Bases (MIBs) of the managed nodes, which contain data regarding managed computer objects. SNMP is a protocol, first established early in the history of the Internet, and refined over the years, to allow distributed management of computers, routers, and other devices (“nodes”) connected to a network. The protocol does not itself prescribe what information is to be collected or how the information should be used, but rather provides a framework within which nodes can collect information deemed relevant and management software can access that information as well as settings. SNMP documentation specifically deals with counters, as these were in common use at the time. In practice, each vendor provides state information and counters in Management Information Bases (MIBs) that can be read (and possibly written) by management software. In some cases these are peculiar to a particular device. In other cases, vendors have agreed, either explicitly or implicitly, to provide certain counters and controls using a common

convention, to improve interoperability and to make management of disparate devices easier. Typical examples include counters for the number of bytes or packets sent or received, the number of connections made, the number of times a particular event has occurred, and so on. Maximum and minimum rates or other values for a period of time were also often used, as were enumerated-state values. The specificity of the counters and other variables can be configured depending on the device. A POSITA would have known that use of such a counter would have been an obvious choice to track malware.

135. And in Tuvell, collection agents are distributed across a network, and that “[t]he system and method works by collecting data from distributed locations, thereby increasing the likelihood that a new malware sample [is] captured once it starts spreading.” EX-1006, [0011], [0073]. In the Tuvell system, “collection agent 1102, 202 notifies 1216 the sample collection center 1112 of the number of times it has seen the executable.” EX-1006, [0089]; *see id.*, Fig. 11. In order to know the number of times an executable has been seen, the system must maintain a counter and increment it whenever the executable is seen. This is the claimed “increasing a count associated with a number of times that the first computer object has been seen.” And a POSITA would have found it obvious that notifying the collection center of the count of times the collection agent has seen the object would necessarily

require increasing a count of the number of times the first computer object has been seen.

136. **“providing the increased count associated with the number of times that the first computer object has been seen to the central server”:**

Because Tuvell discloses that the collection agents’ count can be provided to the sample collection center, Tuvell therefore by definition discloses providing the count to another entity, the sample collection center. EX-1006, [0089] In managed networks, it was common to use Simple Network Management Protocol (SNMP) and associated Management Information Bases (MIBs) to track activity at network nodes. Often MIBs included counts and frequency information (count per time period) to inform network and system management decisions. As demonstrated above, a POSITA would have been motivated to modify Liu to incorporate Tuvell’s count-tracking and providing the count information to mirrored servers in order to offer additional data to support the monitoring of potential threats and their severity.

137. Tracking a count of times a potential threat has been seen in a network was a commonly used technique in malware detection systems at the time of the claimed invention of the ’721 Patent. (Tuvell is a “network management system that monitors malware.” EX-1006, Abstract.). As Tuvell correctly notes, collecting information about potential malware can make prevention and protection easier (the system in Tuvell “collect[s] potential malware applications from

distributed locations throughout a service provider's mobile network 302, increasing the likelihood that new malware samples are captured once they start spreading." EX-1006, [0093]). Along similar lines, Liu is a "threat protection network" that "determine[s] the nature of potential threats." EX-1005, Abstract. Therefore, and as explained above, a POSITA would have known to modify Liu to include a count of times seen, because doing so would have generated the predictable result of increasing the available information about a potential threat, which could make harm-prevention easier.

138. Moreover, because incorporating Tuvell would allow Liu's method to provide further information about the potential threats encountered in a network, a POSITA, would have known to modify Liu to keep a running counter each time the system sees a particular object, and then give that count to the central server.

139. Because Tuvell discloses the different threat information that can be collected, and because a count of the number of times seen was well-known and commonly used in malware detection systems, a POSITA would have reasonably expected that the modification of Liu in view of Tuvell in the manner described would succeed.

[1G]: receiving, at a second threat server, at least a portion of the contents of the at least one database associated with the central server, wherein the at least a portion of the contents of the at least one database associated with the central

server include a subset of the details of the first computer object stored in the second database.

140. Liu discloses [1G].

141. **“receiving, at a second threat server”**: In Liu, “the functions of the alert server can be performed by one or more servers.” EX-1005, [0101]. Furthermore, as noted above, “the threat protection network can include numerous mirrored servers.” *Id.* A POSITA would have known that Liu’s central server would mirror its information to another alert server (e.g., the claimed second threat server). Therefore, a POSITA would have known that Liu’s disclosure of multiple mirrored servers (all of which reside within Liu’s threat-protection network) also discloses the claimed receiving threat data at a second server.

142. **“receiving . . . at least a portion of the contents of the at least one database associated with the central server,”**: In Liu, “the threat protection network can include numerous mirrored servers.” EX-1005, [0101]. A POSITA would have known that information mirrored from the central server to another server would include the information the central server contains.

143. **“wherein the at least a portion of the contents of the at least one database associated with the central server include a subset of the details of the first computer object stored in the second database”**: And, as demonstrated above in [1E], the contents of the claimed second database are provided to the central

server. Therefore, the information mirrored to the claimed second threat server would necessarily include information about the object that was stored in the second database. A “subset” can include any number of elements of a set (including zero or all elements)—that is the plain meaning of the word “subset.” How a database is implemented in a distributed system can include replication (mirroring) and fragmentation (placing a subset of the database in various locations). How the database is fragmented and replicated (perhaps recursively) is a well-known design consideration in databases. *See, e.g.*, EX-1013 826-30.

2. Dependent Claim 8

[8P]: The method according to claim 1, further comprising:

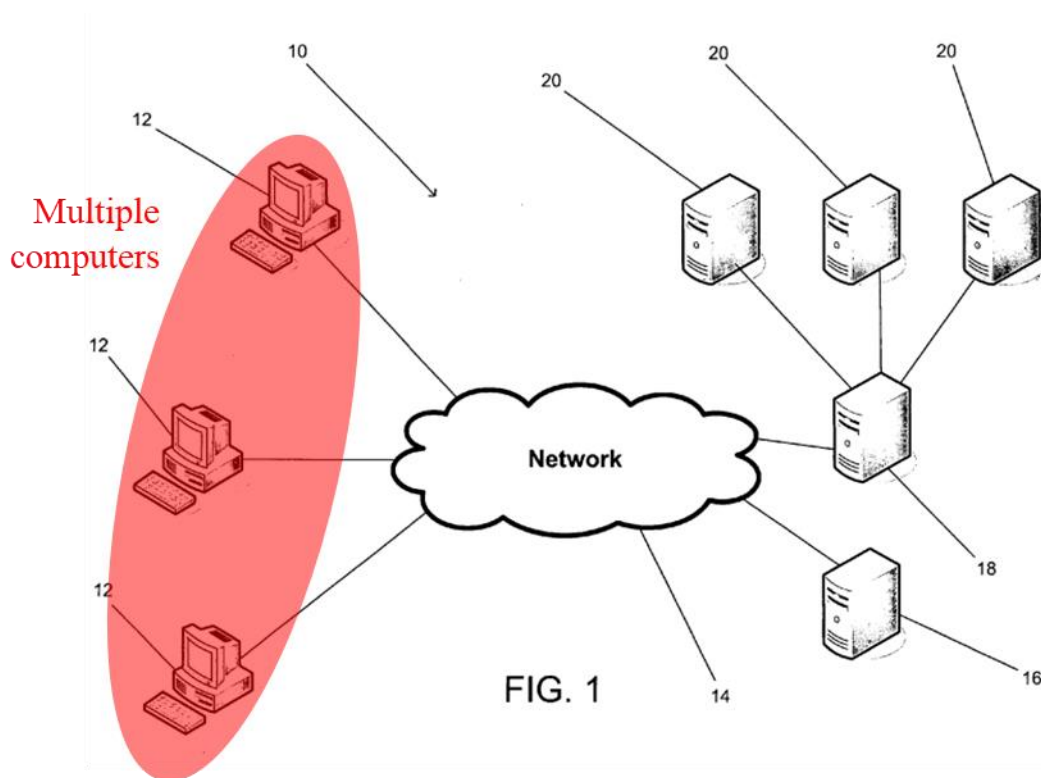
144. *See supra* Independent Claim 1.

[8A]: receiving, at the second threat server, details of a second computer object from a second remote computer, wherein the details of the second computer object include data uniquely identifying the second computer object;

145. Liu discloses [8A].

146. Claim [8A] is analogous to claim [1A], except that in [8A] a second computer sends the unique details of a second computer object to a second threat server (instead of a first computer sending unique details of a first computer object to a first threat server). As explained above in [1G], in Liu “the functions of the alert server can be performed by one or more servers.” EX-1005, [0101]; *see supra* [1G].

“[T]he client can inspect the return code to ascertain (96) whether the potential threat has previously been reported by another client.” EX-1005, [0082]. A POSITA would have known that each of the disclosed threat servers could be associated with multiple computers, and that in a P2P system, multiple clients may be served by multiple alert servers, so a different alert server than the one serving the first client could have received the threat report from the second client. EX-1005, Fig. 1 (showing multiple computers 12):



Furthermore, a POSITA would have known that Liu’s disclosure of sending unique details of a first computer object from a first remote computer to a first threat server would apply in exactly the same way to an additional remote computer connected to

an additional threat server. Therefore, a POSITA would have understood that Liu teaches “receiving, at the second threat server, details of a second computer object from a second remote computer, wherein the details of the second computer object include data uniquely identifying the second computer object.”

[8B]: determining, by the second threat server, whether the second computer object has been previously seen by comparing the data uniquely identifying the second computer object to a plurality of data uniquely identifying plural computer objects in a third database associated with the second threat server;

147. Liu discloses [8B].

148. Claim [8B] is analogous to claim [1B] except that the second threat server compares the computer object’s unique information to information of objects in a third database associated with the second server in order to determine whether the computer object has been previously (instead of the comparison taking place at the first threat server). And, as demonstrated in [1G] above, Liu provides for the existence of multiple servers in the network. *See supra* [1G]. A POSITA would have understood that each threat server could be associated with multiple additional computers.

149. Moreover, a POSITA would have known that Liu’s disclosure of determining whether the system has previously seen a particular object would apply as well to another remote computer connected to another threat server. A POSITA

would have also known that a separate threat server would be associated with its own database (for instance, the third database claimed by the '721 Patent). Therefore, a POSITA would have known that Liu teaches “determining, by the second threat server, whether the second computer object has been previously seen by comparing the data uniquely identifying the second computer object to a plurality of data uniquely identifying plural computer objects in a third database associated with the second threat server.”

[8C]: determining that the second computer object has been seen before;

150. Liu discloses [8C].

151. As demonstrated in [1G], in Liu, the network can include multiple alert servers. *See supra* [1G]. A POSITA would have known that each alert server could be associated with multiple additional computers. Liu discloses that “[t]he server compares (114) the threat signature to the threat definition data. If the threat signature is listed in the threat definition data (the threat signature may be indicated as safe or as a threat), then the server sends (116) a return code that indicates the nature of the potential threat.” EX-1005, [0083]. A POSITA would have known that the condition of whether “the threat signature is listed in the threat definition data” means the condition of whether the object has been seen before.

152. Liu teaches that “expert systems in accordance with the present invention can be deployed without any initial threat definition data and can be deployed in closed networks running proprietary software. An expert system trained in accordance with the present invention can rapidly build appropriate threat definition data in almost any type of networked environment.” EX-1005, [0101]. A POSITA would have known that if an instance of Liu’s system is deployed without threat definition data, and then at some point in the future the threat-definition data contains the threat signature of a particular object, then the system has determined that that object (i.e., potential threat) has been seen before.

[8D]: increasing a count associated with a number of times that the second computer object has been seen and providing the increased count associated with the number of times that the second computer object has been seen to the at least one central server; and

153. Liu and Tuvell disclose [8D].

154. Claim [8D] is analogous to claim [1F] except that the count of number of times seen refers to a second computer object. As explained in [1G], Liu teaches multiple alert servers within the network. *See supra* [1G]. A POSITA would have known that an server could be associated with multiple additional computers.

155. As shown in [1F], Liu and Tuvell disclose maintaining a count of the number of times an object has been seen. A POSITA would have known that this

disclosure could apply just as well to all objects seen on the network. This makes technical and intuitive sense: If the system could only keep count of the number of times some objects, but not others, were seen, the system would not have a complete view of the objects existing on the network. And as Tuvell correctly notes, “collect[ing] potential malware applications from distributed locations throughout a . . . network” can “increase[] the likelihood that new malware samples are captured once they start spreading” and “allow[] preventative measures to be taken sooner, potentially preventing or at least reducing any damage the malware will ultimately cause.” EX-1006, [0093]. If the system had an incorrect or incomplete view of the objects on the network, it might take the wrong preventive measures. A POSITA would have been motivated to combine Liu and Tuvell for the reasons explained above in [1.F]. Therefore, a POSITA would have understood that Liu and Tuvell disclose maintaining a count of the number of times the second computer object has been seen.

[8E]: receiving, at the first threat server, a count associated with the number of times that the second computer object has been seen.

156. Liu discloses [8E].

157. As explained above, Liu teaches the possibility of multiple alert servers. EX-1005, [0101] (“For example, the functions of the alert server can be

performed by one or more servers”). Liu also teaches that “the threat protection network can include numerous mirrored servers.” *Id.* As further explained above, a POSITA would have known that another alert server would mirror the information from the central server. For instance, a second threat server may mirror the alert server. And a POSITA would have known that the information mirrored on another alert server would include details about the times-seen count from the central server. Therefore, a POSITA would have understood that Liu discloses “at the first threat server, a count associated with the number of times that the second computer object has been seen.”

3. Independent Claim 9

158. Claim 9 is the system version of Claim 1 and is otherwise identical in substance. Because Liu, Tuvell, and Kester teach Claim 1, they therefore teach Claim 9 as well.

[9P]: A system for classifying a computer object as malware, the system comprising:

159. Because [9P] is the system version of [1P], [9P] is invalid over prior art for the same reasons that [1P] is invalid over prior art.

[9A]: a first threat server arranged to receive details of a computer object from a first remote computer, wherein the details of the first computer object include data uniquely identifying the first computer object,

160. Because [9A] is the system version of [1A], [9A] is invalid over prior art for the same reasons that [1A] is invalid over prior art.

[9B]: wherein the first threat server is further arranged to receive the details of the computer object from the first remote computer and determine whether the first computer object has been previously seen by comparing the data uniquely identifying the first computer object to a plurality of data uniquely identifying plural computer objects in a first database associated with the first threat server,

161. Because [9B] is the system version of [1B], [9B] is invalid over prior art for the same reasons that [1B] is invalid over prior art.

[9C]: wherein the first threat server is further arranged to receive additional information about the first computer object from the first remote computer when the first computer object has not been previously seen,

162. Because [9C] is the system version of [1C], [9C] is invalid over prior art for the same reasons that [1C] is invalid over prior art.

[9D]: store the details of the first computer object and the received additional information about the first computer object in a second database associated with the first threat server when the first computer object has not been previously seen,

163. Because [9D] is the system version of [1D], [9D] is invalid over prior art for the same reasons that [1D] is invalid over prior art.

[9E]: provide contents of the second database to at least one database associated with a central server wherein the contents comprise a signature of the first

computer object, behavior information about the first computer object, and information about the first remote computer,

164. Because [9E] is the system version of [1E], [9E] is invalid over prior art for the same reasons that [1E] is invalid over prior art.

[9F]: and increase a count associated with a number of times that the first computer object has been seen; the central server arranged to receive the increased count associated with the number of times that the first computer object has been seen; and

165. Because [9F] is the system version of [1F], [9F] is invalid over prior art for the same reasons that [1F] is invalid over prior art.

[9G]: a second threat server arranged to receive at least a portion of the contents of the at least one database associated with the central server, wherein the at least a portion of the contents of the at least one database associated with the central server include a subset of the details of the first computer object stored in the second database.

166. Because [9G] is the system version of [1G], [9G] is invalid over prior art for the same reasons that [1G] is invalid over prior art.

4. Dependent Claim 14

[14P]: The system according to claim 9, wherein the central server comprises:

167. *See supra* Independent Claim 9. As previously noted, the “central server” is disclosed by Liu’s mirrored alert server, which performs the functions described in the Claim 14 sublimitations.

[14A]: a) an object database storing object signatures and metadata about objects;

168. Liu discloses [14A].

169. As explained above, *see* [1E], the system in Liu provides the contents of the second database to at least one other database associated with a central server. Those contents of the second database include a signature of the first computer object, behavior information about the first computer object and information about the first remote computer. *See supra* [1E].

170. Moreover, a POSITA would have found it obvious to store these different types of information across different databases, in part because a POSITA would have also understood from Liu that different databases could be used to store data received at different times. For example, in Liu, if the first threat server receives unique information about an object, like its signature, then, if the server has not seen that object before (i.e., the threat-signature comparison yields no match), the first threat server receives a copy of the potential threat. Last, as discussed above, the threat-identification system observes the object's behavior, and then generates and sends a report regarding the object's behavior. *See supra* [1A], [1C], [1D].

171. It would also have been obvious to a POSITA to store the particular different types of information claimed by [14] in different databases. These different types of information (the threat signature and metadata, the object behavior

information, and the information about the devices on which the objects are present) all have different storage requirements and uses. They are different sizes and have different amounts of variability in size. The system will use them in different ways. A POSITA would have known that it would make sense, therefore, to store these types of information in different databases, because systems become more efficient when similar types of data are stored together (e.g., a database filled with fixed-size entries may be more efficient and easier to manage than a database with variable-sized entries, and databases that group data that are often accessed together and separately from data that are infrequently accessed together allows for more efficient in-RAM storage and virtual memory usage).

172. Therefore, a POSITA would have known that the different types of data could be stored in different databases. For instance, an object's unique signature could be stored in one database (e.g., an object database storing object signatures), its behavior information could be stored in a second database (e.g., a behavior database storing behavior information like the threat report), and other information could be stored on a third database (e.g., a threat-copy database storing information about the object including the copy of the potential threat).

[14B]: b) a behavior database storing object behavior information; and

173. As discussed in [14A], a POSITA would have known to store object-behavior information in its own database. Object-behavior data is likely to be large,

and the entries are likely to vary significantly in size. The data may also include many different instances of behavior for the same objects. It would therefore have been efficient to store object-behavior data together, and a POSITA would have found it obvious to do so.

[14C]: c) a computer-object database storing information about what objects are present on what remote computers.

174. See [14A] *supra*.

175. As discussed in [14A], a POSITA would have known to store information about which objects are present on which computers in its own database. This information is essential for solving problems caused by malware, because in order to effectively remediate, the system needs to know where the problems are located and which systems and devices are affected. It would therefore have been efficient to store object-location information in its own database, and a POSITA would have found it obvious to do so.

176. In addition, Kester discloses “inventories” that are uploaded to an “application server module,” and these inventories include data regarding “what applications are being used by each workstation.” EX-1007, [0056]. And Tuvell discloses reports provided to a server that include device-specific information for remote devices affected by malware (e.g., mobile-device identifier). See EX-1006, [0035], [0036], [0049].

5. Dependent Claim 15

[15]: The system according to claim 14, wherein the first and second threat servers and the central server are implemented using cloud computing.

177. Liu discloses [15].

178. In Liu, “the computers and servers of a threat protection network . . . communicate via a network. In many embodiments, the network includes wired and/or wireless connections. In several embodiments, the network is a network of networks such as the Internet.” EX-1005, [0069]. Liu states that “the functions of the alert server can be performed by one or more servers, [0101]. A POSITA would have known that multiple servers accessed by communication over the Internet would have included cloud computing. The ’721 Patent discloses “protection is provided by signatures that are stored in the cloud, i.e. in a central server to which the remote computers are connected.” *Id.*, 1:56-60. Just as computers communicating over a 5G cellular-phone network today are in some ways comparable to computers communicating over a 10BaseT LAN, so a cloud-based server using current technology is comparable to ’721’s “any type of suitable server” and in particular to a server accessed over a WAN as disclosed in Liu. Provisioning of service through multiple servers is especially pertinent, as cloud systems essentially allow rental of one or more servers from a cloud provider, with the number of servers changing on demand (i.e., elastic computing). The incorporated

cloud-based method “involves communicating, storing and managing vast amounts of data at the central server, which is a challenging problem in itself.” *Id.*, 2:29-31. This would have motivated a POSITA to apply well-known database fragmentation and replication methods, as well as well-known distributed system architectures, to solve the problems of handling large amounts of data and high service demands, including use of elastic cloud computing facilities.

VIII. SECONDARY CONSIDERATIONS

179. I am not aware of any evidence of secondary considerations of non-obviousness relied upon by Webroot. To the extent Webroot is permitted to provide any evidence of secondary considerations to rebut my opinions expressed herein, I reserve the right to respond by supplementing and/or amending this declaration or submitting a reply declaration to consider the alleged secondary considerations of non-obviousness.

IX. RIGHT TO SUPPLEMENT

180. I reserve the right to supplement this declaration, if appropriate, should additional information come to light that is relevant to my opinions in this declaration. I also reserve the right to consider and comment on additional evidence, e.g., objective evidence of non-obviousness, that may be presented by Webroot or its experts.

181. At any trial or hearing at which I may testify in this matter, I may provide demonstrative aids, such as computer animations, excerpts, and figures from relevant exhibits and prior-art references, and physical examples to assist in explaining the subject matter in my report.

182. I will continue my study and evaluation of relevant prior art as these proceedings continue, at least up until the end of the period prescribed by the relevant rules and regulations. In the event further issues arise that concern the subject matter in my declaration or additional information becomes available, I expressly reserve the right to supplement and/or amend my opinions as stated in this declaration as well as the bases supporting my opinions. I specifically reserve the right to supplement and/or amend this declaration at such time that the Board defines the disputed terms in the Challenged Claims, particularly if the Board adopts alternate instructions not advanced by Petitioners. I further reserve the right to supplement and/or amend this declaration to submit a reply declaration based on any information or contentions expressed by Webroot or its experts attempting to rebut my opinions in this declaration. In addition, I reserve the right to use demonstratives or other exhibits at trial in support of my opinions in any of my expert declarations and/or supplements thereto.

X. CONCLUSION

183. For the reasons set forth above, it is my opinion that the Challenged Claims of the '721 Patent are invalid.

Executed on January 19, 2023
Gainesville, Florida

By: /s/ Dr. Richard E. Newman

Dr. Richard E. Newman
8524 SW 23rd Place
Gainesville, FL 32607
(352) 283-1083
nemo@ufl.edu