

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FORCEPOINT LLC,
Petitioner,

v.

WEBROOT, INC. AND OPEN TEXT, INC.,
Patent Owners.

IPR2023-00786
U.S. Patent No. 10,025,928

DECLARATION BY MICHAEL T. GOODRICH, PH.D.

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. SUMMARY OF OPINIONS.....	1
III. BACKGROUND AND QUALIFICATIONS.....	2
A. Education Background and Career History.....	2
B. Publications and Patents.....	2
C. Other Relevant Qualifications.....	4
IV. UNDERSTANDING OF APPLICABLE LAW	7
A. General Patent Law	7
B. Obviousness.....	8
V. PERSON OF ORDINARY SKILL IN THE ART.....	11
VI. OVERVIEW OF THE '928 PATENT	12
VII. PRIOR ART ASSERTED.....	15
A. Devarajan.....	15
B. McAfee	19
C. Hubbard	23
D. Motivation to Combine Devarajan and McAfee.....	24
E. Motivation to Combine Devarajan and Hubbard	26
VIII. CLAIM CONSTRUCTION.....	27
IX. GROUND 1: CLAIMS 1-4, 6-7, 9-12, 14, 16-20 ARE OBVIOUS IN VIEW OF DEVARAJAN.....	28
A. Claim 1	28
B. Claim 2	61
C. Claim 3	66
D. Claim 4	68
E. Claim 6	71
F. Claim 7	73
G. Claim 9	74

H.	Claim 10	77
I.	Claim 11	77
J.	Claim 12	78
K.	Claim 14	78
L.	Claim 16	78
M.	Claim 17	79
N.	Claim 18	80
O.	Claim 19	83
P.	Claim 20	84

X. GROUND 2: CLAIMS 1-4, 6-7, 9-12, 14, 16-20 ARE OBVIOUS OVER DEVARAJAN IN VIEW OF HUBBARD 84

A.	Claim 1	84
B.	Claim 2	89
C.	Claim 3	89
D.	Claim 4	89
E.	Claim 6	89
F.	Claim 7	89
G.	Claim 9	89
H.	Claim 10	92
I.	Claim 11	92
J.	Claim 12	92
K.	Claim 14	92
L.	Claim 16	92
M.	Claim 17	92
N.	Claim 18	93
O.	Claim 19	93
P.	Claim 20	93

XI. GROUND 3: CLAIMS 1-20 ARE OBVIOUS OVER DEVARAJAN IN VIEW OF MCAFEE 93

A.	Claim 1	93
B.	Claim 2	102
C.	Claim 3	105
D.	Claim 4	106
E.	Claim 5	108
F.	Claim 6	110
G.	Claim 7	111
H.	Claim 8	114

I.	Claim 9	116
J.	Claim 10	118
K.	Claim 11	119
L.	Claim 12	119
M.	Claim 13	119
N.	Claim 14	119
O.	Claim 15	120
P.	Claim 16	120
Q.	Claim 17	121
R.	Claim 18	121
S.	Claim 19	125
T.	Claim 20	125
XII.	SECONDARY CONSIDERATIONS	125
XIII.	CONCLUSION	126

I, Michael T. Goodrich, Ph.D., declare under penalty of perjury as follows:

I. INTRODUCTION

1. My name is Michael T. Goodrich, Ph.D. I am a Professor in the Department of Computer Science at the University of California, Irvine. I have been retained on behalf of Forcepoint LLC (“Forcepoint”). I understand that Forcepoint is requesting that the Patent Trial and Appeal Board (“PTAB” or Board) institute an *inter partes* review (“IPR”) proceeding regarding claims 1-20 of U.S. Patent No. 10,025,928 (Ex.1001, “the ’928 patent”). I have been asked to analyze the ’928 patent, and to provide an assessment as to the validity of the ’928 patent in light of prior art.

2. For time spent in connection with this matter, I will be compensated in the amount of \$750 per hour. My compensation does not depend on the content of my opinions or the outcome of this case.

II. SUMMARY OF OPINIONS

3. Based on my analysis, skill, knowledge, education, experience, training, I have formed the following opinions:

- (i) Devarajan renders obvious claims 1-4, 6-7, 9-12, 14, and 16-20 of the ’928 patent.
- (ii) Devarajan in combination with Hubbard renders obvious claims 1-4, 6-7, 9-12, 14, and 16-20 of the ’928 patent.
- (iii) Devarajan in combination with McAfee renders obvious claims 1-20 of the ’928 patent.

III. BACKGROUND AND QUALIFICATIONS

4. Details of my education, work experience, and research are summarized in my curriculum vitae (“CV”) which is attached to this Report as Appendix A. I have summarized some of the relevant information which is on my CV herein.

A. Education Background and Career History

5. I received a Bachelor of Arts degree in Mathematics and Computer Science from Calvin University, in Grand Rapids, Michigan, in 1983, a Master of Science (M.S.) degree in Computer Science from Purdue University, in West Lafayette, Indiana, in 1985, and a Ph.D. in Computer Science from Purdue University in 1987.

6. I am a Distinguished Professor in the Department of Computer Science at the University of California, Irvine, where I have been a faculty member since 2001. The Distinguished Professor title is a campus-level distinction and is reserved for faculty who have achieved the highest levels of scholarship over the course of their careers. I was a professor in the Department of Computer Science at Johns Hopkins University from 1987-2001.

B. Publications and Patents

7. I have authored or coauthored over 350 publications, including several widely adopted books, such as Data Structures and Algorithms in Java, Data Structures and Algorithms in C++, Data Structures and Algorithms in Python,

Introduction to Computer Security, and Algorithm Design and Applications. My publications include contributions to data structures and algorithms, information visualization, networking, graph algorithms, computational geometry, distributed and parallel algorithms, cloud security, information security and privacy, and algorithm engineering. For example, I have published peer-reviewed research articles, including publications J-63, J-71, J-78, C-80, C-83, C-85, C-91, C-100, and C-108, on stopping viruses in email attachments and on authenticating data that is downloaded from untrusted repositories. I have also published peer-reviewed research articles on the use of parallelism in computer systems, including publications J-5, J-7, J-12, J-16, J-19, and J-24. Also, several of my textbooks include entire chapters on software engineering, and several of my publications, including Ch-8, Ch-12, J-36, J-45, C-53, C-62, C-63, C-64, C-208, are on algorithm engineering. Altogether, my publications have, according to Google Scholar, been cited over 16,000 times.¹ My research has been supported by grants from the Defense Advanced Research Projects Agency (“DARPA”), the National Security Agency (“NSA”), the Office of Naval Research (“ONR”), the Army Research Office (“ARO”), and the National Science Foundation (“NSF”).

¹ See <https://scholar.google.com/citations?user=sSS9gSoAAAAJ>

8. I am a co-inventor on several U.S. patents, including U.S. Patent No. 7,257,711, “Efficient Authenticated Dictionaries with Skip Lists and Commutative Hashing,” which discloses secure distributed data authentication schemes. I am also co-inventor of U.S. Patent No. 7,299,219, “High Refresh-Rate Retrieval of Freshly Published Content using Distributed Crawling,” which discloses a technology for quickly retrieving website data that can change frequently, so as to be stored in a search engine. I am also co-inventor of U.S. No. Patent 8,681,145, “Attribute Transfer Between Computer Models Including Identifying Isomorphic Regions in Polygonal Meshes,” which teaches how to map one mesh-based computer model to another. In addition, I am co-inventor of U.S. Patent No. 9,152,716, “Techniques for Verifying Search Results Over a Distributed Collection,” which discloses a system for searching the Internet so as to produce verifiable search results that can be produced by a search engine.

C. Other Relevant Qualifications

9. In addition, I have consulting experience in matters involving algorithms, cryptography, machine learning, digital rights management, computer security, mobile devices (including power management), networking, software, video streaming, and storage technologies.

10. I am a Fellow of the American Association for the Advancement of Science (“AAAS”), a Fellow of the Institute of Electrical and Electronics Engineers

(“IEEE”), and a Fellow of the Association for Computing Machinery (“ACM”), as well as being named as a Foreign Member of the Royal Danish Academy of Sciences and Letters. I am also a recipient of a Fulbright Scholarship (for senior specialist service to University of Aarhus, Denmark). In addition, I am a recipient of the IEEE Computer Society Technical Achievement Award (“for outstanding contributions to the design of parallel and distributed algorithms for fundamental combinatorial and geometric problems”) and the Pond Award for Excellence in Undergraduate Teaching. Also, I am an ACM Distinguished Scientist.

11. My study of computer and network security topics began as an undergraduate student and continued in graduate school, where my Ph.D. research involved the study of computer system components operating in parallel. My study and interest in computer security continued after my Ph.D., as detailed above and in my CV. In addition, I have reviewed and evaluated research papers on computer security, including cryptography, beginning with work as an associate editor for Journal of Computer & System Sciences, as well as my service on program committees of peer-reviewed Computer Science conferences, including the Conference on Electronic Publishing and the Information Superhighway and the ACM Symposium on Theory of Computing (STOC), the latter for which I chaired and edited the conference proceedings.

12. I have taught courses at Johns Hopkins University, Brown University, and University of California, Irvine, at both the undergraduate and graduate levels. Topics of my courses have included computer security, algorithms, data structures, information visualization, computer graphics, networking, algorithm engineering, computational geometry, and parallel processing. In addition, I have mentored twenty-five (25) PhD students over the years, who have written their PhD theses on topics in algorithms, data structures, information visualizations, networking, parallel processing, and computer security and privacy.

13. I have served as an editor on several technical journals, including Computational Geometry: Theory and Applications, Journal of Computer & System Sciences, Journal of Graph Algorithms and Applications, Int. Journal of Computational Geometry & Applications, and Information Processing Letters. I have also served on many program committees (PCs) for top conferences and workshops in Computer Science, including serving as PC chair in several instances. Examples include ACM Symposium on Computational Geometry (“SoCG”), ACM Symposium on Theory of Computing (“STOC”), Workshop/Symposium on Algorithms and Data Structures (“WADS”), Algorithm Engineering and Experimentation (“ALENEX”), which I cofounded with Dr. Catherine McGeoch in 1999), IEEE Symposium on Foundations of Computer Science (“FOCS”), ACM-SIAM Symposium on Discrete Algorithms (SODA), International Symposium on

Graph Drawing (“GD”), International Colloquium on Automata, Languages, and Programming (“ICALP”), ACM Conference on Computer and Communications Security (“CCS”), European Symposium on Algorithms (“ESA”), IEEE International Parallel and Distributed Processing Symposium (“IPDPS”), ACM Symposium on Parallel Algorithms and Architectures (“SPAA”), ACM Symposium on Advances in Geographic Information Systems (“GIS”), IEEE Symposium on Security and Privacy (“S&P”), IEEE International Conference on Big Data, IEEE International Conference on Data Engineering (“ICDE”), and International Symposium on Algorithms and Computation (“ISAAC”), and DAGS Conference on Electronic Publishing and the Information Superhighway.

IV. UNDERSTANDING OF APPLICABLE LAW

14. I am not an attorney. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to my opinions. My understanding of the law was provided to me by the Petitioner’s attorneys.

A. General Patent Law

15. I understand that in order to determine the validity of a patent claim, the meaning and scope of that patent claim must be determined. I understand that in construing or interpreting a patent claim, the words of the claim are generally given their ordinary and customary meaning, which is the meaning that words would have had to a person of ordinary skill in the art (“POSITA”) at the time of the invention.

The POSITA is deemed to read the claim term in the context of not only the particular claim in which the disputed term appears, but also in the context of the entire patent, including the other claims and the specification, as well as the prosecution history of the patent.

16. I further understand that an invalidity analysis requires that the claims (as construed) be compared to the prior art. If a claim was anticipated by or obvious over the prior art, the claim is invalid. I understand that the invalidity analysis is conducted based on the understanding of a POSA at the time of the invention.

17. I understand that prior art to the '928 patent may include patents and printed publications that predate the alleged priority date of the '928 patent.

B. Obviousness

18. It is my understanding that a patent claim is invalid for obviousness if the claimed invention would have been obvious at the time the invention was made to a POSITA. I further understand that a claimed invention may be invalid even if each and every limitation is not disclosed in a single prior art reference. I have been informed that, under the doctrine of obviousness, a claim may be invalid if the differences between the invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA to which the subject matter pertains.

19. Obviousness, as I understand it, is based on the scope and content of the prior art, the differences between the prior art and the claim, the level of ordinary skill in the art, and secondary indicia of obviousness and nonobviousness to the extent they exist.

20. I understand that a claim can be found invalid as obvious if design incentives or market forces provided a reason to make an adaptation, and the invention resulted from the application of prior knowledge in a predictable manner. I understand that a claim can be found invalid as obvious if the claim would have been obvious because the substitution of one known element for another would have yielded predictable results to one of ordinary skill in the art at the time of the invention. I understand that a claim can be found invalid as obvious if the claim would have been obvious because the technique for improving a particular class of devices was part of the ordinary capabilities of a POSITA. I understand that a claim can be found invalid as obvious if the claim would have been obvious because a particular known technique was recognized as part of the ordinary capabilities of one skilled in the art.

21. I understand that a claim can be obvious in light of a single reference, without the need to combine references, if the elements of the claim that are not found in the reference can be supplied by the common sense of one of skill in the art.

22. Where an invention is a combination of known elements, I understand that I should consider whether a reason existed at the time of the invention that would have prompted one of ordinary skill in the art to combine the known elements in the way the claimed invention does. The reason could come from the prior art, the background knowledge of one of ordinary skill in the art, the nature of the problem to be solved, market demand, or common sense.

23. I also understand that a finding that a POSITA would not have been able to practice the invention without undue experimentation or unpredictable results can defeat a claim of obviousness.

24. I have been informed and understand that in evaluating whether patent claims are invalid as obvious, objective indicia of nonobviousness can be considered. Such objective indicia can include commercial success, long-felt but unsolved needs, copying, praise, unexpected results, industry acceptance, failure of others, and skepticism by experts.

25. I also understand that it is impermissible to use the claimed invention itself as a blueprint for piecing together elements in the art. In other words, it is impermissible to use hindsight reconstruction to pick and choose among disclosures in the prior art to reconstruct the claimed invention.

V. PERSON OF ORDINARY SKILL IN THE ART

26. I understand that there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including the educational level of active workers in the field at the time of the alleged invention, the sophistication of the technology, the type of problems encountered in the art, and the prior art solutions to those problems.

27. In determining the characteristics of a hypothetical person of ordinary skill in the art of the '928 patent at the time of the alleged invention, I considered several things, including the type of problems encountered in this field, and the rapidity with which innovations were made. I also considered the sophistication of the technology involved, and the educational background and experience of those actively working in the field, and the level of education that would be necessary to understand the '928 patent. Finally, I placed myself back in the relevant periods of time, in this case, before 2012, and considered the state of the art and the level of skill of the engineers working in this field.

28. In my opinion, a POSITA for the '928 patent would hold at least a four-year undergraduate degree in computer science or a closely related field and at least two years of experience in the field of computer and network security. Superior education would compensate for a deficiency in experience, and vice versa. For example, the '928 patent identifies the field of its invention as relating “computer

system management.” More specifically, the ’928 patent states that its invention is related to “systems and methods for controlling pestware or malware or other undesirable or unwanted applications and/or instructions.” Ex.1001 (’928 patent) at 1:16-21. Also, the ’928 patent’s specification discloses the use of computer hardware and software (*see, e.g.*, Ex.1001 (’928 patent) at 8:25-28, 9:65-10:20).

29. My opinions provided in this Declaration would not change in view of any minor modifications to this level of skill.

VI. OVERVIEW OF THE ’928 PATENT

30. It is my understanding that the ’928 patent, titled “PROACTIVE BROWSER CONTENT ANALYSIS,” issued on July 17, 2018, from U.S. Patent Application No. 13/633,956 (the “’956 Application”), filed on October 3, 2012. I was informed that the ’928 patent does not claim priority to any other patent or application.

31. The ’928 patent includes two independent claims, claims 1 and 9, both of which relate to content analysis and malware prevention as described by the patent. Like the prior art references that preceded it, the ’928 patent describes a protection agent that analyzes web content at a protocol level to detect and remove any potential malware before the content is served to the user. Ex.1001 (’928 patent) at Abstract.

32. Figure 1 of the '928 patent depicts “an environment in which several embodiments of the invention may be implemented.” Ex.1001 ('928 patent) at 3:2-4. As shown below, the environment includes “a security center 102 (also referred to herein as a ‘central’ or ‘base’ computer), remote user 104 operating a remote computer 105, a malware source 106, and a web server 108 are all communicatively coupled through one or more networks (e.g., the Internet and/or local or wide area networks) 110 and links 5.” Ex.1001 ('928 patent) at 3:4-9.

33. The protection agent, 116, which is part of remote computer 105, “operates to analyze threats, at the protocol level (e.g., at the HTML level), by intercepting all requests that the browser engine 118 sends and receives, and the protection agent 116 completes the requests without the help of the browser engine 118.” Ex.1001 ('928 patent) at 4:7-11. Once the request is complete, “the protection agent 116 analyzes and/or modifies the completed data before the browser engine 118 has access to it.” Ex.1001 ('928 patent) at 4:11-14. In this step, the protection agent, if necessary, “decrypts and decompresses the web content and then assembles the requested web page (e.g., in a decrypted and decompressed HTML format that the web page existed in at the remote server 108).” Ex.1001 ('928 patent) at 4:56-61. This analysis of the assembled webpage “may include communicating with the security center 102 so that the malware management component 112 may analyze one or more portions of the content of the assembled web page.” Ex.1001 ('928

patent) at 4:63-5:1. Once this is complete, “the protection agent feeds the HTML content back to the browser engine 118, and the browser engine 118 receives responses from the protection agent 116 as if it was ‘speaking’ to an actual web server (e.g., web server 108) when in fact, it is speaking to an analysis engine of the protection agent 116.” Ex.1001 (’928 patent) at 4:14-21.

34. Figures 2 and 3 show the operations that take place within the remote computer 105 in greater detail. Figure 2 separates out browser processes from agent processes, however, the agent processes 220, are “implemented by additional code that is wrapped around a typical browser engine 218 to intercept what is requested and received.” Ex.1001 (’928 patent) at 5:45-47. Figure 3 “depicts exemplary aspects of the process flow that occurs in connection with the components depicted in FIG. 2.” Ex.1001 (’928 patent) at 5:26-29.

35. When a user initiates a request via the browser/application 222, “a connection request 360 is initiated as a POST/GET request 362 to a website (e.g., hosted by the webserver 108) and the analysis engine 224 looks at the context of the request to assess whether the request is the first request in a session (Block 364), and if the request is the first request, a determination is made as to whether the request is associated with known, malicious content (Block 366), and if so, the request is blocked (Blocks 368, 370).” Ex.1001 (’928 patent) at 5:52-62. If the request is not blocked, then “the content acquisition component 226 iteratively sends requests and

receives content (Blocks 374, 376, 378) from the website 108 until the requested content has been completely received.” Ex.1001 (’928 patent) at 6:17-23. Once the content acquisition component 226 completely gathers the web content, the content is passed along to the service process component 228 where “the data is then preprocessed (Block 380), if necessary, to decrypt, remove chunks, and decompress the gathered content.” Ex.1001 (’928 patent) at 6:27-32. At this stage, “the protection [agent] has the content in the same HTML format that the website 108 had the content in,” as a result, “the browser page may be analyzed by the protocol-level analysis/modification component 382 depicted in FIG. 3 outside of the context of the browser engine 218 before the browser engine 218 has operated on the webpage.” Ex.1001 (’928 patent) at 6:42-51. If malicious script is found during this analysis, “it may be commented out before the content is handed to the browser engine.” Ex.1001 (’928 patent) at 6:55-57.

VII. PRIOR ART ASSERTED

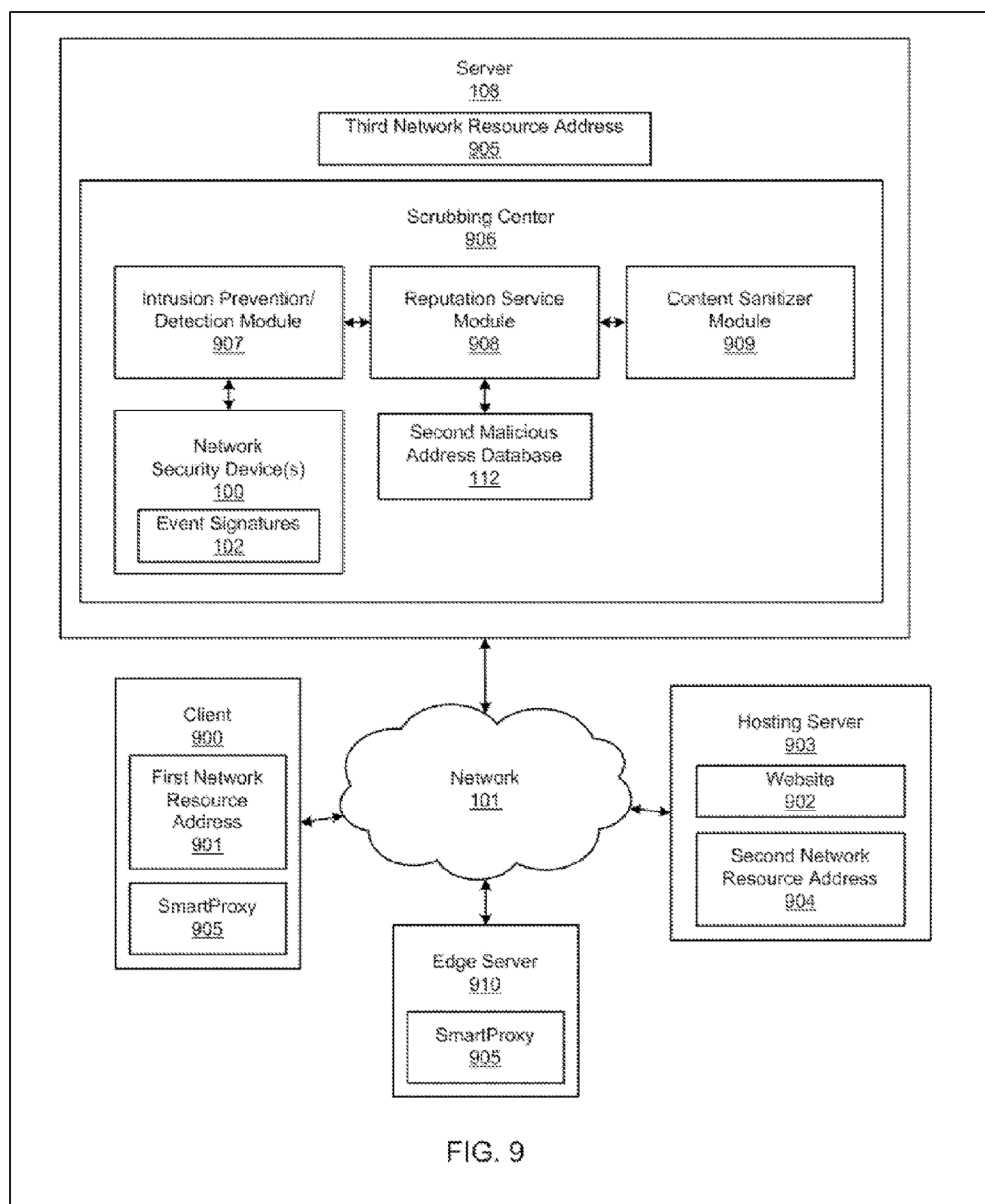
A. Devarajan

36. U.S. Patent Application Publication 2013/0007870 (Devarajan) was filed September 29, 2011, and issued January 3, 2013. I understand that Devarajan is prior art to the ’928 patent.

37. Devarajan is directed to a “bi-directional network traffic malware detection and removal system” that comprises “a scrubbing center running one or

more server computer[s] communicatively coupled to a network configured to receive a request for website content, remove any server-directed malware from the content request, transmit the scrubbed content request to the website's hosting server, receive the responsive website content, remove [any] client-directed malware from the content, and transit the scrubbed content to the requesting client.” Ex.1004 (Devarajan) at Abstract.

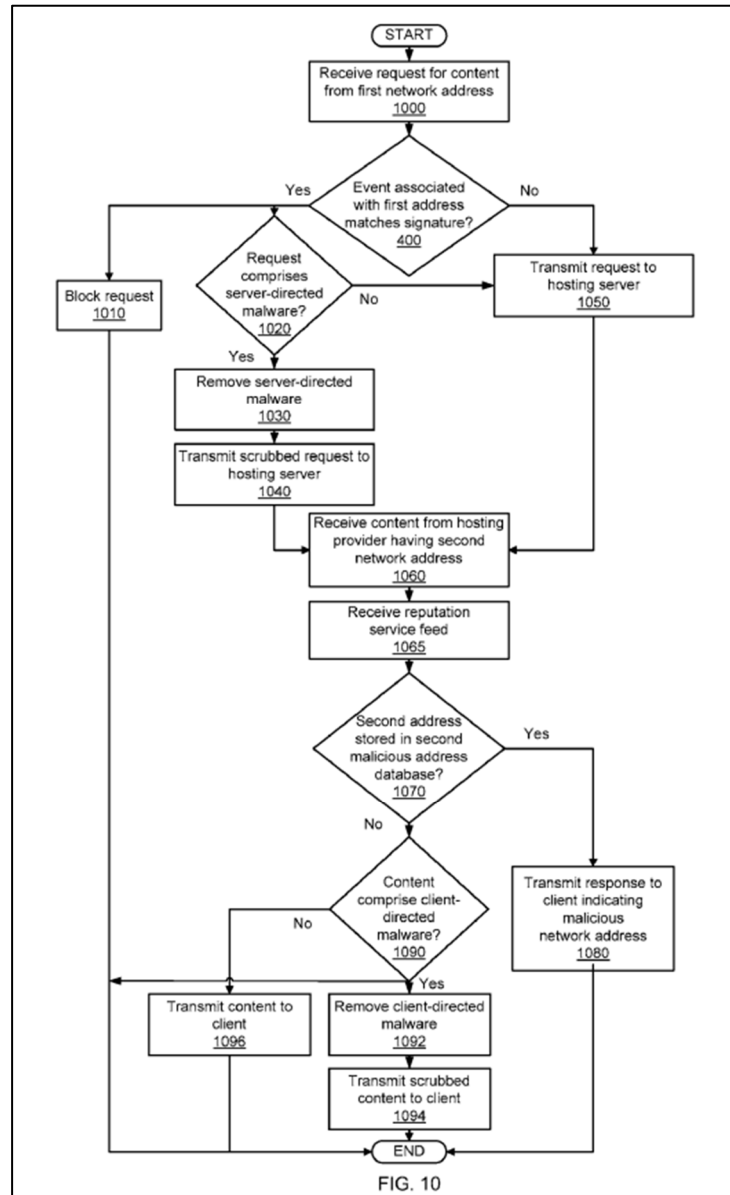
38. Devarajan discloses several embodiments of its invention, one of which is illustrated in Figure 9. Devarajan discloses that “Fig. 9 illustrates an embodiment of a bi-directional network traffic malware detection and removal system that may comprise one or more server 108 (having a third network resource address 905) communicatively coupled to a network 101.” Ex.1004 (Devarajan) at [0088].



Ex.1004 (Devarajan) at Fig. 9.

39. Devarajan further discloses that its system includes a scrubbing center 906 (depicted above in Fig. 9) that “may comprise a plurality of software modules running on the one or more server 108, and may comprise an intrusion prevention and detection module 907, a reputation service module 908, and/or a content sanitizer module 909.” Ex.1004 (Devarajan) at [0095].

40. Furthermore, another embodiment of Devarajan is depicted below in Figure 10.



Ex.1004 (Devarajan) at Fig. 10.

41. Devarajan discloses that “Fig. 10 illustrates an embodiment of a method for bi-directional detection and removal of network traffic malware that may

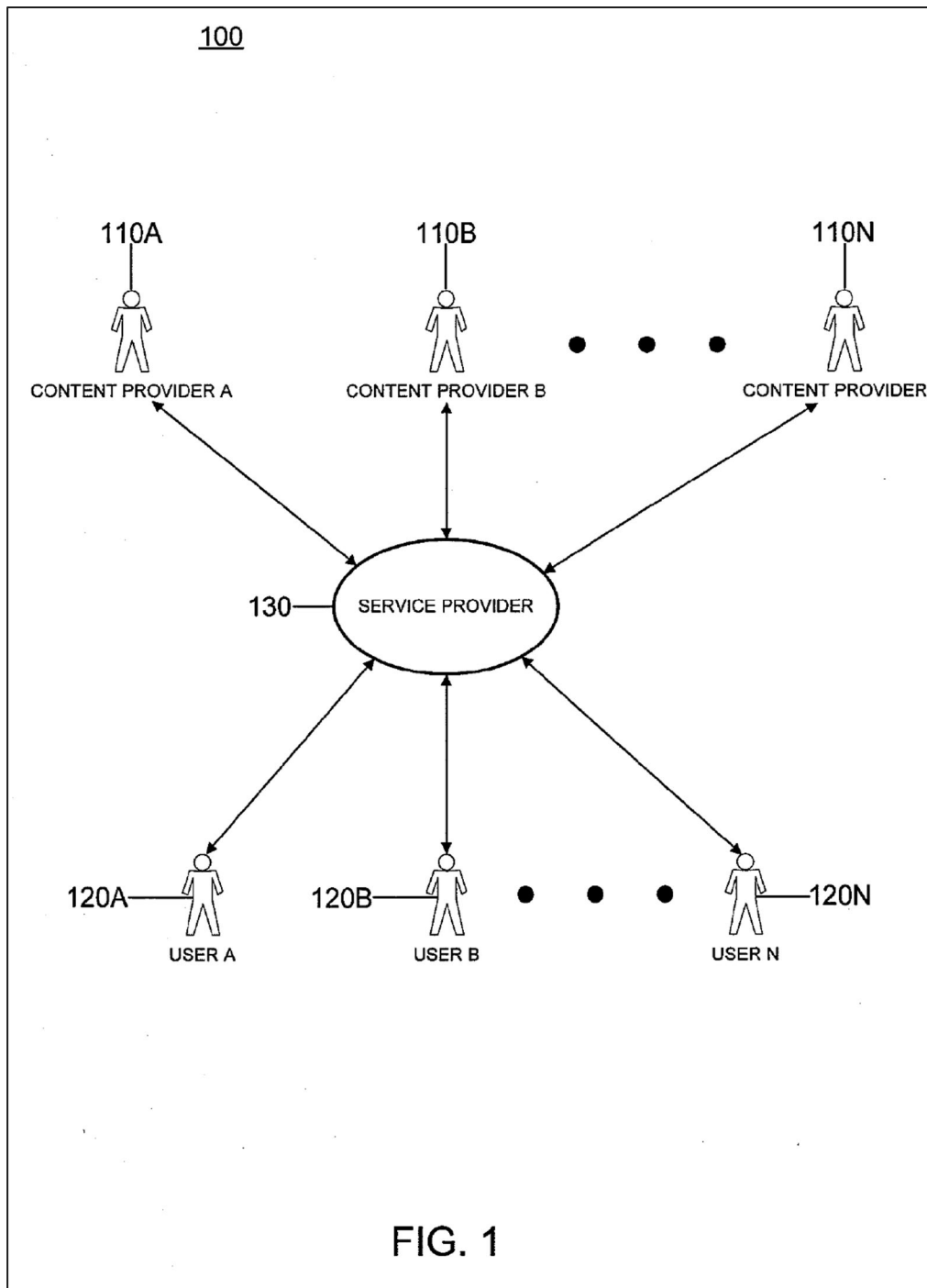
comprise the steps of receiving, from a client 900 having a first network resource address 901, a request for content from a website 902 hosted on a hosting server 903 having a second network resource address 904 and resolving from a URL such as domain name, wherein the URL (e.g., domain name) is pointed in the DNS to a third network resource address 905 for one or more server 108 running a scrubbing center 906 (Step 1000).” Ex.1004 (Devarajan) at [0109].

B. McAfee

42. U.S. Patent Application Publication 2009/0070873 (McAfee) was filed September 11, 2007, and issued March 12, 2009. I understand that McAfee is prior art to the '928 patent.

43. McAfee is directed to a system “for providing safe web based interactions.” Ex.1005 (McAfee) at Abstract. McAfee teaches that its systems enables “users to safely browse web pages by removing or disabling malicious code, or malware, associated with a web page before a user browses the page on a web capable device” by redirecting the “user’s web traffic through the service provider’s servers.” Ex.1005 (McAfee) at [0023].

44. McAfee discloses an embodiment of its architecture as follows:

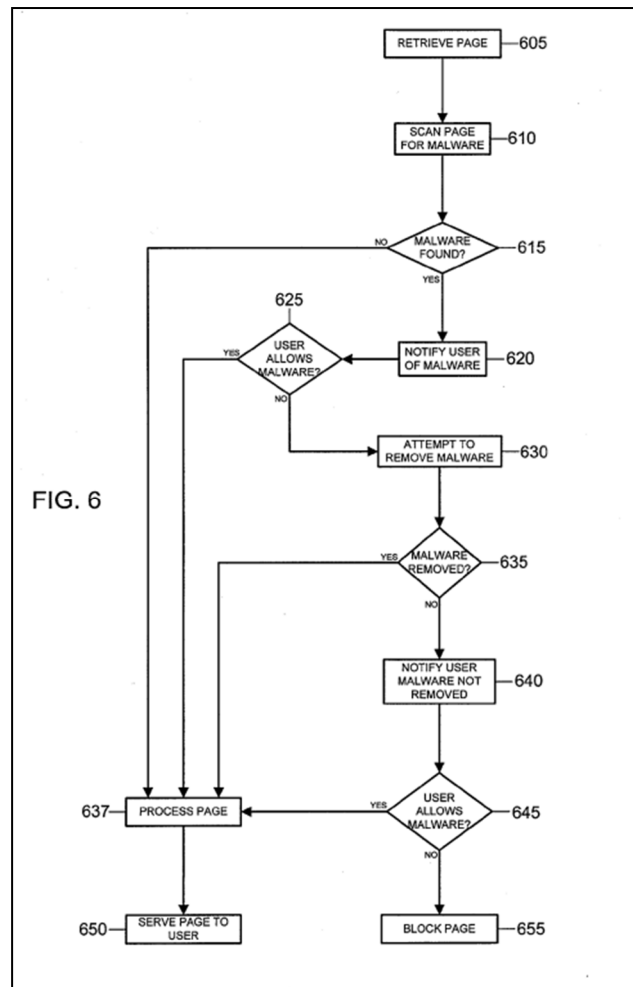


Ex.1005 (McAfee) at Fig. 1.

45. McAfee discloses that the “system 100 may include one or more content providers 110A-N, such as an entity that makes content available on a web page published on the world wide web, a service provider 130, such as a search

engine marketing service provider, and one or more users 120A-N, such as web surfers or consumers.” Ex.1005 (McAfee) at [0030]. “The users 120A-N may access the services provided by the service provider 130 through a web application, such as web browser or any other application capable of displaying web content.” Ex.1005 (McAfee) at [0034]. “When the user A 120A performs a search via the safe search engine, the system 100 may act as an intermediary, redirecting the resultant web traffic of the user A 120A through the servers of the service provider 130, referred to as the ‘main servers.’ The traffic may be redirected by returning the search results in an invisible frame and replacing all of the uniform resource locators (‘URLs’) in the search results with URLs redirecting the traffic through the servers of the service provider 130. The service provider 130 may then retrieve the actual web page the user A 120A wishes to browse, scan the page for malware, process the page, such as by replacing the URLs on the page.” Ex.1005 (McAfee) at [0035]. Then, the system provides the web page to the user.

46. McAfee’s Figure 6, reproduced below, shows the system’s process for analyzing content and preventing malware.



Ex.1005 (McAfee) at Fig. 1.

47. As shown above, at “block 605 the page processor 312 may retrieve the page requested by the user A 120A” and at the next step, block 610, “the page processor 312 may scan the page for malware such as viruses, worms, Trojan horses, rootkits, backdoors, spyware, botnets, loggers, dialers, or generally any code that may be hostile, intrusive, or otherwise bothersome to the user A 120A.” Ex.1005 (McAfee) at [0082]. If the page processor finds malware, it provides the user with “the option to allow the malware, and browse the page with the malware present, or

have the service provider server 240 attempt to remove the malware.” Ex.1005 (McAfee) at [0082]. If the user opts to have the malware removed, at “block 630 the page processor 312 may attempt to remove the malware” by attempting “to remove the malware code.” Ex.1005 (McAfee) at [0083]. If the malware was successfully removed “the system 100 may move to block 637” at which point “the page may be further processed by the page processor 312, the URL processor 314, or the advertisement processor 318.” Ex.1005 (McAfee) at [0083]. Once this is completed, “the service provider server 240 may serve the page to the user A 120A via the network 230.” Ex.1005 (McAfee) at [0083].

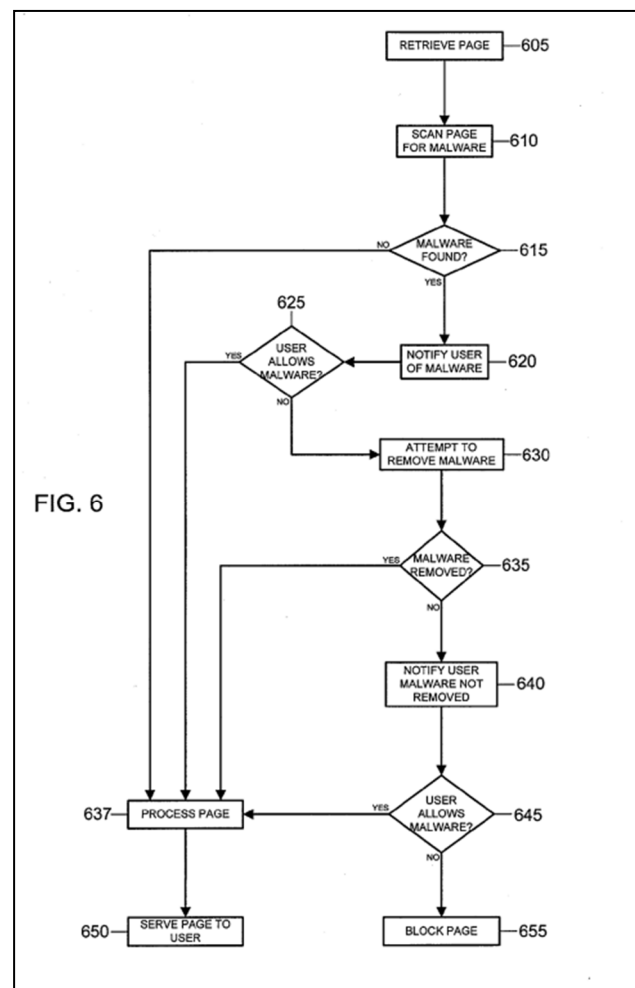
C. Hubbard

48. U.S. Patent No. 9,654,495 (Hubbard) was filed February 28, 2007, and issued May 16, 2017. I understand that Hubbard is prior art to the '928 patent. Hubbard discloses methods and system for “controlling access to web content.” Ex.1008 (Hubbard), 2:33-61. Hubbard discloses “receiving a request for at least one uniform resource locator (URL)”, “generating an indicator of active content associated with the URL” that is “based on data associated with at least one component of the URL,” and “determining whether to allow the request based at least partly on the at least indicator.” *Id.*, 2:34-40. A “reputation scoring module generates a reputation score or other reputation indicator of targeted content associated with the URL” and “the gateway server module 120 determines whether

to allow access to the URL based at least partly on the reputation score.” *Id.*, 27:38-40, 51-53.

D. Motivation to Combine Devarajan and McAfee

49. Like Devarajan, McAfee is directed to a system for “providing safe web based interactions” and specifically, to “providing a safe environment for searching and browsing the web.” Ex.1005 (McAfee) at [0021]; *see also*, Ex.1005 (McAfee) at [0022], [0082]. McAfee’s Figure 6, reproduced below, shows the system’s process for analyzing content and preventing malware.



Ex.1005 (McAfee) at Fig. 6.

50. McAfee teaches that malware may infect a user's computer through a variety of means such as when a user visits a web page that "installs or otherwise executes the malware on the computer" or when a user "downloads and installs software from a malicious web page". Ex.1005 (McAfee) at [0002]. Devarajan detects malware based on known malware network addresses stored in its databases, and in the event that malware is detected, Devarajan teaches removing the malware from the content request. Ex.1004 (Devarajan) at [0004], [0007]. Devarajan's malware detection system can detect malware using "signature-based, statistical anomaly-based, and stateful protocol analysis methods." Ex.1004 (Devarajan) at [0044]. A POSITA in the field of network security would understand the importance of malware detection prior to the execution of the malware on a user's computer, and would turn to references like McAfee, which teach a variety of malware detection methods, according to a similar architecture, to further build on the teachings of Devarajan. For example, a POSITA would be aware that the system of Devarajan would benefit from the web page caching taught in McAfee to speed up subsequent requests for a web page. *See, e.g.*, Ex.1005 (McAfee) at [0061], [0063], [0074]. A POSITA would also understand that adding McAfee's teachings regarding shortened URLs would improve the security of the system of Devarajan, since it would allow the detection of URLs associated with malware even if they

were replaced with (compressed) shortened vcd URLs. *See, e.g.*, Ex.1005 (McAfee) at [0028], [0041]. Thus, a POSITA would be motivated to combine Devarajan with McAfee.

E. Motivation to Combine Devarajan and Hubbard

51. A POSITA would understand that Devarajan and Hubbard are analogous art. For example, Devarajan states that its field of invention relates “to network security and, more particularly, systems, methods, and other tools for providing a network resource address reputation service and systems and methods for bi-directional detection and removal of network traffic malware.” Ex.1004 (Devarajan) at [0003]. Analogously, Hubbard states that its field of invention “relates to data and application security,” stating, “[i]n particular, this application discloses systems methods of collecting and mining data to predict the content or nature of a web site based on its web address.” Ex.1008 (Hubbard) at 1:13-16. Like Devarajan discloses a server 108, which receives a client request for web content, processes the request for possible server-directed malware before sending the request on to the server storing that content, which in turn sends a response back to the server 108, which processes the response to look for client-directed malware before forwarding the response to the client. Ex.1004 (Devarajan) at Fig. 10, [0109]-[0118]. Analogously, Hubbard discloses a “a gateway server module 120,” which “may reside at the edge of the network 110 so that traffic sent to and from the Internet

112 may pass through it on its way into or out of the network 110.” Ex.1008 (Hubbard) at 4:51-54. Thus, a POSITA would look to Hubbard for guidance on ways to configure and enhance the system of Devarajan to improve the interception of web requests and responses with respect to remediating malware. For example, a POSITA would understand the benefits of Hubbard’s disclosure that “[a] page returned by the search engine 113 to the client workstation 116 may be intercepted by the gateway server module 120 and analyzed by the content filtering module 150” (Ex.1008 (Hubbard) at 8:34-37) and would be able to easily apply this teaching to Devarajan’s disclosure of a server 108, e.g., for Devarajan’s “bi-directional network traffic malware detection and removal.” Ex.1004 (Devarajan) at Abstract, [0088]. That is, a POSITA would have an expectation of success in this combination.

VIII. CLAIM CONSTRUCTION

52. For purposes of my analysis in this *inter partes* review proceeding, I have analyzed the terms appearing in the patent claims according to their “plain and ordinary meaning.” In determining the plain and ordinary meaning, I understand that I am to first give the terms their plain meaning that those words would have had to a POSITA in the context of the patent specification and claims. I understand that the structure of the claims, the specification, and the file history also may be used to better construe a claim insofar as the plain meaning of the claims cannot be understood. Moreover, treatises and dictionaries may be used, albeit under limited

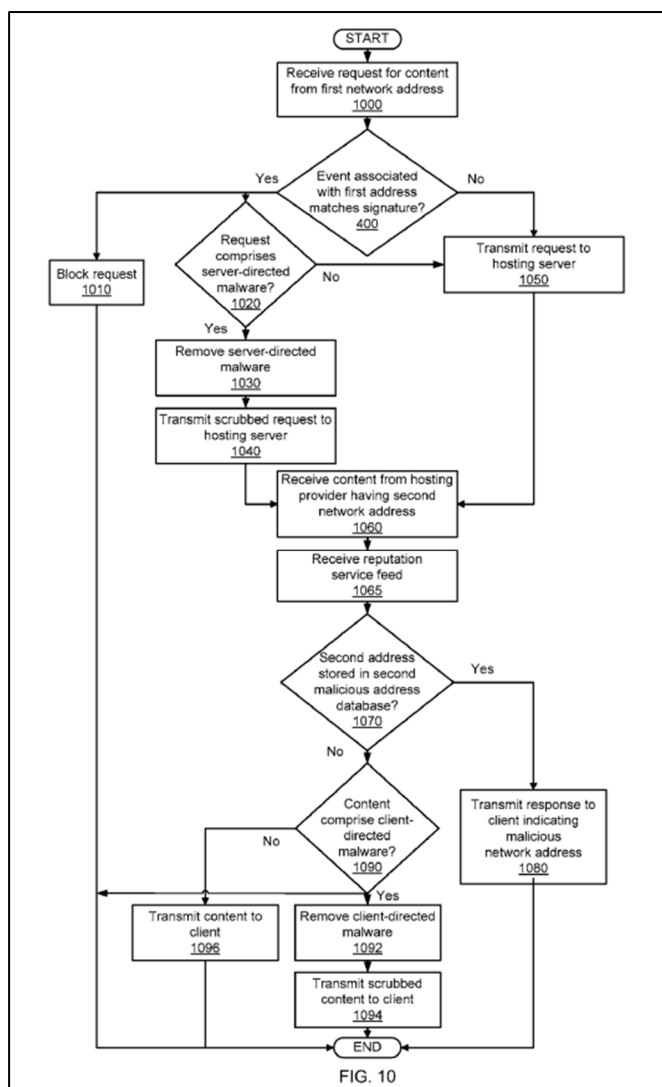
circumstances, to determine the meaning attributed by a POSITA to a claim term at the time of filing. I have followed this approach in my analysis, and for all of the claim terms considered in this declaration, I have applied the plain and ordinary meaning of those terms.

IX. GROUND 1: CLAIMS 1-4, 6-7, 9-12, 14, 16-20 ARE OBVIOUS IN VIEW OF DEVARAJAN

A. Claim 1

1. [1P] A content analysis and malware prevention method comprising:

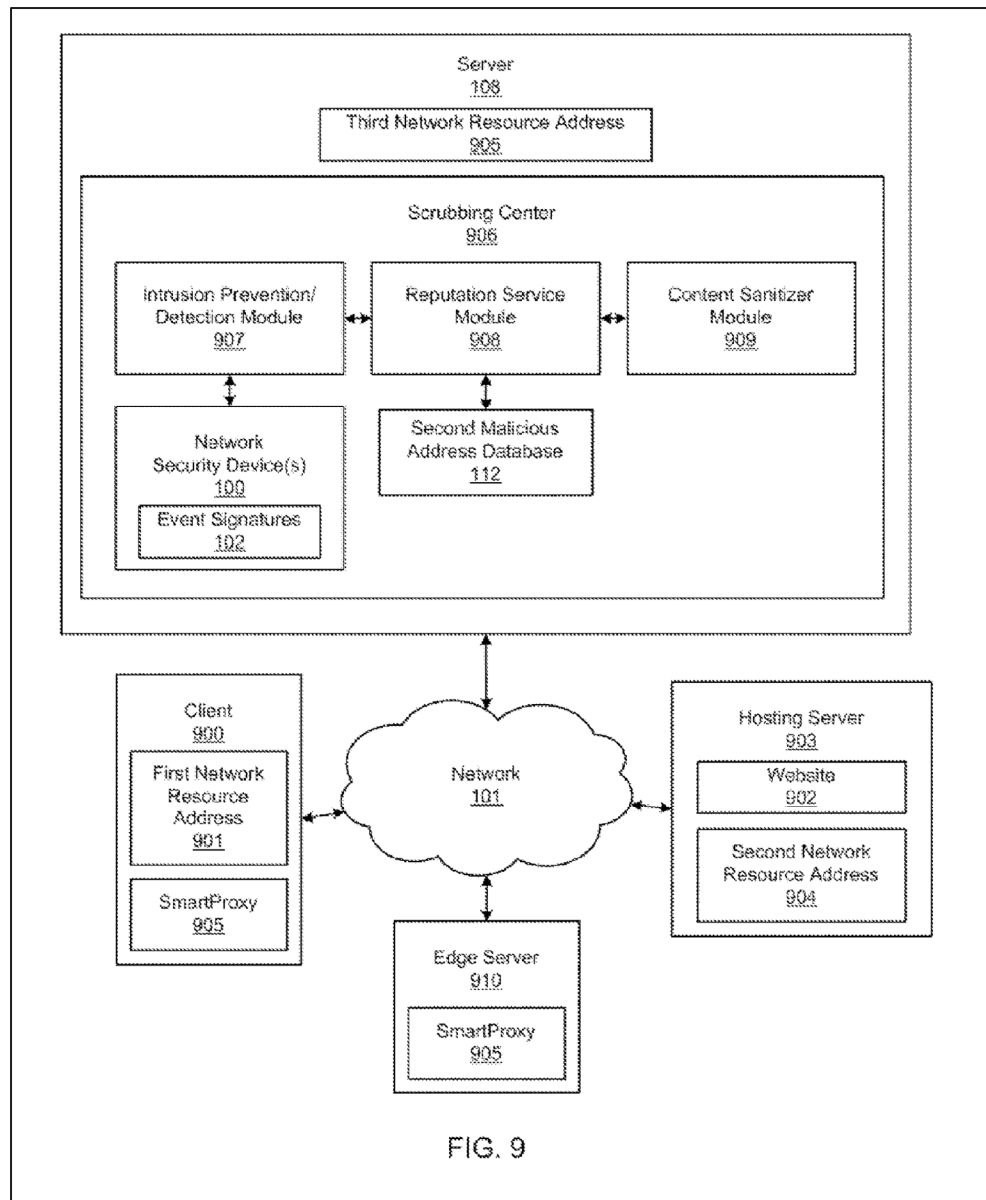
53. To the extent the preamble is limiting, Devarajan teaches a content analysis and malware prevention method. For example, Devarajan teaches a system for “bi-directional network traffic malware detection and removal.” Ex.1004 (Devarajan) at Abstract. Devarajan’s Figure 10, reproduced below, shows a method for analyzing content and preventing malware, which a POSITA would understand to disclose “a content analysis and malware prevention method.”



Ex.1004 (Devarajan) at Fig. 10.

2. [1A] intercepting, by a protection agent, a request from a web browser;

54. Devarajan discloses or renders obvious “intercepting, by a protection agent, a request from a web browser.” For example, Devarajan discloses an architecture diagram in its Figure 9 as follows:

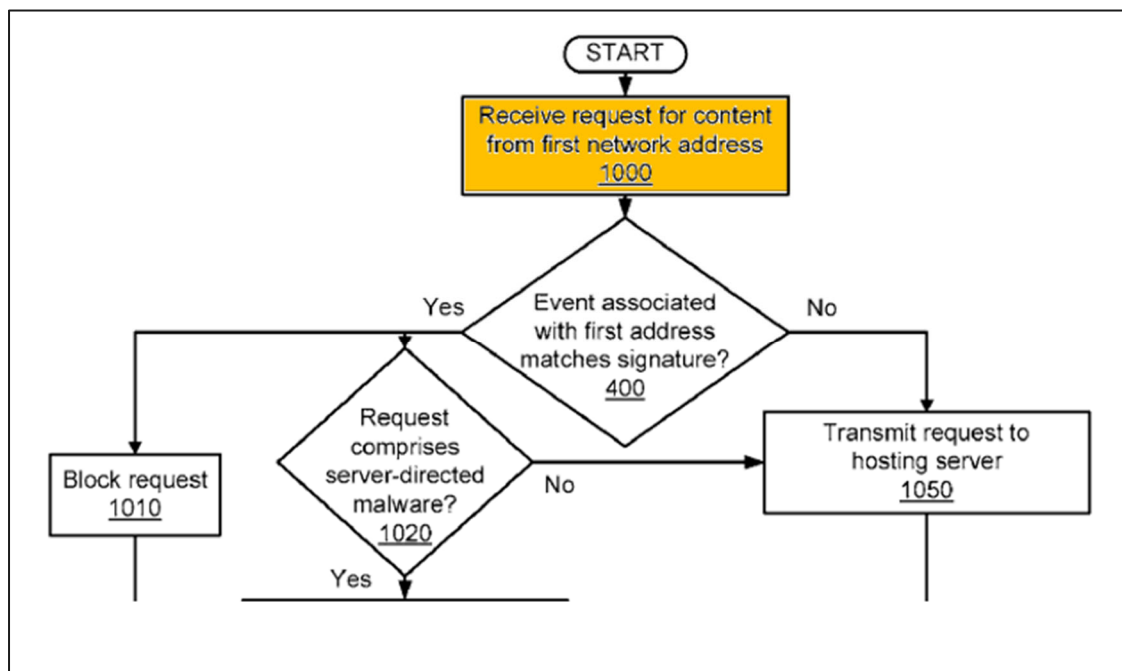


Ex.1004 (Devarajan) at Fig. 9.

55. Devarajan discloses that “Fig. 9 illustrates an embodiment of a bi-directional network traffic malware detection and removal system that may comprise one or more server 108 (having a third network resource address 905) communicatively coupled to a network 101.” Ex.1004 (Devarajan) at [0088]. In addition, Devarajan discloses that the “exemplary bi-directional network traffic

malware detection and removal system” is “configured to receive a request for website content.” Ex.1004 (Devarajan) at [0006]. Devarajan also discloses that its system includes a scrubbing center 906 (depicted above in Fig. 9) that “may comprise a plurality of software modules running on the one or more server 108, and may comprise an intrusion prevention and detection module 907, a reputation service module 908, and/or a content sanitizer module 909.” Ex.1004 (Devarajan) at [0095].

56. Furthermore, an embodiment of Devarajan depicted below in Fig. 10 shows that the system comprises “the steps of receiving, from a client 900 having a first network resource address 901, a request for content from a website 902 hosted on a hosting server.” Ex.1004 (Devarajan) at [0109]. This step is illustrated at block 1000 in the figure below:



Ex.1004 (Devarajan) at Fig. 10 (annotated).

57. Devarajan discloses that “Fig. 10 illustrates an embodiment of a method for bi-directional detection and removal of network traffic malware that may comprise the steps of receiving, from a client 900 having a first network resource address 901, a request for content from a website 902 hosted on a hosting server 903 having a second network resource address 904 and resolving from a URL such as domain name, wherein the URL (e.g., domain name) is pointed in the DNS to a third network resource address 905 for one or more server 108 running a scrubbing center 906 (Step 1000).” Ex.1004 (Devarajan) at [0109].

58. Further, Devarajan discloses, “[e]ach of the at least one servers 108 (and/or any other server described herein) **could be any computer or program** that provides services to other computers, **programs**, or users **either in the same computer or over a computer network 101**. As non-limiting examples, the one of [sic] more server 108 could be application, communication, mail, database, proxy, fax, file, media, web, peer-to-peer, standalone, **software**, or hardware servers (i.e., server computers) and may use any server format known in the art or developed in the future (possibly a shared hosting server, a virtual dedicated hosting server, a dedicated hosting server, or any combination thereof).” Ex.1004 (Devarajan) at [0053]². Thus, a POSITA would understand that a server 108 may be software

² All emphasis added unless otherwise noted.

executing, e.g., on a desktop computer, such as a client 900, and providing services to other software programs running on that computer. *See also, e.g.,* Ex.1004 (Devarajan) at [0092] (disclosing that the client 900 may be, e.g., a desktop computer or a laptop computer).

59. Further, given the above disclosures in Devarajan, a POSITA would find it obvious to implement server 108 as a software server executing on the same computer (client 900) as the web browser. For example, a POSITA would understand that this reduces the communication time between a client and the server 108, since such communications would occur locally rather than going over a network, thereby improving latency for the overall system in Devarajan. Also, a POSITA would understand that placing the server 108 as a software server on the client 900, as described, e.g., in Devarajan at [0053], would reduce the amount of hardware needed to implement the system of Devarajan, thereby reducing costs. Further, Devarajan discloses a SmartProxy 905 that “may act as an intermediary between the client 900 or edge server 910 and the hosting server 903,” and Devarajan shows the SmartProxy 905 as a component on the client 900. Ex.1004 (Devarajan) at [0105], Fig. 9, Fig. 10. A POSITA would understand Devarajan’s description of the functionality of the SmartProxy 905 to include communicating directly to the scrubbing center 906 (which is disclosed in Devarajan as a software module in the server 108, as shown in the excerpt below:

In another possible embodiment, a SmartProxy 905 may be used to divert traffic through the scrubbing center 906. **The SmartProxy 905 may comprise a proxy server application, software, or script that may run on an Internet user's client 900,** or perhaps on the network edge. **The SmartProxy 905 may communicate directly to the scrubbing center 906.** The SmartProxy 905 may act as an intermediary between the client 900 or edge server 910 and the hosting server 903. In one embodiment, the SmartProxy 905 may comprise an application, perhaps downloaded to the client 900 or edge server 910 from a scrubbing center 906 service provider, that ensures all traffic from and/or to the client 900 is routed to, and filtered through, the scrubbing center 906.

As a non-limiting example, the SmartProxy 905 may be configured (e.g., programmed) to receive website 902 content requests from the client 900 and redirect such requests to the scrubbing center 906. As a non-limiting example, the SmartProxy may accomplish this by storing the website 902 hosting server's 903 second network address 904 (e.g., IP address) in association with the scrubbing center 906 server's 108 third network resource address (e.g., IP address), along with instructions to route requests for the website 902 to the scrubbing center 906.

When website 902 content is returned to the client 900, it may be redirected to the scrubbing center 906 to ensure

that, for example, any client-directed malware is removed. Where the client 900 initiates the content request, the returned website 902 content may automatically be redirected to the scrubbing center 906. Where the hosting server 903 initiates a connection with the client 900, the SmartProxy 905 may intercept and redirect the traffic to the scrubbing center 906. As a non-limiting example, the SmartProxy may accomplish this by storing the scrubbing center 906 server's 108 third network resource address (e.g., IP address), along with instructions to route all incoming traffic to the scrubbing center 906 and request that the scrubbing center 906 return scrubbed content to the SmartProxy 905.

Ex.1004 (Devarajan) at [0105]-[0107].

60. Accordingly, a POSITA would understand Devarajan to disclose an embodiment where a SmartProxy 905 and server 108 are software executing on the client 900 computer (e.g., a desktop computer or laptop computer) to provide the functionality of the “protection agent”. *See, e.g.*, Ex.1004 (Devarajan) at [0095]. In my opinion, a POSITA would understand this embodiment in Devarajan to disclose and/or suggest “a protection agent,” as required by claim 1, that in this embodiment comprises a (software) server 108 executing on the client 900 computer.

61. Moreover, Devarajan discloses that a request may come from a web browser:

The content request may be received (Step 1000) by any method, system, or protocol for receiving data, perhaps via an electronic communication received at the server 108 including, but not limited to, a HyperText Transfer Protocol (HTTP) or a File Transfer Protocol (FTP) transmission, an email message, and/or a Short Message Service (SMS) message (i.e., text message). As a specific non-limiting example, the content request may be received via HTTP protocol, the **request perhaps being initiated by the client's 900 browser.**

Ex.1004 (Devarajan) at [0110].

62. Further, Devarajan discloses the server 108 “intercepting” a request from a web browser. As an initial matter, a POSITA would understand that the '928 patent is using the term “intercepting” to have its plain-and-ordinary meaning. *See, e.g.,* Ex.1001 ('928 patent) at 4:6-11, 6:17-26. The '928 Patent describes the protection agent “intercepting” a request from a web browser by receiving and/or stopping the request before it arrives at its intended destination. *See, e.g.,* Ex.1001, 4:6-11, 6:17-26. Likewise, in the context of an electronic communication, a POSITA would understand an interception, under its plain and ordinary meaning, to include interrupting an electronic transmission before it reaches the intended recipient.

63. Devarajan teaches that the server 108 receives electronic communications from a client's web browser before they reach the intended recipient, which is the hosting server 903. *See, e.g.*, Ex.1004 (Devarajan) at [0105]-[0107], excerpted above, and as follows:

The server 108 may be configured (perhaps by installing software and or scripts causing the server 108 to perform the steps for which it is configured) to receive (perhaps from a client 900 having a first network resource address 901) a request for content from a website 902, perhaps resolving from a domain name and hosted on one or more hosting server 903 having a second network resource address 904.

Ex.1004 (Devarajan) at [0089].

64. Further, Devarajan teaches that the content request intercepted by the server 108 may be initiated by a client 900's web browser:

The content request may be received by any method, system, or protocol for receiving data, perhaps via an electronic communication **received at the server 108** including, but not limited to, a HyperText Transfer Protocol (HTTP) or a File Transfer Protocol (FTP) transmission, an email message, and/or a Short Message Service (SMS) message (i.e., text message). As a specific non-limiting example, the content request may be received

via HTTP protocol, the request perhaps **being initiated by the client's 900 browser.**

Ex.1004 (Devarajan) at [0093].

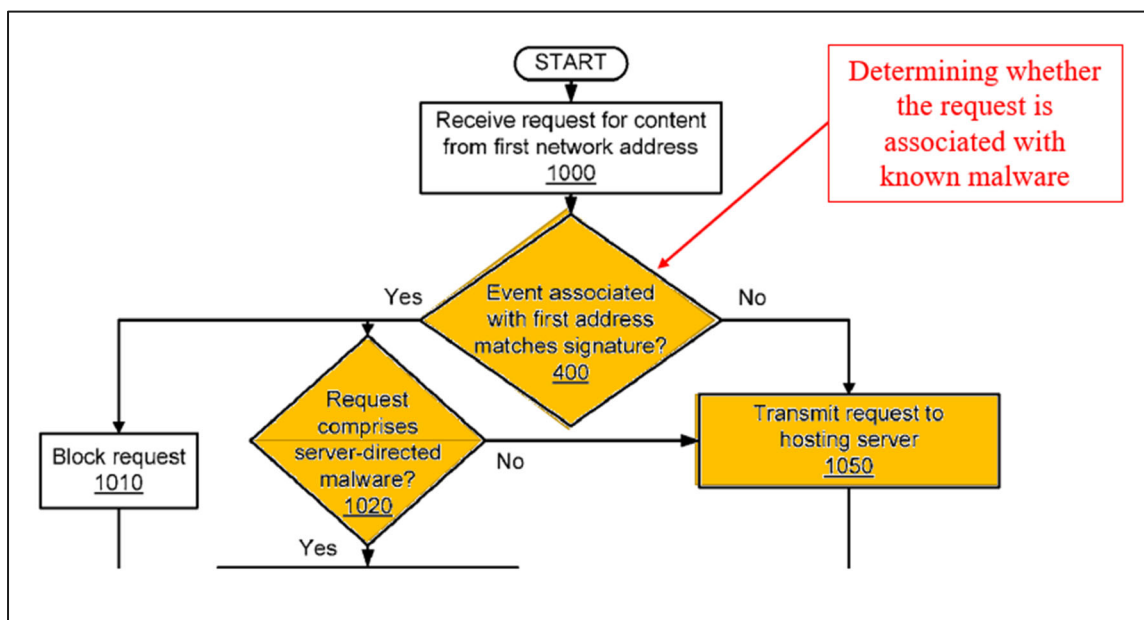
65. Based at least on the above disclosures, a POSITA would understand the server 108 in Devarajan to teach and render obvious “a protection agent,” which intercepts a request for content from a web browser, as required by claim 1. Thus, Devarajan discloses or renders obvious “intercepting, by a protection agent, a request from a web browser.”

3. [1B] determining, by the protection agent, whether the request is associated with known malicious content;

66. Devarajan discloses or renders obvious “determining, by the protection agent, whether the request is associated with known malicious content.”

67. For example, the Figure 10 embodiment of Devarajan, reproduced below, discloses at Step 400 “determining whether an event associated with the client 900 or its first network resource address 901 matches one or more of a plurality of event signatures 102 in one or more network security device 100 communicatively couples to the network 101.” Ex.1004 (Devarajan) at [0111]. A POSITA would understand “an event associated with a client 900” to mean a content request from the client to a website 902. Ex.1004 (Devarajan) at [0090], [0092]. Devarajan further describes that its “signature based security network device 100 may store, or otherwise have access to...a plurality of event signatures 102”, which “are simply

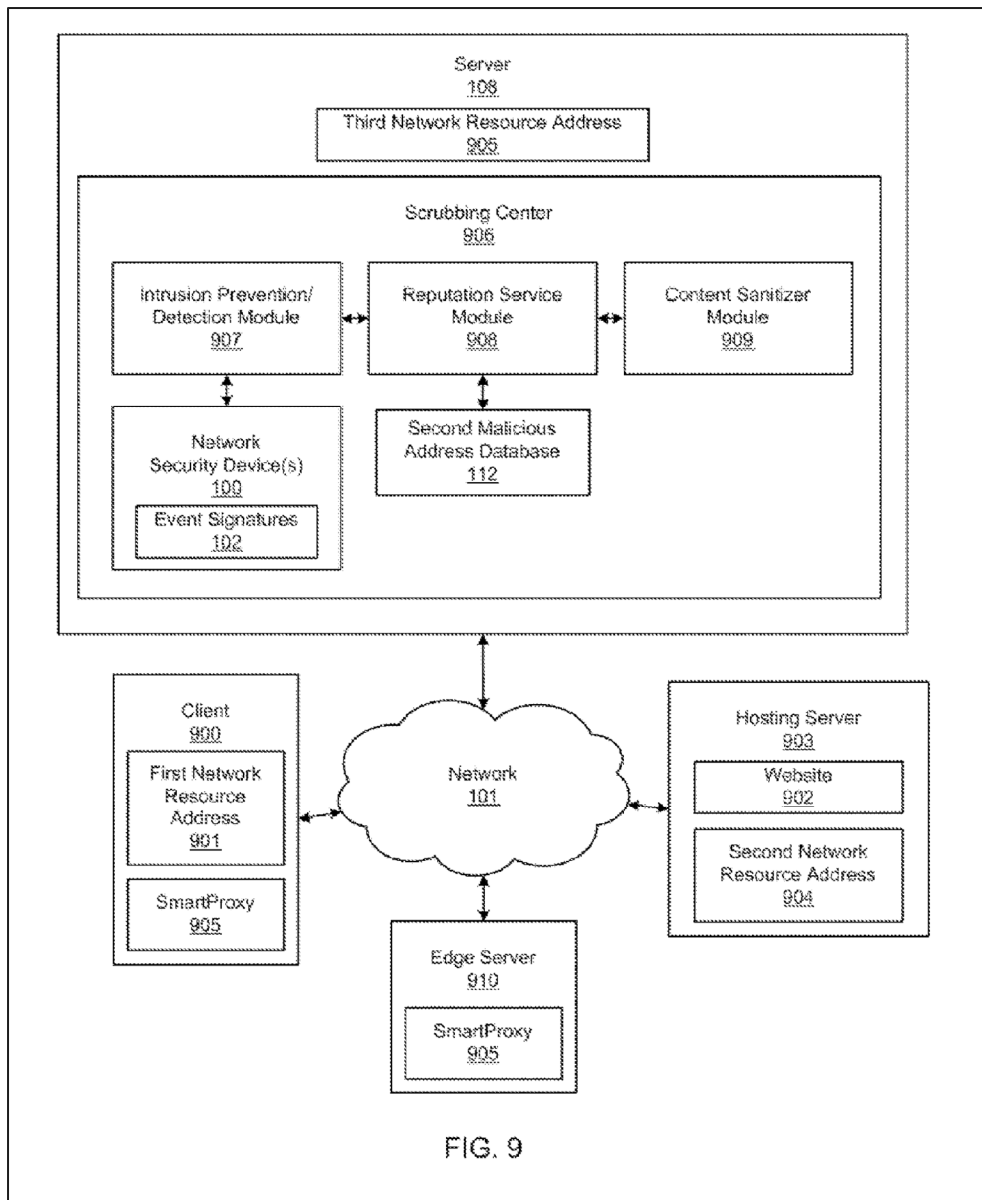
known attack patterns.” Ex.1004 (Devarajan) at [0044]- [0045]. Devarajan explains that these signatures “comprise particular strings that have been discovered in known malicious activity” such as “a virus signature, a worm signature, a trojan horse signature, a rootkit signature, a backdoor signature, a spyware signature, a keystroke logger signature, or a phishing application signature.” Ex.1004 (Devarajan) at [0045]-[0046]. Further, Devarajan discloses that “[a]n ‘event’ may comprise any malicious or unwanted activity, perhaps performed by or via a network resource 103 having a network resource address 104” (Ex.1004 (Devarajan) at [0048]) and that “the network resource address may comprise an IP address, a URL, or a domain (e.g., domain name) for the network resource 103.” Ex.1004 (Devarajan) at [0049].



Ex.1004 (Devarajan) at Fig. 10 (annotated).

68. Moreover, as I explain above for claim element [1A], Devarajan discloses that the server 108 may intercept a content request from a web browser.

For example, Devarajan discloses that the SmartProxy 905 “may act as an intermediary between the client 900 or edge server 910 and the hosting server 903,” as I explain above, and that the SmartProxy 905 “may be used to divert traffic through the scrubbing center 906” (Ex.1004 (Devarajan) at [0105]), which is a software module in the server 108. Devarajan also discloses that the SmartProxy 905 may include “instructions to route all incoming traffic to the scrubbing center 906 and request that the scrubbing center 906 return scrubbed content to the SmartProxy 905.” Ex.1004 (Devarajan) at [0107]. These connections are shown in Figure 9, reproduced below:



Ex.1004 (Devarajan) at Fig. 9.

69. Devarajan teaches that the server 108 receives electronic communications from a client's web browser before they reach the intended recipient, which is the hosting server 903. *See, e.g.,* Ex.1004 (Devarajan) at [0105]-[0107], excerpted above, and as follows:

The server 108 may be configured (perhaps by installing software and or scripts causing the server 108 to perform the steps for which it is configured) to receive (perhaps from a client 900 having a first network resource address 901) a request for content from a website 902, perhaps resolving from a domain name and hosted on one or more hosting server 903 having a second network resource address 904.

Ex.1004 (Devarajan) at [0089].

The content request may be received by any method, system, or protocol for receiving data, perhaps via an electronic communication **received at the server 108** including, but not limited to, a HyperText Transfer Protocol (HTTP) or a File Transfer Protocol (FTP) transmission, an email message, and/or a Short Message Service (SMS) message (i.e., text message). As a specific non-limiting example, the content request may be received via HTTP protocol, the request perhaps **being initiated by the client's 900 browser**.

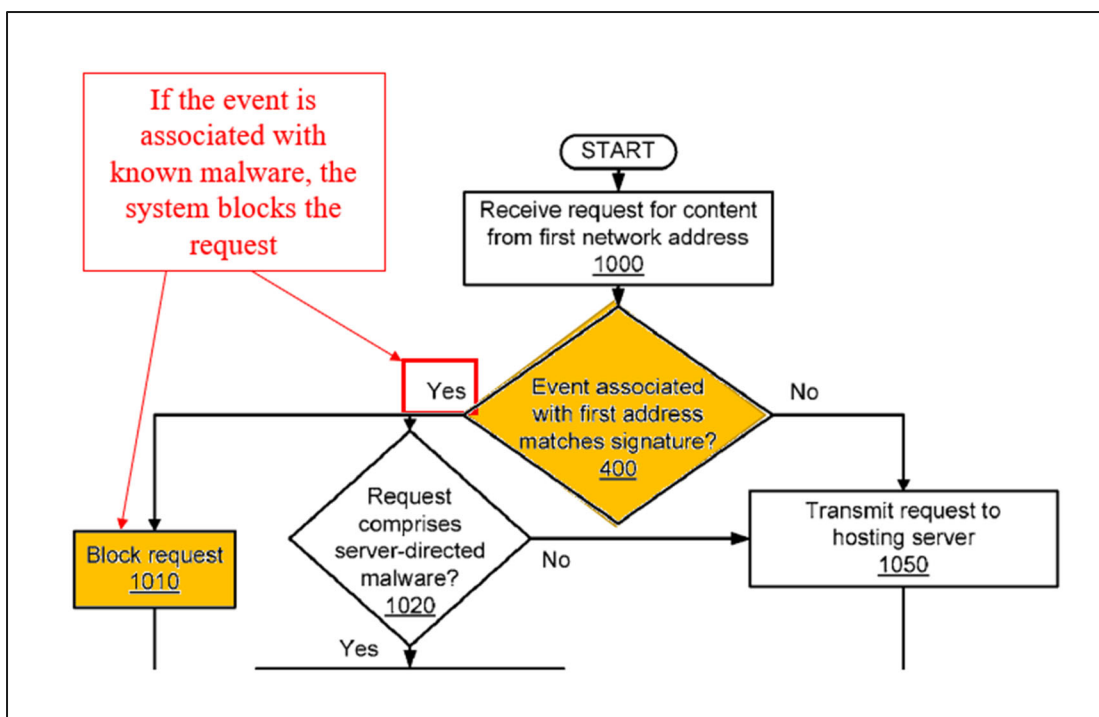
Ex.1004 (Devarajan) at [0093]. *See also, e.g.,* Ex.1004 (Devarajan) at [0110], which has similar language.

70. Thus, a POSITA would understand that the server 108 (e.g., a protection agent), in an embodiment as a software server executing on the client 900, as I explain above, and with its scrubbing center 906, will determine whether the

request from the client is associated with known malicious content. That is, Devarajan discloses or renders obvious “determining, by the protection agent, whether the request is associated with known malicious content.”

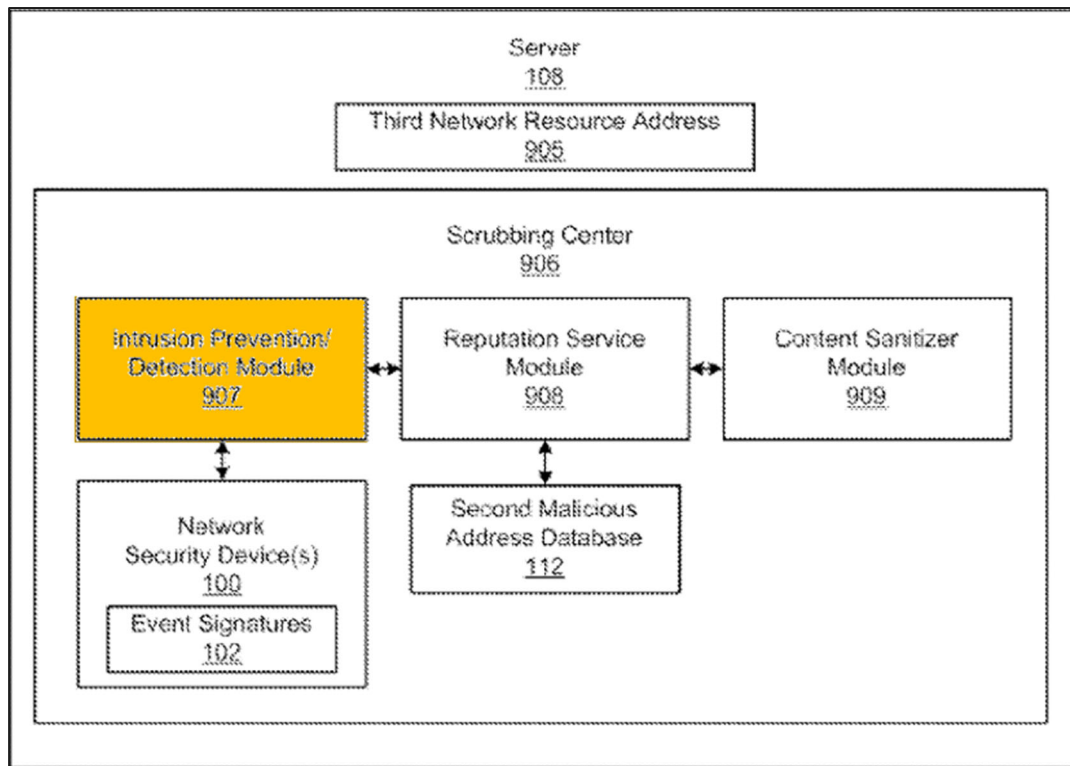
4. [1C] when the request is associated with known malicious content, blocking, by the protection agent, the request;

71. In my opinion, Devarajan discloses or renders obvious “when the request is associated with known malicious content, blocking, by the protection agent, the request.” For example, as explained above, Devarajan’s Figure 10 embodiment discloses, at step 400, “determining whether an event...matches one or more of a plurality of event signatures.” Ex.1004 (Devarajan) at [0111]. Devarajan further teaches that “if the event matches one or more event signature 102, the method further may comprise blocking the request for content from reaching the hosting server 903 (Step 1010)” (Ex.1004 (Devarajan at [0112])), as depicted in the figure below.



Ex.1004 (Devarajan) at Fig. 10 (annotated).

72. Further, in an embodiment with a software server 108, as described above (and included by reference here), Devarajan discloses that the server 108 works (e.g., as a protection agent) to perform the blocking step 1010. *See also, e.g.*, Ex.1004 (Devarajan) at [0098] (“the intrusion prevention and detection module 907 may block the request for content from reaching the hosting server 903”). *See also, e.g.*, Ex.1004 (Devarajan) at [0101], [0112].

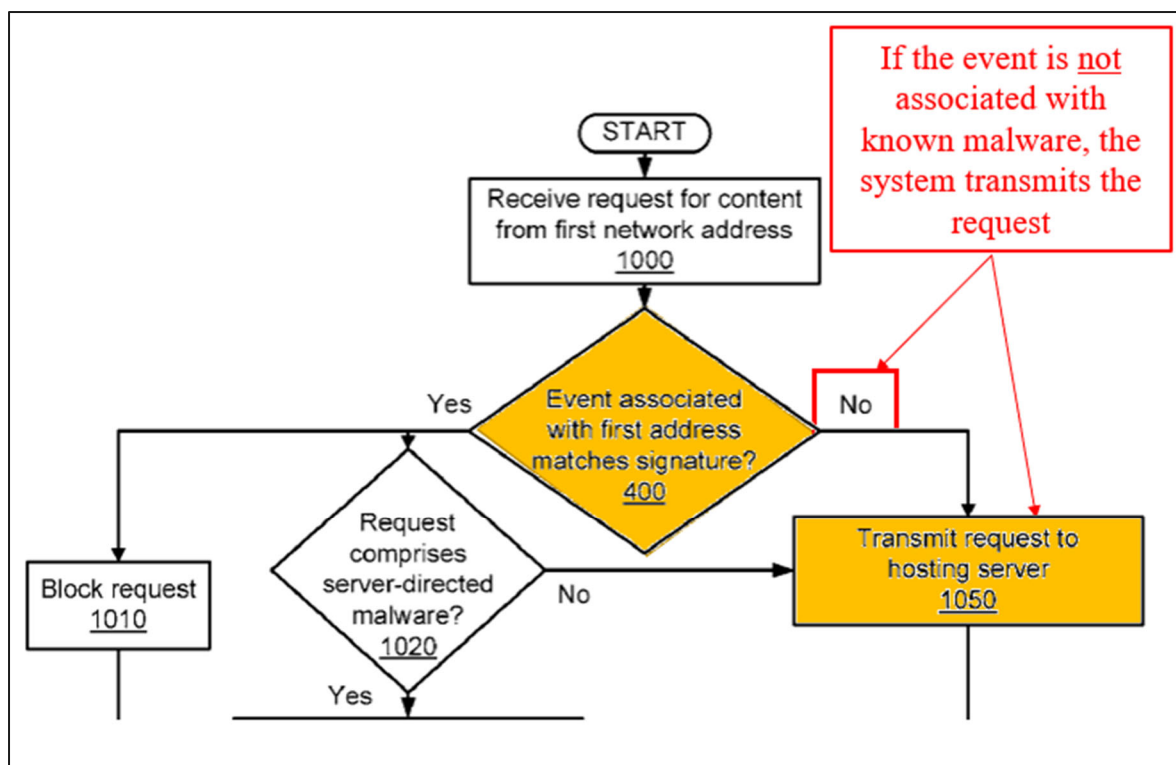


EX1004 (Devarajan) at Fig. 9 (annotated).

73. Thus, Devarajan teaches or renders obvious “when the request is associated with known malicious content, blocking, by the protection agent, the request.”

5. **[1D] when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers**

74. Devarajan discloses or renders obvious “when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers.” For example, Devarajan teaches sending requests to one or more web servers in the event malware is not detected.



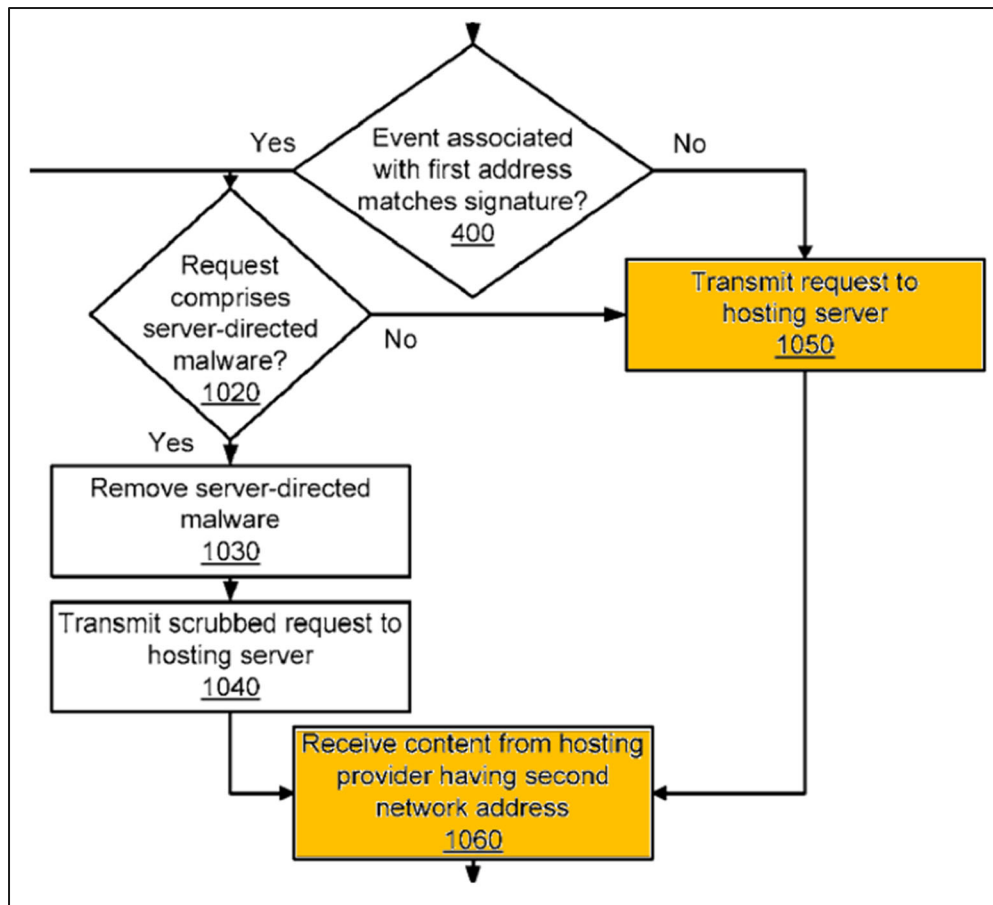
Ex.1004 (Devarajan) at Fig. 10 (annotated).

75. As shown in Figure 10 above, if at step 400 the system determines that “the event does not match an event signature 102, the request for content may be transmitted ...to the hosting server 903” as depicted in step 1050. Ex.1004 (Devarajan) at [0111]. The content is sent “responsive to receiving the request for content...*from the server 108*,” i.e., the “protection agent.” Ex.1004, [0089], [0115]. A POSITA would understand this to disclose “sending..., by the protection agent, the request to one or more web servers.”

76. Thus, it is my opinion that Devarajan discloses or renders obvious “when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers.”

6. **[1E] receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request;**

77. In my opinion, Devarajan discloses or renders obvious “receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request.” As discussed above, when the system in Devarajan determines that an event that comprises a request for web content is not associated with malicious content (e.g., because it does not match a signature for an IP address, URL, or domain name associated with malicious content, such as a phishing web site (*see, e.g.*, Ex.1004 (Devarajan) at [0046])), Devarajan discloses the sever 108 (e.g., a protection agent) sending a request to the content’s web server to retrieve the content (in step 1050 shown in Fig. 10, excerpted below).



Ex.1004 (Devarajan) at Fig. 10 (annotated).

78. Further, Devarajan discloses that once the request for content is transmitted to the hosting server, the illustrated method may further comprise the step of a (software) server 108, which includes the scrubbing center 906, receiving the content from the hosting server as depicted in step 1060 in Figure 10 (excerpted above) and explained in Devarajan as follows:

When website 902 content is returned to the client 900, it may be redirected to the scrubbing center 906 to ensure that, for example, any client-directed malware is removed. Where the client 900 initiates the content request, the

returned website 902 content may automatically be redirected to the scrubbing center 906. Where the hosting server 903 initiates a connection with the client 900, the SmartProxy 905 may intercept and redirect the traffic to the scrubbing center 906.

Ex.1004 (Devarajan) at [0107].

The illustrated method further may comprise the step of receiving the content from the hosting server 903 (Step 1060). In one embodiment, content may be received (perhaps at the server 108 running the scrubbing center 906) after the content is transmitted by the hosting server 903 responsive to receiving the request for content (scrubbed or un-scrubbed) from the server 108.

Ex.1004 (Devarajan) at [0115].

79. Further, as also explained above, the server 108 may be a software server in the same computer. Ex.1004 (Devarajan) at [0053]. For example, Devarajan discloses, “[e]ach of the at least one servers 108 (and/or any other server described herein) **could be any computer or program** that provides services to other computers, **programs**, or users **either in the same computer or over a computer network 101**. As non-limiting examples, the one of more server 108 could be application, communication, mail, database, proxy, fax, file, media, web, peer-to-peer, standalone, **software**, or hardware servers (i.e., server computers) and

may use any server format known in the art or developed in the future (possibly a shared hosting server, a virtual dedicated hosting server, a dedicated hosting server, or any combination thereof).” Ex.1004 (Devarajan) at [0053]. Thus, a POSITA would understand that a server 108 may be software executing, e.g., on a desktop computer, such as a client 900, and providing services to other software programs running on that computer. *See also, e.g.,* Ex.1004 (Devarajan) at [0092], which discloses the client 900 may be, e.g., a desktop computer or a laptop computer.

80. Further, as I explain above, a POSITA would find it obvious to choose this embodiment of the server 108 as a software server executing on the same computer (client 900) as the web browser. For example, a POSITA would understand that this embodiment reduces the communication time between a client and the server 108, since such communications would occur locally rather than going over a network, thereby improving latency for the overall system in Devarajan. Also, a POSITA would understand that placing the server 108 as a software server on the client 900, as described, e.g., in Ex.1004 (Devarajan) at [0053], would reduce the amount of hardware needed to implement the system of Devarajan, thereby reducing costs. Further, Devarajan discloses a SmartProxy 905 that “may act as an intermediary between the client 900 or edge server 910 and the hosting server 903,” and Devarajan shows the SmartProxy 905 as a component on the client 900. Ex.1004 (Devarajan) at [0105], Fig. 9. A POSITA would understand Devarajan’s description

of the functionality of the SmartProxy 905 to include communicating directly to the scrubbing center 906 (which is disclosed in Devarajan as a software module in the server 108, as shown in the excerpt below:

In another possible embodiment, a SmartProxy 905 may be used to divert traffic through the scrubbing center 906. **The SmartProxy 905 may comprise a proxy server application, software, or script that may run on an Internet user's client 900,** or perhaps on the network edge. **The SmartProxy 905 may communicate directly to the scrubbing center 906.** The SmartProxy 905 may act as an intermediary between the client 900 or edge server 910 and the hosting server 903. In one embodiment, the SmartProxy 905 may comprise an application, perhaps downloaded to the client 900 or edge server 910 from a scrubbing center 906 service provider, that ensures all traffic from and/or to the client 900 is routed to, and filtered through, the scrubbing center 906.

As a non-limiting example, the SmartProxy 905 may be configured (e.g., programmed) to receive website 902 content requests from the client 900 and redirect such requests to the scrubbing center 906. As a non-limiting example, the SmartProxy may accomplish this by storing the website 902 hosting server's 903 second network address 904 (e.g., IP address) in association with the scrubbing center 906 server's 108 third network resource

address (e.g., IP address), along with instructions to route requests for the website 902 to the scrubbing center 906.

When website 902 content is returned to the client 900, it may be redirected to the scrubbing center 906 to ensure that, for example, any client-directed malware is removed. Where the client 900 initiates the content request, the returned website 902 content may automatically be redirected to the scrubbing center 906. Where the hosting server 903 initiates a connection with the client 900, the SmartProxy 905 may intercept and redirect the traffic to the scrubbing center 906. As a non-limiting example, the SmartProxy may accomplish this by storing the scrubbing center 906 server's 108 third network resource address (e.g., IP address), along with instructions to route all incoming traffic to the scrubbing center 906 and request that the scrubbing center 906 return scrubbed content to the SmartProxy 905.

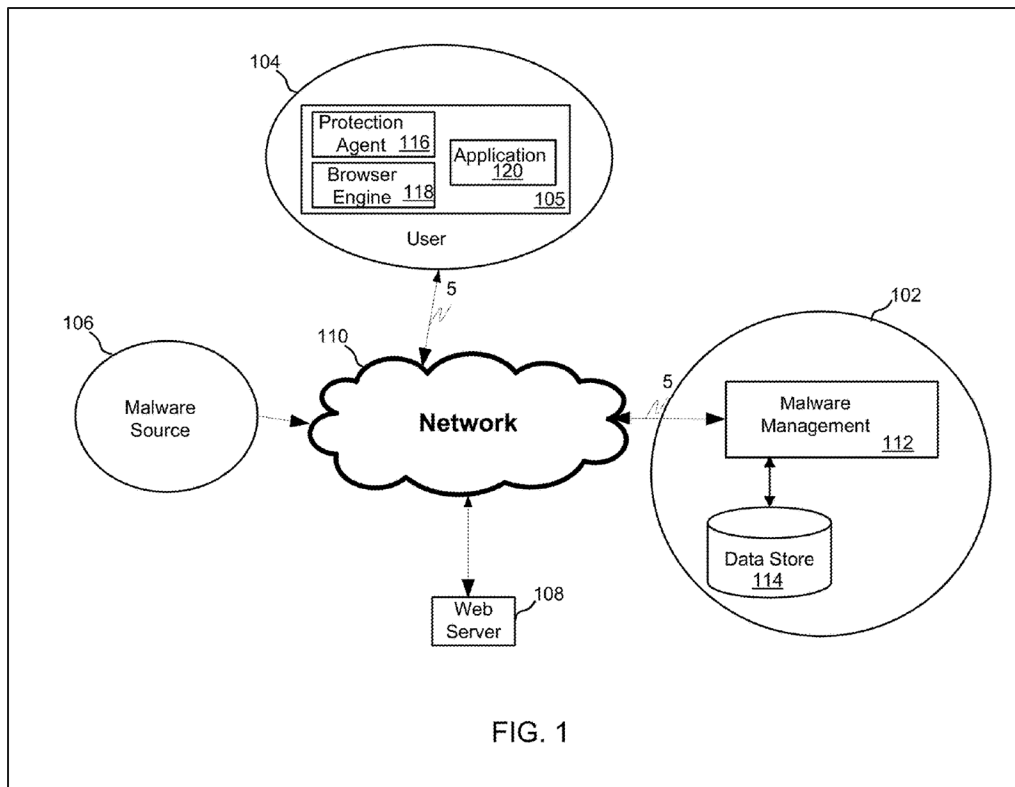
Ex.1004 (Devarajan)at [0105]-[0107].

81. Accordingly, a POSITA would understand Devarajan to disclose an embodiment where a SmartProxy 905 and server 108 are software executing on the client 900 computer (e.g., a desktop computer or laptop computer) to provide the functionality of the “protection agent”. *See, e.g.*, Ex.1004 (Devarajan) at [0095]. In my opinion, a POSITA would understand this embodiment in Devarajan to disclose

and/or suggest “a protection agent,” as required by claim 1, that in this embodiment comprises a (software) server 108 executing on the client 900 computer.

82. In addition, as shown above, Devarajan discloses the server 108 receiving web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request. Ex.1004 (Devarajan) at [0115].

83. Thus, Devarajan discloses the receiving step may be performed by a (software) server 108 (e.g., a protection agent) on the client 900’s computer (e.g., a “remote computer”). For instance, a POSITA would understand that the ’928 patent the ’928 Patent describes that the remote computer could be “a personal computer, a server of any time, a PDA, mobile phone, tablet, netbook, an interactive television, or any other device capable of loading and operating computer objects.” Ex.1001, 3:15-20. The ’928 Patent further discloses that the “remote computer” contains a protection agent and a web browser application, as recited in claim 1 and shown in its Figure 1; hence, a POSITA would understand “remote computer” to have its plain-and-ordinary meaning in light of the claims. *See, e.g.*, Ex.1001 (’928 patent) at Fig. 1 (at block 105), excerpted below, and at 3:2-20.



Ex.1001 ('928 patent) at Fig. 1.

84. In addition, Devarajan discloses that the website 902 “may comprise any collection of data and/or files accessible to a client 900 or server 108 communicatively coupled to the network 101. As a non-limiting example, the website 902 may comprise a single webpage or multiple interconnected and related webpages, perhaps resolving from a domain name, each of which may provide access to static, dynamic, multimedia, or any other content, perhaps by accessing files (e.g., text, audio, video, graphics, executable, HTML, eXtensible Markup Language (XML). Active Server Pages (ASP), Hypertext Preprocessor (PHP), Flash files, server-side scripting, etc.) that enable the website 902 to display when rendered by a browser on a client 900.” Ex.1004 (Devarajan) at [0090]. Accordingly, a

POSITA would understand that the web content 902 intercepted by the server 108, from the host server 903 in response to a query to the host server 903 originating from the client 900 (as explained above) comprises “data for assembling a web page.”

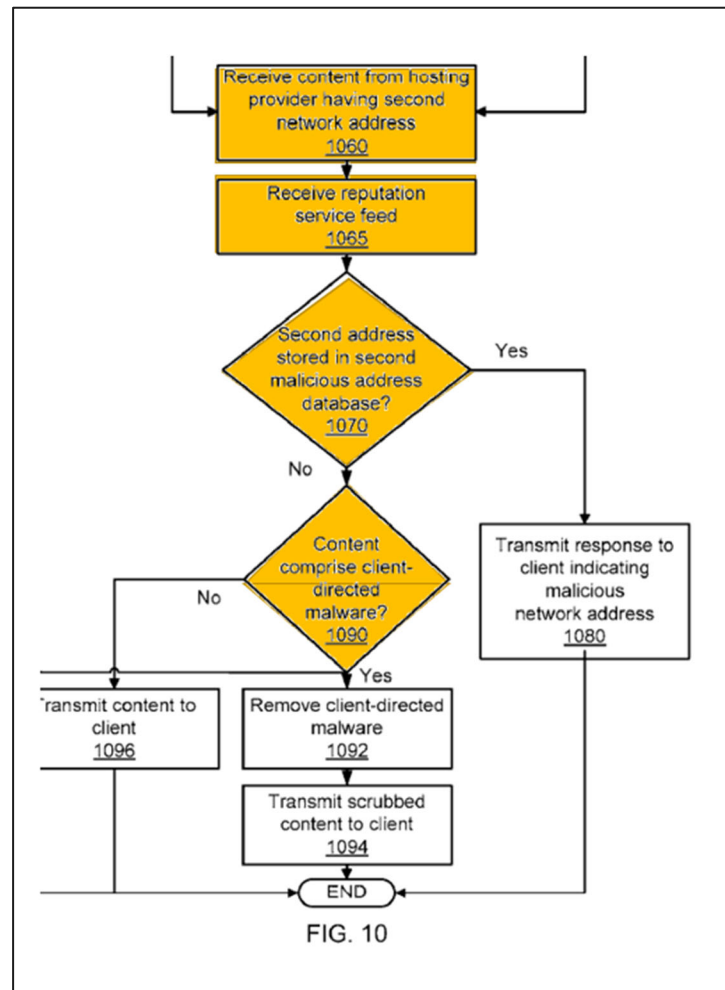
85. As such, Devarajan discloses or renders obvious “receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request.”

7. [1F] identifying, by the protection agent, a malware threat within the web content;

86. In my opinion, Devarajan discloses or renders obvious “identifying, by the protection agent, a malware threat within the web content.” Following the method described in the Figure 10 embodiment disclosed in Devarajan, once the server 108 transmits the request to the hosting server 903, in response, the server 108 receives the web content from the hosting server 903. Ex.1004 (Devarajan) at [0107], [0115].

87. At step 1065 illustrated in Figure 10 below, “a reputation feed may be received...from a...provider...having a second malicious network resource address database.” Ex.1004 (Devarajan) at [0116]. At the next step (1070), the system determines whether the “hosting server’s 903 network resource address 904 is stored

in the second malicious network resource address database.” Ex.1004 (Devarajan) at [0117].



Ex.1004 (Devarajan) at Fig. 10 (annotated).

88. If the system determines that the network address is not stored in the database of known malware, it moves on to step 1090, which comprises “determining whether the content comprises a client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a phishing application, a script injection, or an iFrame injection).” Ex.1004

(Devarajan) at [0118]. If the content comprises client-directed malware, then the system moves on to step 1092, which involves “removing the client-directed malware from the content...and transmitting a scrubbed content (e.g., website content having the client-directed malware removed) to the client. Ex.1004 (Devarajan) at [0118]. Thus, if the system follows the “yes” branch of step 1090, it has performed a step of identifying, by the protection agent, a malware threat within the web content. Moreover, Devarajan discloses that this method may be performed by a (software) server 108 (e.g., a protection agent) on the client 900, as I explain above.

89. As such, Devarajan discloses or renders obvious “identifying, by the protection agent, a malware threat within the web content.”

8. [1G] in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content;

90. In my opinion, Devarajan discloses or renders obvious “in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content.” As discussed above, Devarajan discloses modifying the web content in response to detecting malware such that the modified webpage no longer comprises malware.

91. Namely, if the system of Devarajan determines that the network address is not stored in the database of known malware, it moves on to step 1090, which comprises “determining whether the content comprises a client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a phishing application, a script injection, or an iFrame injection).” Ex.1004 (Devarajan) at [0118]. If the content comprises client-directed malware, then the system moves on to step 1092, which involves “removing the client-directed malware from the content...and transmitting a scrubbed content (e.g., website content having the client-directed malware removed) to the client. Ex.1004 (Devarajan) at [0118]. Moreover, as discussed above and include by reference here, the component performing these steps may be a software server 108 (e.g., a protection agent) executing on the same computer, that is, the client 900.

92. The '928 patent explains that that modifying a HTML document is an example of a “protocol-level” modification. Ex.1001 ('928 patent) at 1:43-48, 4:6-11. A POSITA would understand that HTML is HyperText Markup Language used for web pages and that it includes text and/or links to other web content. *See, e.g.,* Ex.1001 ('928 patent) at 4:56-60. Devarajan discloses, “[i]n another example, if the intrusion detection and prevention module 907 identifies a link to a known malware website in the content, the content sanitizer module 909 may either block the content or remove the link from the content. After the malware has been removed, the

content sanitizer module 909 may transmit a ‘scrubbed’ content (e.g., the content having the client-directed malware removed) to the client 900.” Ex.1004 (Devarajan) at [0104]. A POSITA would therefore understand this disclosure regarding removing a link from web content in Devarajan to be performing a modification of web content at the protocol level identified in the ’928 patent.

93. Thus, it is my opinion that Devarajan discloses or renders obvious “in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content.”

- 9. [1H] assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and**

94. In my opinion, Devarajan discloses or renders obvious “assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat.” As discussed above, Devarajan teaches modifying the web content and transmitting the scrubbed content sans malware to the user. Moreover, a POSITA would understand that the modifications described, e.g., in Ex.1004 (Devarajan) at [0104], as described above, comprise assembling a modified version of a web page, given the context of removing links from web content. Devarajan also discloses, “[w]ebsites may be

created using HyperText Markup Language (HTML) to generate a standard set of tags that define how the webpages for the website are to be displayed. Such websites may comprise a collection of HTML and subordinate documents (i.e., files) stored on the Web that are typically **accessible** from the same Uniform Resource Locator (URL) and reside on the same server, although such files may be distributed in numerous servers.” Ex.1004 (Devarajan) at [0023].

95. Further, as described above, Devarajan discloses performing the steps 1090 and 1092, as shown in Figure 10, by the server 108 (e.g., a protection agent).

96. Thus, Devarajan discloses or renders obvious “assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat.”

10. [1I] providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.

97. In my opinion, Devarajan discloses or renders obvious “providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.” As discussed above, Devarajan discloses the server 108 (e.g., a protection agent) modifying the webpage using the content sanitizer module and “after the malware has been removed” transmitting the “scrubbed content” to the client (e.g., in step 1094 of the method

shown in Fig. 10). Ex.1004 (Devarajan) at [0118]. Further, Devarajan discloses, “[t]he website 902 may comprise any collection of data and/or files accessible to a client 900 or server 108 communicatively coupled to the network 101. As a non-limiting example, the website 902 may comprise a single webpage or multiple interconnected and related webpages, perhaps resolving from a domain name, each of which may provide access to static, dynamic, multimedia, or any other content, perhaps by accessing files (e.g., text, audio, video, graphics, executable, HTML, eXtensible Markup Language (XML). Active Server Pages (ASP), Hypertext Preprocessor (PHP), Flash files, server-side scripting, etc.) that enable the website 902 to display when rendered by a browser on a client 900.” Ex.1004 (Devarajan) at [0090].

98. Thus, Devarajan discloses or renders obvious “providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.”

99. Therefore, Devarajan renders claim 1 of the '928 patent obvious.

B. Claim 2

1. The method of claim 1, further comprising supplementing, by the protection agent, the web content with indicators regarding a vulnerability of one or more links within the web content.

100. Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising supplementing, by the protection agent, the web content with

indicators regarding a vulnerability of one or more links within the web content.” I incorporate by reference my analysis for claim 1.

101. As discussed above, Devarajan discloses the server 108 receiving the web content (in step 1060) from the hosting server 903 in response to receiving a request from the server 108 for content. Ex.1004 (Devarajan) at [0115]. Devarajan further discloses that a reputation feed may be received (in step 1065), as described elsewhere in Devarajan:

A reputation feed then may be received (Step 1065), perhaps from a network resource address reputation service provider (e.g., a reputation service module 908) having a second malicious network resource address database 112. The reputation feed may be implemented as described in detail above with respect to the systems and methods for providing a network resource address reputation service.

Ex.1004 (Devarajan) at [0116].

102. A POSITA would understand that it is common for web content coming from a hosting server 903 to contain links to sections within the same webpage or to other webpages that are also hosted at the same hosting server. Such links would, of course, have the same resource address as the hosting server 903. Further, Devarajan discloses the method of Figure 10 (which may be executed by the server 108, as I explain above) including a step of transmitting a message (i.e., an indication) to the

client 900 that the hosting server or its address have been associated with a malicious network address:

If it is determined that the hosting server's 903 network resource address 904 is stored in the second malicious network resource address database 112 (Step 1070), a response may be transmitted to the client 900 indicating that the second network resource address 904 is stored in the second malicious network resource address database 112 (Step 1080). As one non-limiting example, the content may be blocked and a HTTP 404 error code may be transmitted back to the requesting client 900. Or the client 900 may be transmitted a message indicating that the hosting server 903 or its IP address (i.e., second network resource address 904) have been associated with a malicious network address.

Ex.1004 (Devarajan) at [0117].

103. Thus, a POSITA would understand that in the likely instances where web content includes links to content also hosted at hosting server 903, the server 108 would supplement the web content being returned to the client 900 with indicators regarding a vulnerability of one or more links within the web content.

104. Further, Devarajan discloses removing links from web content to a known malware website before the content is returned to the client 900:

In another example, if the intrusion detection and prevention module 907 identifies a link to a known malware website in the content, the content sanitizer module 909 may either block the content or remove the link from the content. After the malware has been removed, the content sanitizer module 909 may transmit a “scrubbed” content (e.g., the content having the client-directed malware removed) to the client 900.

Ex.1004 (Devarajan) at [0104].

105. In an embodiment, Devarajan also discloses scoring the reputation of links with a range from 0 to 1, 1 to 10, 0% to 100%, A+ to F-, or a color rating system, e.g., with “red” indicating a “poor reputation.”

Responsive to a determination that the event associated with the network resource address 104 matches an event signature(s) 102, the illustrated method further may comprise generating a reputation score for the network resource address 104 (Step 410). As a non-limiting example, this step may be accomplished by a server 108 configured to generate a reputation score for the network resource address 104 by having instructions 110 installed in computer readable media 109 causing the microprocessor 111 to generate such a reputation score. The reputation score may comprise any score indicating the reputation for the network resource address 104 and may comprise any rating or ranking scale known in the art

or developed in the future. As non-limiting examples, the link match score may range from 0 to 1, 1 to 10, 0% to 100%, and/or A+ to F- (e.g., grades). Alternatively, it may comprise a star rating system or a color rating system (e.g., red indicates a poor reputation, yellow indicates an average reputation, and green indicates a good reputation).

Ex.1004 (Devarajan) at [0074].

106. Thus, even in instances where web content does not contain links to content hosted by the hosting server 903 (hence, would be a link with the same resource address as the hosting server), given Devarajan's teaching about removing links to known malware sites and Devarajan's teaching about supplementing links with reputation scores (such as numbers, letter grades, or colors), a POSITA would find it obvious from Devarajan to supplement the web content (e.g., for links with poor reputation but not necessarily known malware) with link reputation scores as taught in Devarajan. Devarajan further explains that the reputation score may be modified "to indicate a more negative reputation for the network resource address." Ex.1004 (Devarajan) at [0076]. "The modified reputation then may be stored...in the second malicious network resource address database 112." Ex.1004 (Devarajan) at [0076]. After generating a reputation score step 1065 and storing that score in the second malicious network resource address database at step 1070, Devarajan discloses that "a response may be transmitted to the client 900 indicating that the

second network resource address 904 is stored in the second malicious network resource address database 112 (Step 1080).” Ex.1004 (Devarajan) at [0076], [0117]. Devarajan also teaches “supplementing...the web content with indicators regarding a vulnerability of one or more links within the web content,” wherein Devarajan “identifies a link to a known malware website in the content” (e.g., identifies “a vulnerability of one or more links within the web content”) and “either block(s) the content or remove(s) the link from the content” (e.g., “supplementing...the web content with indicators”). Ex.1004 (Devarajan) at [0104]. The blocking or removal of the links are examples of “indicators.” Ex.1004 (Devarajan) at [0104]. Thus, Devarajan teaches supplementing “the web content with indicators” as required by the claim elements.

107. Thus, Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising supplementing, by the protection agent, the web content with indicators regarding a vulnerability of one or more links within the web content.”

C. Claim 3

- 1. The method of claim 1, wherein an analysis to identify a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis.**

108. Devarajan discloses or renders obvious “[t]he method of claim 1, wherein an analysis to identify a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol)

analysis, an image analysis, and a script/HTML analysis.” I incorporate by reference my analysis for claim 1.

109. For example, Devarajan discloses that in the event that “the hosting server’s 903 network resource address 904 is stored in the second malicious network resource address database 112 (Step 1070), a response may be transmitted to the client 900... indicating that the hosting server 903 or its IP address (i.e., second network resource address 904) have been associated with a malicious network address.” Ex.1004 (Devarajan) at [0117]. A POSITA would understand this to mean that the system performs an IP (Internet Protocol) analysis to determine the presence of malware.

110. In addition, Devarajan also discloses identifying a malware threat using script and/or HTML (e.g., link) analysis:

In another example, if the intrusion detection and prevention module 907 **identifies a link to a known malware website in the content**, the content sanitizer module 909 may either block the content or remove the link from the content. After the malware has been removed, the content sanitizer module 909 may transmit a “scrubbed” content (e.g., the content having the client-directed malware removed) to the client 900.

Ex.1004 (Devarajan) at [0104].

Alternatively, rather than transmit an error (or content unavailable) message to the client 900, the illustrated method further may comprise determining whether the content comprises a client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a phishing application, **a script injection**, or an iFrame injection) (Step 1090) and, if so, removing the client-directed malware from the content (Step 1092) and transmitting a scrubbed content (e.g., website content having the client-directed malware removed) to the client 900 (Step 1094).

Ex.1004 (Devarajan) at [0118].

111. As such, Devarajan discloses or renders obvious “[t]he method of claim 1, wherein an analysis to identify a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis.”

D. Claim 4

- 1. The method of claim 1, further comprising determining, by the protection agent, whether a request for the web content from the web browser application is a first request.**

112. Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising determining, by the protection agent, whether a request for the web content from the web browser application is a first request.” I incorporate by reference my analysis for claim 1.

113. For example, Devarajan discloses blocking a request for web content if the hosting server's network address is stored in a malicious resource address database 112:

If it is determined that the hosting server's 903 network resource address 904 is stored in the second malicious network resource address database 112 (Step 1070), a response may be transmitted to the client 900 indicating that the second network resource address 904 is stored in the second malicious network resource address database 112 (Step 1080). As one non-limiting example, the content may be blocked and a HTTP404 error code may be transmitted back to the requesting client 900. Or the client 900 may be transmitted a message indicating that the hosting server 903 or its IP address (i.e., second network resource address 904) have been associated with a malicious network address.

Ex.1004 (Devarajan) at [0117].

114. Further, Devarajan discloses adding a network resource address to a blacklist for blocked addresses in a malicious network research address database 112 if it is determined to have a reputation score above a threshold, thereby precluding future communication with that network resource address.

FIG. 5 illustrates an alternate embodiment of a method of providing a network resource address reputation service

that builds upon that illustrated in FIG. 4 and further may comprise determining whether the network resource addresses' 104 reputation score exceeds a predetermined value (Step 500) and, if so, adding the network resource address 104 to a blacklist (Step 510), perhaps stored in the second malicious network resource address database 112 or any other network storage device or computer memory communicatively coupled to the network 101. For example, if the predetermined value is 50%, any network resource address having a worse reputation score than 50% may be added to the blacklist, perhaps resulting in blocked connection to—or otherwise precluding communication with—that network resource address.

Ex.1004 (Devarajan) at [0077].

115. Thus, a POSITA would understand that Devarajan is disclosing that a first request for web content from a web browser to a network resource address with a poor reputation score will in a first instance be considered and the system will add that network resource address to a blacklist, thereby precluding future communication with that network resource address. That is, depending on whether a network resource address for a host server with a poor reputation score is in the blacklist or not, the server 108 (e.g., a protection agent) will be determining whether this request for content is a first request. Therefore, Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising determining, by the protection

agent, whether a request for the web content from the web browser application is a first request.”

E. Claim 6

1. The method of claim 1, wherein a protocol-level analysis is utilized to identify the malware threat within the web content.

116. In my opinion, Devarajan discloses or renders obvious “[t]he method of claim 1, wherein a protocol-level analysis is utilized to identify the malware threat within the web content.” As discussed above, Devarajan discloses modifying the web content in response to detecting malware such that the modified webpage no longer comprises malware, with the modification based on protocol-level analysis. I incorporate by reference my analysis for claim 1.

117. Further, Devarajan discloses performing protocol-level analysis prior to this modification step. For example, if the system of Devarajan determines that the network address is not stored in the database of known malware, it moves on to step 1090, which comprises “determining whether the content comprises a client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a phishing application, a script injection, or an iFrame injection).” Ex.1004 (Devarajan) at [0118]. If the content comprises client-directed malware, then the system moves on to step 1092, which involves “removing the client-directed malware from the content...and transmitting a scrubbed content (e.g., website content having the client-directed malware removed) to the client. Ex.1004

(Devarajan) at [0118]. Moreover, as discussed above and include by reference here, the component performing these steps may be a software server 108 (e.g., a protection agent) executing on the same computer, that is, the client 900.

118. The '928 patent discloses that analyzing an HTML document is inclusive of a “protocol-level” analysis. Ex.1001 ('928 patent) at 1:43-48, 4:6-11. A POSITA would understand that HTML is HyperText Markup Language used for web pages and that it includes text and/or links to other web content. *See, e.g.,* Ex.1001 ('928 patent) at 4:56-60. Devarajan discloses, “[i]n another example, if the intrusion detection and prevention module 907 identifies a link to a known malware website in the content, the content sanitizer module 909 may either block the content or remove the link from the content. After the malware has been removed, the content sanitizer module 909 may transmit a ‘scrubbed’ content (e.g., the content having the client-directed malware removed) to the client 900.” Ex.1004 (Devarajan) at [0104]. A POSITA would therefore understand this disclosure regarding identifying a link to a known malware website in web content in Devarajan to be performing an analysis of web content at the protocol level identified in the '928 patent.

119. Thus, Devarajan discloses or renders obvious, “[t]he method of claim 1, wherein a protocol-level analysis is utilized to identify the malware threat within the web content.”

F. Claim 7

- 1. The method of claim 3, further comprising performing cloud verification of the web content, wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses.**

120. Devarajan discloses or renders obvious “[t]he method of claim 3, further comprising performing cloud verification of the web content, wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses.” I incorporate my analysis for claims 1 and 3.

121. Further, Devarajan discloses that the instructions for the server 108 can be done in a cloud-computing environment, which a POSITA would understand to imply the “cloud verification of web content” limitation of claim 7.

As further illustrated in FIG. 1, the server 108 may comprise a computer-readable storage media 109 storing instructions 110 that, when executed by a microprocessor 111, cause the server 108 to perform the steps for which it is configured. The computer-readable media 109 may comprise any data storage medium capable of storing instructions 110 for execution by a computing device. It may comprise, as non-limiting examples, magnetic, optical, semiconductor, paper, or any other data storage media, a database or other network storage device, hard disk drives, portable disks, CD ROM, DVD, RAM, ROM, flash memory, and/or holographic data storage. The instructions 110 may, as non-limiting examples, comprise

software and/or scripts stored in the computer-readable media 109 that may be stored locally in the server 108 or, alternatively, in a highly-distributed format in a plurality of computer-readable media 109 accessible via the network 101, perhaps via a grid or **cloud-computing environment**.

Ex.1004 (Devarajan) at [0054].

122. Further, Devarajan discloses the server 108 performing determining malware vulnerabilities in web content based on an aggregation of analyses that includes “determining whether the content comprises a client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a phishing application, a script injection, or an iFrame injection) (Step 1090).”

Ex.1004 (Devarajan) at [0118].

123. Thus, Devarajan discloses or renders obvious “[t]he method of claim 3, further comprising performing cloud verification of the web content, wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses.”

G. Claim 9

1. [9P] A content analysis and malware prevention system comprising:

124. This claim is disclosed or rendered obvious by Devarajan. *See* discussion above for claim [1P].

2. [9A] at least one processor; and

125. Devarajan discloses a processor. For example, Devarajan states, “[a]s further illustrated in FIG. 1, the server 108 may comprise a computer-readable storage media 109 storing instructions 110 that, when executed by a microprocessor 111, cause the server 108 to perform the steps for which it is configured.” Ex.1004 (Devarajan) at [0054]. Thus Devarajan discloses a “processor.”

3. [9B] memory encoding computer executable instructions that, when executed by the at least one processor, perform a method comprising:

126. Devarajan discloses “memory encoding computer executable instructions that, when executed by the at least one processor, perform a method comprising.” For example, Devarajan states, “[a]s further illustrated in FIG. 1, the server 108 may comprise a computer-readable storage media 109 storing instructions 110 that, when executed by a microprocessor 111, cause the server 108 to perform the steps for which it is configured.” Ex.1004 (Devarajan) at [0054].

4. [9C] intercepting a request from a web browser;

127. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1A].

5. [9D] determining whether the request is associated with known malicious content;

128. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1B].

6. **[9E] when the request is associated with known malicious content, blocking, by a protection agent, the request;**

129. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1C].

7. **[9F] when the request is not associated with known malicious content: sending the request to one or more web servers;**

130. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1D].

8. **[9G] receiving web content from one or more web servers, wherein web content comprises data for assembling a web page;**

131. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1E].

9. **[9H] identifying a malware threat within the web content;**

132. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1F].

10. **[9I] in response to identifying the malware threat, modifying the web content, wherein modifying the web content comprises modifying the web at protocol level to remove the malware threat from the web content;**

133. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1G].

11. **[9J] assembling a modified version of the web page, wherein the modified web page is assembled using the modified web content,**

and wherein the modified version of the web page does not comprise the malware threat; and

134. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1H].

12. [9K] providing the modified version of the web page to a web browser application on the remote computer for rendering and display.

135. This claim is rendered obvious by Devarajan. *See* discussion above for claim [1I].

H. Claim 10

1. The system of claim 9, wherein the method further comprises supplementing the web content with indicators regarding a vulnerability of one or more links within the web content.

136. This claim is rendered obvious by Devarajan. *See* discussion above for claim 2.

I. Claim 11

1. The system of claim 9, wherein the method further comprises identifying the malware threat within the web content by performing one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis.

137. This claim is rendered obvious by Devarajan. *See* discussion above for claim 3.

J. Claim 12

- 1. The system of claim 9, wherein the method further comprises determining whether a request for the web content from the web browser application is a first request.**

138. This claim is rendered obvious by Devarajan. *See* discussion above for claim 4.

K. Claim 14

- 1. The system of claim 9, wherein the method further comprises identifying the malware threat utilizing a protocol-level analysis.**

139. This claim is rendered obvious by Devarajan. *See* discussion above for claim 6.

L. Claim 16

- 1. The method of claim 1, wherein the malware threat comprises a Uniform Resource Locator (URL).**

140. Devarajan discloses or renders obvious “[t]he method of claim 1, wherein the malware threat comprises a Uniform Resource Locator (URL).” I incorporate by reference my analysis for claim 1.

141. For example, Devarajan teaches an intrusion detection and prevention module, which analyzes the content on a page if it “identifies a link to a known malware website in the content, the content sanitizer module 909 may either block the content or remove the link from the content.” Ex.1004 (Devarajan) at [0104]. A POSITA would understand that in this context a “link” to be inclusive of a URL. Devarajan further teaches, “[a]n external feed 107 may comprise a malware domain

list feed, a malware URL list feed, an emerging threat feed, an intrusion detection feed, a botnet tracking feed, a phishing tracking feed, a spam tracking feed, or a compromised network feed.” Ex.1004 (Devarajan) at [0052]. In addition, Devarajan teaches, “[a]s non-limiting examples, the network resource address may comprise an IP address, a URL, or a domain (e.g., domain name) for the network resource 103.” Ex.1004 (Devarajan) at [0049]. As I explain above and include by reference here, the analysis of Devarajan is inclusive of determining whether network resource addresses are associated with known malware. Given the teachings in Devarajan, e.g., at [0049], [0052], [0104], a POSITA would understand that this analysis comprises a URL. Thus, Devarajan discloses or renders obvious “[t]he method of claim 1, wherein the malware threat comprises a Uniform Resource Locator (URL).”

M. Claim 17

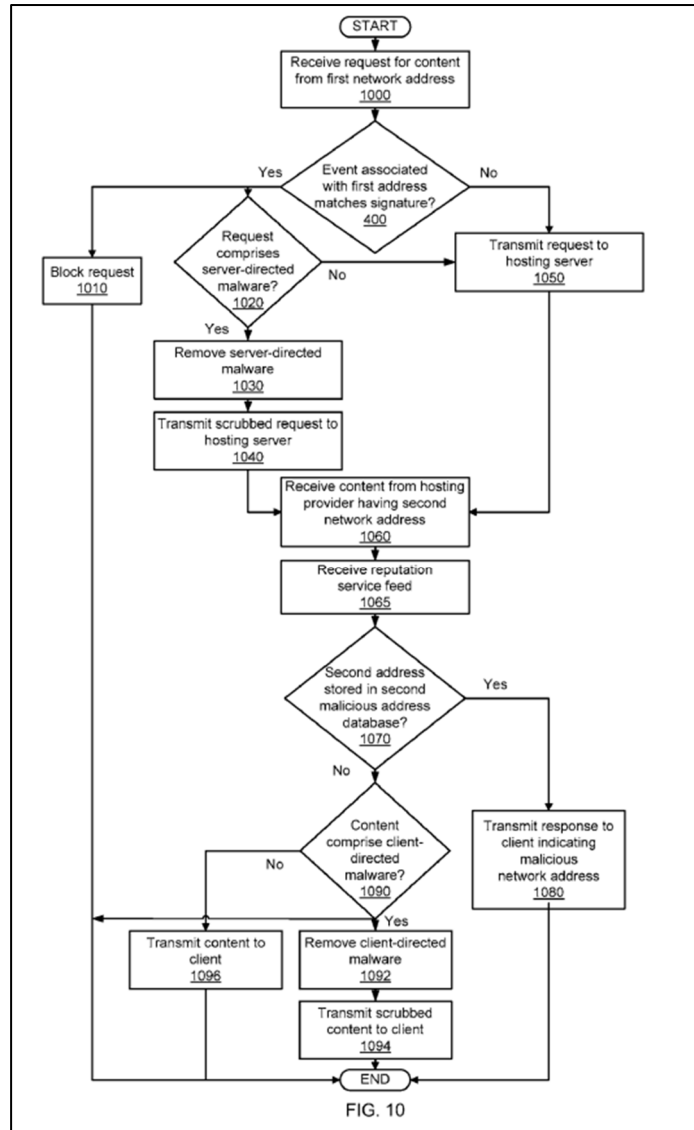
- 1. The system of claim 9, wherein the method further comprises determining if a malware threat comprises a Uniform Resource Locator (URL).**

142. This claim is rendered obvious by Devarajan. *See* discussion above for claim 16.

N. Claim 18

- 1. The method of claim 1, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.**

143. Devarajan teaches “[t]he method of claim 1, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.” I incorporate by reference my analysis for claim 1. For example, Figure 10 in Devarajan illustrates a “method for bi-directional detection and removal of network traffic malware”. Ex.1004 (Devarajan) at [0109].



Ex.1004 (Devarajan) at Fig. 10

144. As shown in Figure 10, the process begins when the system “receive[s]...a request for content” as shown in block 1000. Ex.1004 (Devarajan) at [0109]. At the next block (400), the system determines “whether an event associated with the client 900 or its first network resource address 901 matches one or more of a plurality of event signatures 102.” Ex.1004 (Devarajan) at [0111].

These “signatures 102” are “simply known attack patterns” and “comprise a plurality of malware signatures including, but not limited to a virus signature, a worm signature, a trojan horse signature, a rootkit signature, a backdoor signature, a spyware signature, a keystroke logger signature, or a phishing application signature.” Ex.1004 (Devarajan) at [0046]. If at step 400 the system determines that “the event does not match an event signature 102, the request for content may be transmitted ...to the hosting server 903” as depicted in step 1050. Ex.1004 (Devarajan) at [0111]. Moreover, Devarajan discloses the following:

As a non-limiting example, the SmartProxy 905 may be configured (e.g., programmed) to receive website 902 content **requests** from the client 900 and redirect such **requests** to the scrubbing center 906. As a non-limiting example, the SmartProxy may accomplish this by storing the website 902 hosting server’s 903 second network address 904 (e.g., IP address) in association with the scrubbing center 906 server’s 108 third network resource address (e.g., IP address), along with instructions to route **requests** for the website 902 to the scrubbing center 906.

Ex.1004 (Devarajan) at [0106]; *see also, e.g.*, claims 17 and 21 in Devarajan.

145. A POSITA would understand the above disclosures to mean a request for content may be sent iteratively by the server 108 to comprise a plurality of

requests to the hosting server 903. Thus, it is my opinion that Devarajan discloses and renders obvious “[t]he method of claim 1, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.”

O. Claim 19

1. The method of claim 1, further comprising processing the web content prior to identifying the malware threat within the web content.

146. Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising processing the web content prior to identifying the malware threat within the web content.” I include by reference my analysis for claim 1.

147. Further, Devarajan discloses processing the web content by analyzing its IP address prior to determining whether or not it comprises malware. Ex.1004 (Devarajan) at [0117].

148. Thus, Devarajan discloses or renders obvious “[t]he method of claim 1, further comprising processing the web content prior to identifying the malware threat within the web content.”

P. Claim 20

- 1. The system of claim 9, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.**

149. This claim is rendered obvious by Devarajan. *See* discussion above for claim 18.

X. GROUND 2: CLAIMS 1-4, 6-7, 9-12, 14, 16-20 ARE OBVIOUS OVER DEVARAJAN IN VIEW OF HUBBARD

A. Claim 1

- 1. [1P] A content analysis and malware prevention method comprising:**

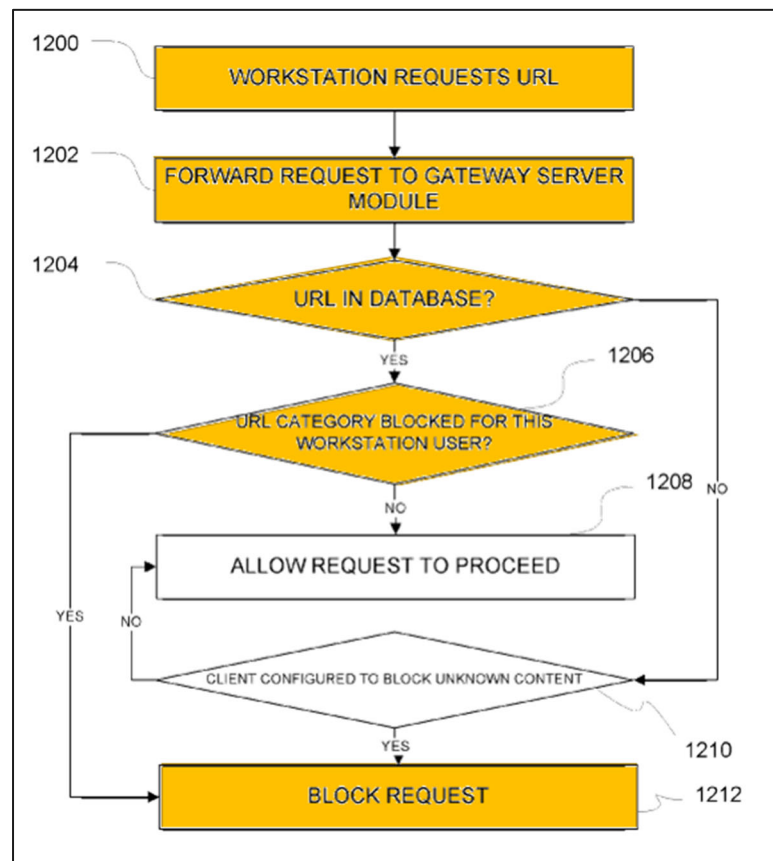
150. Claim element [1P] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.1, above.

- 2. [1A] intercepting, by a protection agent, a request from a web browser;**

151. Devarajan discloses or renders obvious “*intercepting, by a protection agent, a request from a web browser.*” *See* Section IX.A.2, above.

152. Further, to the degree that the Board disagrees with my analysis given above that Devarajan alone renders obvious claim 1A, Hubbard also discloses “*intercepting . . . a request from a web browser.*” Hubbard discloses a system that comprises a “gateway server module 120,” which is a “software module that is installed on a server that stands as a gateway to a wider area network 112.” Ex.1008 (Hubbard) at 4:54-58. The “gateway server module 120” is connected to

“workstation 116,” which comprises “various types of client computers that are attached to the network” 110. Ex.1008 (Hubbard) at 4:41-44. The “gateway server module” is “in two way communication with the workstation 116,” and “requests originating within the workstations 116 of the network 110 may be required to pass through the gateway server module 120 as the proceed to the Internet.” Ex.1008 (Hubbard) at 6:23-26. This request “is **intercepted** . . . and forwarded to the gateway server module 120 at block 1202.” Ex.1008 (Hubbard) at 22:54-56. At block 1206, the gateway server module determines “whether the category of the URL is one that should be blocked for the workstation user.” Ex.1008 (Hubbard) at 22:58-62.



Ex.1008 (Hubbard) at Fig. 16 (annotated).

153. As I explain above, and include by reference here, POSITA would be motivated to combine the (“*protection agent*”) server 108 of Devarajan with the “*intercepting*” methods of Hubbard because it would allow the system to further its goal of remediating malware before it reaches the user.

154. Like Devarajan, Hubbard teaches systems and methods for “controlling access to web content” in order to avoid URL links “that lead to webpages that are categorized as being ‘Malicious’ or ‘Spyware.’” Ex.1008 (Hubbard) at 2:41-42, 8:46-49. Hubbard teaches that the increased sophistication of newer web applications has created an environment where “malicious code and applications. . . [can] download themselves to users’ machines” without any affirmative action by the user. Ex.1008 (Hubbard) at 1:53-2:2. Hubbard notes that traditional malware blocking methods like “heightened security setting on [a] web browser” are impractical as they “impede or obstruct effect use of [various web applications].” Ex.1008 (Hubbard) at 2:8-15. Thus, Hubbard discloses a need for malware detection and potential removal without compromising the efficiency of users’ web based applications. Devarajan detects malware based on known malware network addresses stored in its databases, and in the event malware is detected, Devarajan teaches removing the malware from the content request. Ex.1004 (Devarajan) at [0004], [0007]. Devarajan’s malware detection system can detect malware using “signature-based, statistical anomaly-based, and stateful protocol analysis methods.”

Ex.1004 (Devarajan) at [0044]. Hubbard further teaches URL analysis and identification. Ex.1008, 2:41-52. Devarajan notes that a “URL may include a Hypertext Transfer Protocol (HTTP) request combined with the website’s Internet address.” Ex.1004 (Devarajan) at [0026]. To further enhance Devarajan’s protocol analysis malware detection system, a POSITA in the field of network security would have had a reasonable expectation of success in turning to a reference like Hubbard, which specifically teach analyzing and categorizing URLs. A POSITA could easily adapt Hubbard’s analysis methods to work with the protection agent system taught by Devarajan, because the systems would work together.

155. As such, Devarajan in view of Hubbard renders this limitation obvious.

3. [1B] determining, by the protection agent, whether the request is associated with known malicious content;

156. Claim element [1B] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.3, above.

4. [1C] when the request is associated with known malicious content, blocking, by the protection agent, the request;

157. Claim element [1C] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.4, above.

5. [1D] when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers

158. Claim element [1D] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.4, above.

6. **[1E] receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request;**

159. Claim element [1E] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.6, above.

7. **[1F] identifying, by the protection agent, a malware threat within the web content;**

160. Claim element [1F] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.7, above.

8. **[1G] in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content;**

161. Claim element [1G] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.8, above.

9. **[1H] assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and**

162. Claim element [1H] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.9, above.

10. **[1I] providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.**

163. Claim element [1I] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.A.10, above.

B. Claim 2

164. Claim 2 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.B, above.

C. Claim 3

165. Claim 3 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.C above.

D. Claim 4

166. Claim 4 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.D, above.

E. Claim 6

167. Claim 6 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.E, above.

F. Claim 7

168. Claim 7 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.F, above.

G. Claim 9

1. [9P] A content analysis and malware prevention system comprising:

169. Claim element [9P] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.1, above.

2. [9A] at least one processor; and

170. Claim element [9A] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.2, above.

3. [9B] memory encoding computer executable instructions that, when executed by the at least one processor, perform a method comprising:

171. Claim element [9B] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.3, above.

4. [9C] intercepting a request from a web browser;

172. Claim element [9C] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.4, above.

5. [9D] determining whether the request is associated with known malicious content;

173. Claim element [9D] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.5, above.

6. [9E] when the request is associated with known malicious content, blocking, by a protection agent, the request;

174. Claim element [9E] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.6, above.

7. [9F] when the request is not associated with known malicious content: sending the request to one or more web servers;

175. Claim element [9F] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.7, above.

- 8. [9G] receiving web content from one or more web servers, wherein web content comprises data for assembling a web page;**

176. Claim element [9G] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.8, above.

- 9. [9H] identifying a malware threat within the web content;**

177. Claim element [9H] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.9, above.

- 10. [9I] in response to identifying the malware threat, modifying the web content, wherein modifying the web content comprises modifying the web at protocol level to remove the malware threat from the web content;**

178. Claim element [9I] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.10, above.

- 11. [9J] assembling a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and**

179. Claim element [9J] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.11, above.

- 12. [9K] providing the modified version of the web page to a web browser application on the remote computer for rendering and display.**

180. Claim element [9K] is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.G.12 above.

H. Claim 10

181. Claim 10 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.H, above.

I. Claim 11

182. Claim 11 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.I, above.

J. Claim 12

183. Claim 12 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.J, above.

K. Claim 14

184. Claim 14 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.K, above.

L. Claim 16

185. Claim 16 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.L, above.

M. Claim 17

186. Claim 20 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.M, above.

N. Claim 18

187. Claim 18 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.N, above.

O. Claim 19

188. Claim 19 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.O, above.

P. Claim 20

189. Claim 20 is rendered obvious by Devarajan in view of Hubbard. *See* Section IX.P, above.

XI. GROUND 3: CLAIMS 1-20 ARE OBVIOUS OVER DEVARAJAN IN VIEW OF MCAFEE

A. Claim 1

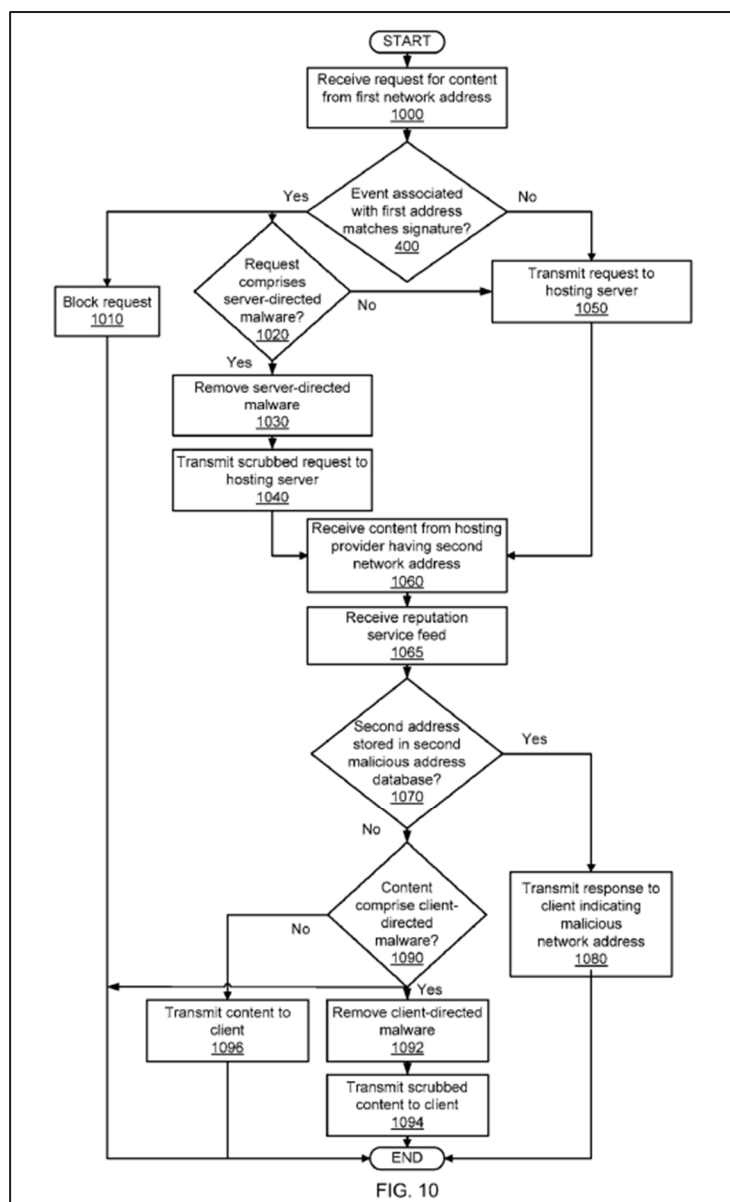
1. [1P] A content analysis and malware prevention method comprising:

190. To the extent the preamble is limiting, Devarajan teaches a content analysis and malware prevention method. I incorporate by reference my analysis for claim [1P] above with respect to Devarajan. *See* Section IX.A.1, above. As such, Devarajan in view of McAfee discloses or renders this limitation obvious.

2. [1A] intercepting, by a protection agent, a request from a web browser;

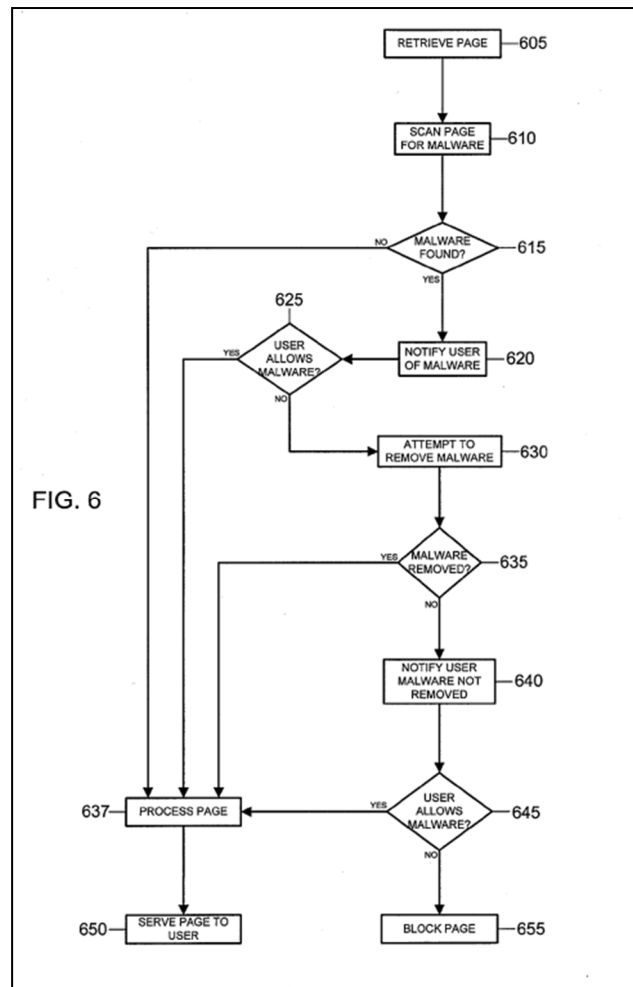
191. Devarajan discloses or renders obvious “intercepting, by a protection agent, a request from a web browser.” I incorporate by reference my analysis for claim [1A] above with respect to Devarajan. *See* Section IX.A.2, above.

192. Further, Devarajan, in view of McAfee, also teaches intercepting a request by a web browser by a protection agent. For example, Devarajan discloses that the “exemplary bi-directional network traffic malware detection and removal system” is “configured to receive a request for web content.” Ex.1004 (Devarajan) at [0006]. Furthermore, one embodiment of Devarajan depicted below in figure 10 shows that the system comprises “the steps of receiving, from a client 900 having a first network resource address 901, a request for content from a website 902 hosted on a hosting server.” Ex.1004 (Devarajan) at [0109]. This step is illustrated at block 1000 in the figure below:



Ex.1004 (Devarajan) at Fig. 10.

193. Like Devarajan, McAfee is also directed to a system for “providing safe web based interactions” and specifically, to “providing a safe environment for searching and browsing the web.” Ex.1005 (McAfee) at [0021]; *see also*, Ex.1005 (McAfee) at [0022], [0082]. McAfee’s Figure 6, reproduced below, shows the system’s process for analyzing content and preventing malware.



Ex.1005 (McAfee) at Fig. 6.

194. McAfee teaches that malware may infect a user's computer through a variety of means such as when a user visits a web page that "installs or otherwise executes the malware on the computer" or when a user "downloads or installs software from a malicious web page". Ex.1005 (McAfee) at [0002]. Devarajan detects malware based on known malware network addresses stored in its databases, and in the event malware is detected, Devarajan teaches removing the malware from the content request. Ex.1004 (Devarajan) at [0004], [0007]. Devarajan's malware

detection system can detect malware using “signature-based, statistical anomaly-based, and stateful protocol analysis methods.” Ex.1004 (Devarajan) at [0044]. A POSITA in the field of network security would understand the importance of malware detection prior to the execution of the malware on a user’s computer, and would have had a reasonable expectation of success in turning to references like McAfee, which teach a variety of malware detection methods, to further build on the teachings of Devarajan. A POSITA could easily adapt McAfee’s malware detection methods to work with the protection agent system taught by Devarajan, because the systems would work together.

195. To the extent Devarajan does not fully disclose the above limitation, McAfee discloses a system that comprises a “service provider” that “redirect[s] the...web traffic of the user” through its servers, thus acting as an “intermediary” in order to “scan the page for malware” and “process the page” before it is served to the user. Ex.1005 (McAfee) at [0035]. McAfee further teaches that the service provider server comprises a “safe page component” that is “operative to detect malware on [the requested page] and disable or remove the malware.” Ex.1005 (McAfee) at [0062]. This safe page component further comprises a “page processor” that “retrieves[s] the web page desired by the user.” Ex.1005 (McAfee) at [0064]. A POSITA would understand that by acting as an “intermediary” the service provider server intercepts a user’s requests in order to evaluate the results for

potential malware before it is served to the user in a manner akin to the claimed invention in the '928 patent.

196. As such, Devarajan in view of McAfee renders this limitation obvious.

3. [1B] determining, by the protection agent, whether the request is associated with known malicious content;

197. Devarajan discloses or renders obvious “determining, by the protection agent, whether the request is associated with known malicious content.” I incorporate by reference my analysis for claim 1[B] above with respect to Devarajan. *See* Section IX.A.3, above. As such, Devarajan in view of McAfee renders this limitation obvious.

4. [1C] when the request is associated with known malicious content, blocking, by the protection agent, the request;

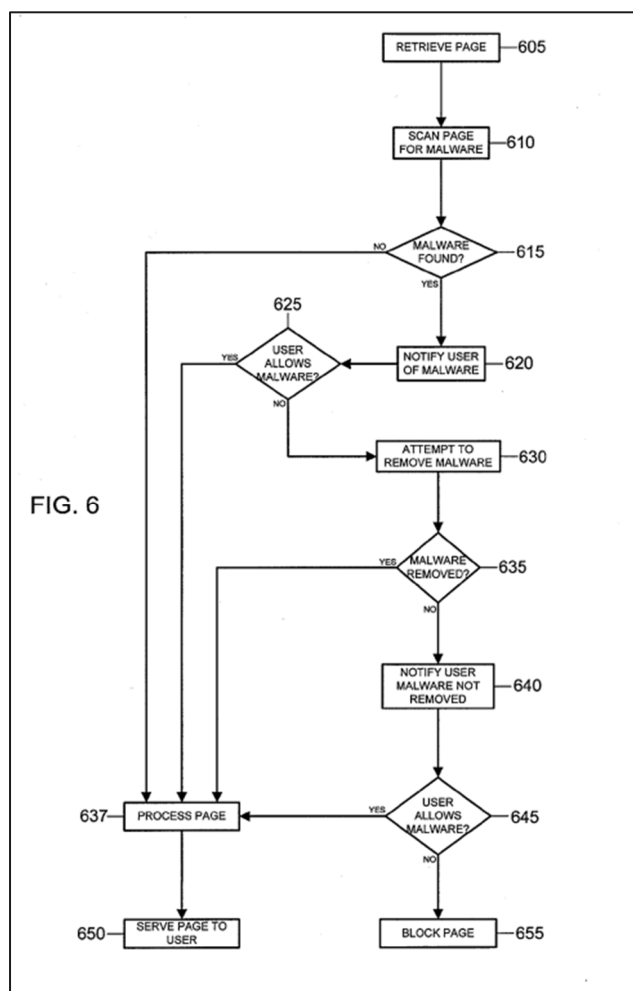
198. Devarajan discloses or renders obvious “when the request is associated with known malicious content, blocking, by the protection agent, the request.” I incorporate by reference my analysis for claim 1[C] above with respect to Devarajan. *See* Section IX.A.4, above. As such, Devarajan in view of McAfee renders this limitation obvious.

5. [1D] when the request is not associated with known malicious content: sending, by the protection agent, the request to one or more web servers

199. Devarajan, in view of McAfee, discloses or renders obvious “when the request is not associated with known malicious content: sending, by the protection

agent, the request to one or more web servers.” I incorporate by reference my analysis for claim 1[D] above with respect to Devarajan. *See* Section IX.A.5, above.

200. McAfee, as depicted in Figure 6 below, also teaches that if no malware is found at block 615, the request is moved down to block 637, where “the page may be further processed by the page processor.” Ex.1005 (McAfee) at [0082].



Ex.1005 (McAfee) at Fig. 6.

201. McAfee teaches that the page processor “may retrieve the page from the content provider server,” which constitutes sending a request to the server in

order to retrieve the page's content. Ex.1005 (McAfee) at [0077]; *see also*, [0005]; [0082].

202. Thus, it is my opinion that this limitation is disclosed or rendered obvious by Devarajan in view of McAfee.

6. **[1E] receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request;**

203. Devarajan discloses or renders obvious “receiving, at the protection agent on a remote computer, web content from the one or more web servers, wherein web content comprises data for assembling a web page, and wherein the web content is received in response to the request.” I incorporate by reference my analysis for claim 1[E] above with respect to Devarajan. *See* Section IX.A.6, above. As such, Devarajan in view of McAfee discloses or renders this limitation obvious

7. **[1F] identifying, by the protection agent, a malware threat within the web content;**

204. Devarajan discloses or renders obvious “identifying, by the protection agent, a malware threat within the web content.” I incorporate by reference my analysis for claim 1[F] above with respect to Devarajan. *See* Section IX.A.7, above. As such, Devarajan in view of McAfee discloses or renders this limitation obvious.

8. **[1G] in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web**

content comprises modifying the web content at protocol level to remove the malware threat from the web content;

205. Devarajan discloses or renders obvious “in response to identifying the malware threat, modifying, by the protection agent, the web content, wherein modifying the web content comprises modifying the web content at protocol level to remove the malware threat from the web content.” I incorporate by reference my analysis for claim 1[G] above with respect to Devarajan. *See* Section IX.A.8, above. As such, Devarajan in view of McAfee discloses or renders this limitation obvious.

- 9. [1H] assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and**

206. Devarajan discloses or renders obvious “assembling, by the protection agent, a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat.” I incorporate by reference my analysis for claim 1[H] above with respect to Devarajan. *See* Section IX.A.9, above. As such, Devarajan in view of McAfee discloses or renders this limitation obvious.

- 10. [1I] providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display.**

207. Devarajan discloses or renders obvious “providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display”. I incorporate by reference my analysis

for claim 1[I] above with respect to Devarajan. *See* Section IX.A.10, above. As such, Devarajan in view of McAfee discloses or renders this limitation obvious.

208. Devarajan in view of McAfee discloses providing, by the protection agent, the modified version of the web page to the web browser application on the remote computer for rendering and display. As discussed above, Devarajan discloses modifying the webpage using the content sanitizer module and “after the malware has been removed” transmitting the “‘scrubbed’ content request...to the hosting server.” Ex.1004 (Devarajan) at [0101].

209. Therefore, Devarajan in view of McAfee renders claim 1 obvious.

B. Claim 2

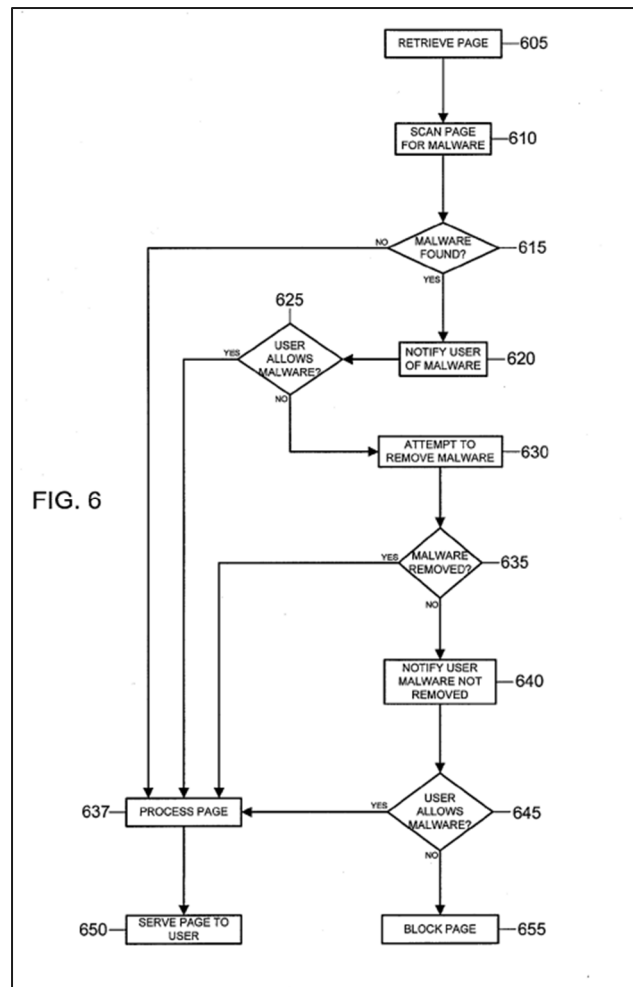
1. The method of claim 1, further comprising supplementing, by the protection agent, the web content with indicators regarding a vulnerability of one or more links within the web content.

210. Devarajan teaches “[t]he method of claim 1, further comprising supplementing, by the protection agent, the web content with indicators regarding a vulnerability of one or more links within the web content”. I incorporate by reference my analysis for claim 2 above with respect to Devarajan. *See* Section IX.B, above.

211. Although Devarajan teaches supplementing, by the protection agent, the web content with indicators regarding a vulnerability of one or more links within the web content, Devarajan does not expressly disclose providing such indicators in the web page itself. If it is determined that this is required for this claim, then, given

Devarajan's teaching that such indicators can include colors, in my opinion, it would be obvious to a POSITA to include such indicators within the web page itself.

212. Further, Devarajan in view of McAfee renders obvious supplementing, by the protection agent, the web content with indicators regarding a vulnerability of one or more links within the web content, where such indicators are included within the web page itself. Going back to Figure 6 of McAfee, reproduced below, at block 615 if the page processor finds malware on the page, the system moves to the next block, 620. Ex.1005 (McAfee) at [0082]. At block 620, "the service provider server 240 may notify the user A 120A that the page contains malware." Ex.1005 (McAfee) at [0082]. The service provider may further indicate to the user "the type of malware, and generally any information that may describe the malware." Ex.1005 (McAfee) at [0082]. Further, McAfee teaches, "[w]hen the user performs a search on a supported search engine the software may, for example, consult a 'hot list' of web sites known to promulgate malware to advise the user as to whether the web pages referenced in the search results may contain malware. The user may then make a determination as whether to browse the web pages in the search results or not." Ex.1005 (McAfee) at [0003].



Ex.1005 (McAfee) at Fig. 6.

213. McAfee further teaches such indicators can be included in the web page returned to a user:

If the service provider 130 detects malware on a page the service provider 130 may notify the user A 120A of the malware, and provide the user A 120A with the option of removing the malware from the page or otherwise neutralizing it or browsing the page with the malware intact. Malware “on a page” may refer to malware being included in the underlying code which defines the visual

representation of the page, or in other code which may be executed by the loading, interpretation or execution of the page code or which may be triggered by the mere access, e.g. sending of a get request, to the URL of the particular page.

Ex.1005 (McAfee) at [0036].

214. Thus, Devarajan in view of McAfee renders obvious claim 2.

C. Claim 3

- 1. The method of claim 1, wherein an analysis to identify a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis.**

215. Devarajan teaches “[t]he method of claim 1, wherein an analysis to identify a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis”. I incorporate by reference my analysis for claim 3 above with respect to Devarajan. *See* Section IX.C, above.

216. Further, Devarajan in view of McAfee renders obvious the method of claim 1, wherein an analysis to identify a malware threat within the web content includes one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis. McAfee’s system comprises a “URL processor” that is “operative to encode and decode the URLs on a page.” Ex.1005 (McAfee) at [0063]. These URLs may be “encoded by

the URL processor 314 to ensure each of the URLs redirects the user A 120A through the service provider server 240.” Ex.1005 (McAfee) at [0063]. McAfee further discloses a “safe search” mode that provides the user with safe web based interactions. Ex.1005 (McAfee) at [0087]. This system teaches “exemplary HTML code for displaying safe search results”, where the HTML code “may utilize invisible frame, or an HTML iframe element, to implement the safe browsing system.” Ex.1005 (McAfee) at [0087]. A POSITA would understand that this disclosure indicates, at the very least, that the system performs HTML and URL analysis.

217. As such, Devarajan in view of McAfee discloses or renders this claim obvious.

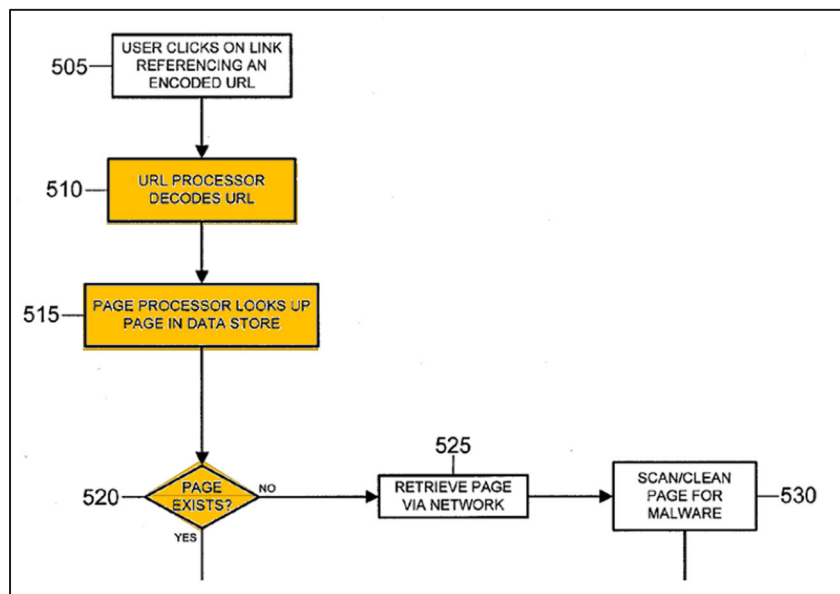
D. Claim 4

- 1. The method of claim 1, further comprising determining, by the protection agent, whether a request for the web content from the web browser application is a first request.**

218. Devarajan teaches “[t]he method of claim 1, further comprising determining, by the protection agent, whether a request for the web content from the web browser application is a first request”. I incorporate by reference my analysis for claim 4 above with respect to Devarajan. *See* Section IX.D, above.

219. Further, Devarajan in view of McAfee renders obvious determining, by the protection agent, whether a request for the web content from the web browser

application is a first request. Figure 5 of McAfee, reproduced below, discloses the workings of the URL processor in the claimed system. The process begins when a user “click[s] on a link representing a page of the content provider.” Ex.1005 (McAfee) at [0073]. At the next block, 510, the “service provider server 240 may receive the request and may communicate [it] to the URL processor,” which decodes the URL. Ex.1005 (McAfee) at [0073]. At the next block, the page processor looks “up the page referenced by the actual URL in the page data store 320” in order to determine whether the request had been made before and if “a clean copy of the page is cached in the page data store.” Ex.1005 (McAfee) at [0074].



Ex.1005 (McAfee) at Fig. 5 (annotated).

220. A POSITA would understand the process of trying to determine whether a request was previously made and processed by the system as determining whether the request for web content is a first request. Moreover, as I explain above

with respect to the motivation to combine Devarajan and McAfee, a POSITA would be motivated to add the teachings from McAfee regarding caching and a first request to the system of Devarajan, as it would improve the response time for the system, since it would replace a full analysis and scrubbing with a simple memory lookup. In addition, this would be a simple modification, since caching is a well-known technique; hence, a POSITA would have an expectation of success and would be motivated to try this change. As such, Devarajan in view of McAfee discloses or renders this claim obvious.

E. Claim 5

1. The method of claim 1, further comprising performing one or more of pre-process data decryption, de-chunking and decompressing.

221. Devarajan in view of McAfee renders obvious “[t]he method of claim 1, further comprising performing one or more of pre-process data decryption, de-chunking and decompressing.”

222. A POSITA would understand that web content that is encrypted, chunked, and/or compressed poses a challenge with respect to the detection of malware in the content, since such content is not immediately viewable. For example, a POSITA would understand that when a long URL, such as https://en.wikipedia.org/wiki/URL_shortening, is replaced by a shortened URL, such as <https://w.wiki/U>, the compressed, shortened URL may be hiding a URL that is associated with malware. Thus, a POSITA would be motivated to look to a

reference such as McAfee for guidance regarding dealing with compressed URLs, e.g., to find guidance on decompressing such URLs so as to apply the Devarajan URL analysis method to the decompressed web content.

223. McAfee teaches that “[t]he system may allow users to designate shortened URLs which may reference a longer URL. The service provider may then serve the page referenced by the longer URL when a user browses to the shortened URL. Furthermore, the system may allow a user to reference a highlighted page or notated page through a shortened URL” Ex.1005 (McAfee) at [0028]. McAfee also teaches, “[t]he URL processor 314 may be operative to encode and decode the URLs on a page. The URLs on a page may be encoded by the URL processor 314 to ensure each of the URLs redirects the user A 120A through the service provider server 240. The URL processor 314 may store the mapping between the original URL and the encoded URL in the URL mapping store 330. The URL processor 314 may be operative to decode an encoded URL by looking up the original URL in the URL mapping data store 330. The URL mapping data store 330 may be a data structure, such as a database, a hash table, or generally any data structure capable of mapping an encoded URL to a URL.” Ex.1005 (McAfee) at [0063]. McAfee further teaches that when a user clicks on a link for a webpage, McAfee’s system communicates the “encoded URL to the URL processor,” which then “decode[s] the URL by looking up the actual URL mapped to the encoded URL in the URL mapping data store 330.”

Ex.1005 (McAfee) at [0073]. A POSITA would understand this decoding as decompressing the web content prior to the request being processed by the system. Further, once this decompression is done, a POSITA would understand that the analysis of Devarajan can proceed as disclosed in Devarajan without further modifications being required. Thus, a POSITA would have an expectation of success in adding the URL decoding processor of McAfee to the Devarajan system, and such a modification would add URL decompressing to the Devarajan system along with the benefits of additional malware detection that would be provided. Thus, in my opinion, Devarajan in view of McAfee discloses or renders this claim obvious.

F. Claim 6

1. The method of claim 1, wherein a protocol-level analysis is utilized to identify the malware threat within the web content.

224. Devarajan teaches “[t]he method of claim 1, wherein a protocol-level analysis is utilized to identify the malware threat within the web content”. I incorporate by reference my analysis for claim 6 above with respect to Devarajan. *See* Section IX.E, above.

225. Further, Devarajan in view of McAfee renders obvious the method of claim 1, wherein a protocol-level analysis is utilized to identify the malware threat within the web content. McAfee’s system comprises a “URL processor” that is “operative to encode and decode the URLs on a page.” Ex.1005 (McAfee) at [0063]. These URLs may be “encoded by the URL processor 314 to ensure each of the URLs

redirects the user A 120A through the service provider server 240.” Ex.1005 (McAfee) at [0063]. McAfee further discloses a “safe search” mode that provides the user with safe web based interactions. Ex.1005 (McAfee) at [0087]. This system teaches “exemplary HTML code for displaying safe search results”, where the HTML code “may utilize invisible frame, or an HTML iframe element, to implement the safe browsing system.” Ex.1005 (McAfee) at [0087]. A POSITA would understand that these disclosures indicate, at the very least, that the system performs HTML and URL analysis, with are protocol-level operations.

226. As such, Devarajan in view of McAfee discloses or renders this claim obvious.

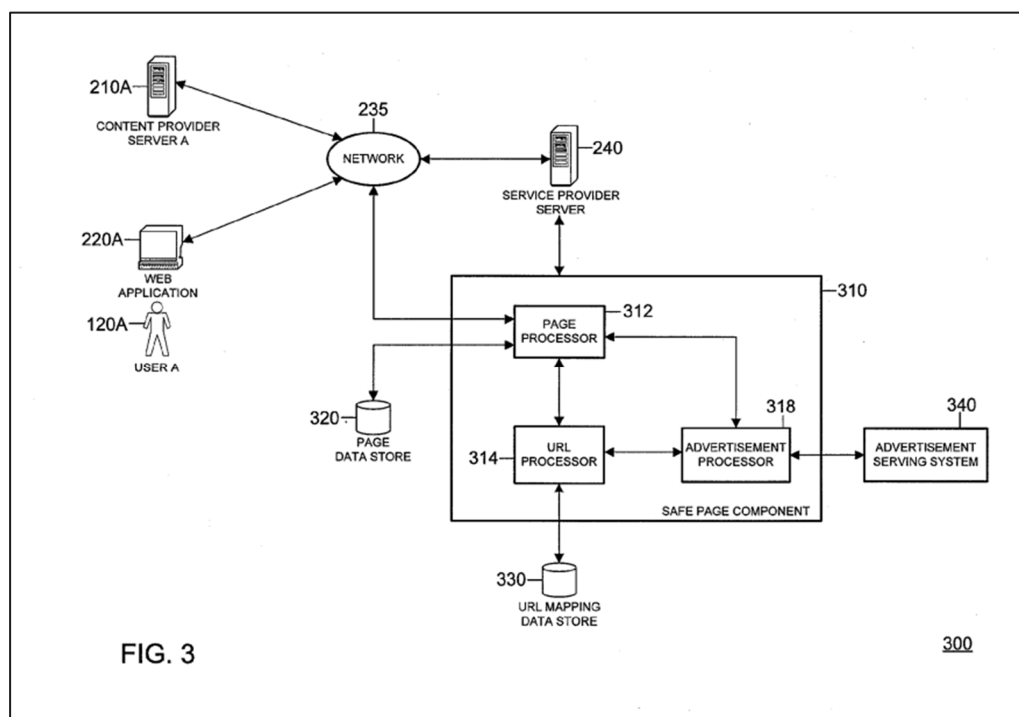
G. Claim 7

- 1. The method of claim 3, further comprising performing cloud verification of the web content, wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses.**

227. Devarajan teaches “[t]he method of claim 3, further comprising performing cloud verification of the web content, wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses”. I incorporate by reference my analysis for claim 7 above with respect to Devarajan. *See* Section IX.F, above.

228. Further, Devarajan in view of McAfee renders obvious the method of claim 3, further comprising performing cloud verification of the web content,

wherein cloud verification determines one or more malware vulnerabilities based on an aggregation of the analyses. As discussed above, McAfee describes a system with a service provide 240, as depicted below in figure 3:



Ex.1005 (McAfee) at Fig. 3

229. Further, McAfee teaches, “[t]he service provider server 240 may exist on one machine or may be running in a distributed configuration on one or more machines.” Ex.1005 (McAfee) at [0055]. A POSITA would understand that implementing the service provider in a distributed computing, i.e., “cloud” environment, would provide additional benefits, such as being more tolerant to computer or network failures (since the computations are distributed across multiple computers). Given McAfee’s teaching, a POSITA would be motivated to implement

the system of Devarajan in a distributed (i.e., cloud computing) environment, where, e.g., some of the components of the server 108 (such as the malicious address database) are distributed and mirrored across multiple computers in the cloud. Further, this teaching is consistent with the teaching in Devarajan that “[t]he instructions 110 may, as non-limiting examples, comprise software and/or scripts stored in the computer-readable media 109 that may be stored locally in the server 108 or, alternatively, in a highly-distributed format in a plurality of computer-readable media 109 accessible via the network 101, perhaps via a grid or cloud-computing environment.” Ex.1004 (Devarajan) at [0054]. Further, given that the system in Devarajan is already performing an aggregation of analyses on web content, a POSITA would understand this modification to perform the method of Devarajan in cloud to be a simple modification that would have a likelihood of success. For example, Devarajan already discloses that its analysis of web content includes an aggregation for identifying “client-directed malware (e.g., a virus, a worm, a trojan horse, a rootkit, a backdoor, a spyware, a keystroke logger, a phishing application, a script injection, or an iFrame injection).” Ex.1004 (Devarajan) at [0102]. A POSITA would understand that distributing this aggregated analysis to the cloud would improve robustness and availability of the system; hence, a POSITA would be motivated to make this modification of Devarajan in view of McAfee and there would be an expectation of success.

230. As such, Devarajan in view of McAfee discloses or renders this claim obvious.

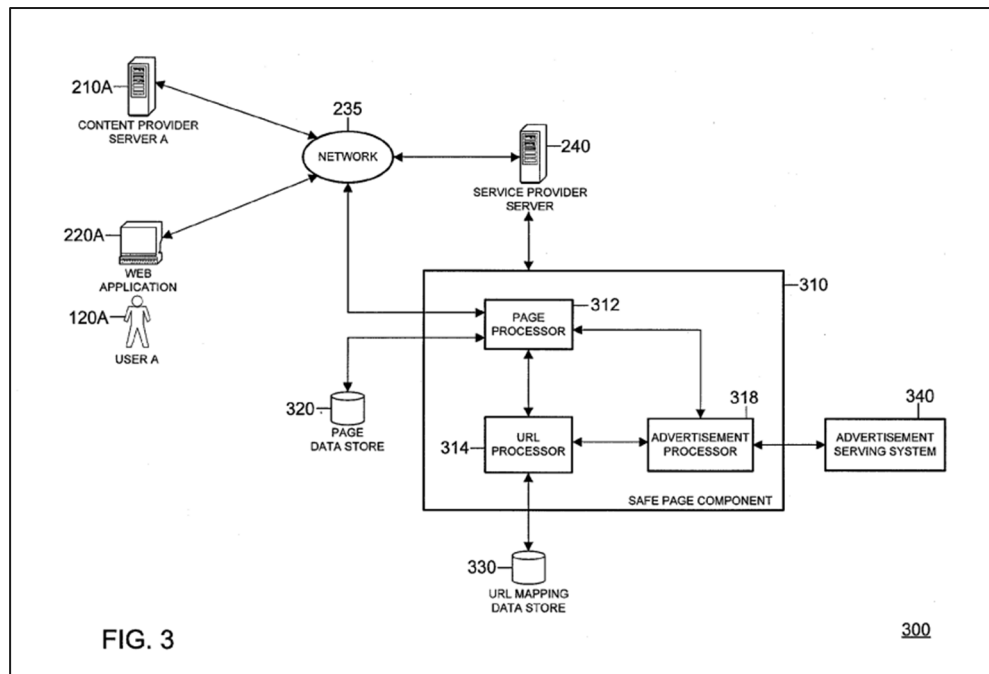
H. Claim 8

1. The method of claim 1, further comprising allocating memory for caching the web content.

231. Devarajan in view of McAfee renders obvious “[t]he method of claim 1, further comprising allocating memory for caching the web content.”

232. A POSITA would be aware of the benefits of caching (e.g., in reducing latency) and would be motivated to look to a reference such as McAfee for guidance.

233. McAfee’s Figure 3, reproduced below, illustrates the server-side components of its system. The system comprises a “a page data store 320, which may cache clean versions of web pages.” Ex.1005 (McAfee) at [0061]. A POSITA would immediately recognize the benefits of such caching, as it allows the system to reuse previously scrubbed web content rather than redoing all the work to scrub a web page that has previously been analyzed and cleaned. For example, McAfee teaches, “[t]he pages stored, or cached, in the page data stores 320 may be immediately served to the user A 120A, and may be utilized to improve the overall performance of the system 100.” Ex.1005 (McAfee) at [0079].



Ex.1005 (McAfee) at Fig. 3.

234. Similarly, as discussed above, when a request is made for web content through McAfee's system, the page processor determined "whether the page is cached in the page data store 320." Ex.1005 (McAfee) at [0064]. A POSITA would therefore be motivated to incorporate this simple teaching regarding caching into the system of Devarajan, and there would be an expectation of success, given the prevalence of caching in computing generally. As such, Devarajan in view of McAfee discloses or renders this claim obvious.

I. Claim 9

1. [9P] A content analysis and malware prevention system comprising:

235. This claim is rendered obvious by Devarajan in view of McAfee. *See* discussion above for claim [1P]. *See* Section XI.A.1, above.

2. [9A] at least one processor; and

236. Devarajan in view of McAfee discloses a processor. I incorporate by reference my analysis for claim [9A] above with respect to Devarajan. *See* Section IX.G.2, above. Furthermore, McAfee also discloses that its application “may be implemented with a processor, such as a personal computer, personal digital assistant, mobile phone, or any other machine capable of implementing a web application.” Ex.1005 (McAfee) at [0034]. Thus, in my opinion, this claim is rendered obvious by Devarajan in view of McAfee.

3. [9B] memory encoding computer executable instructions that, when executed by the at least one processor, perform a method comprising:

237. Devarajan in view of McAfee discloses or renders obvious a “memory encoding computer executable instructions that, when executed by the at least one processor, perform a method.” *See* Section IX.G.3, above. I incorporate by reference my analysis for claim [9B] above with respect to Devarajan. Further, McAfee discloses that the functions of its systems “may be performed by the programmed processor 1302 executing the instructions 1324 stored in the memory 1304.”

Ex.1005 (McAfee) at [0095]. Thus, in my opinion, this claim is rendered obvious by Devarajan in view of McAfee.

4. [9C] intercepting a request from a web browser;

238. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.2, above.

5. [9D] determining whether the request is associated with known malicious content;

239. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.3, above.

6. [9E] when the request is associated with known malicious content, blocking, by a protection agent, the request;

240. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.4, above.

7. [9F] when the request is not associated with known malicious content: sending the request to one or more web servers;

241. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.5, above.

8. [9G] receiving web content from one or more web servers, wherein web content comprises data for assembling a web page;

242. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.6, above.

9. [9H] identifying a malware threat within the web content;

243. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.7, above.

10. [9I] in response to identifying the malware threat, modifying the web content, wherein modifying the web content comprises modifying the web at protocol level to remove the malware threat from the web content;

244. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.8, above.

11. [9J] assembling a modified version of the web page, wherein the modified web page is assembled using the modified web content, and wherein the modified version of the web page does not comprise the malware threat; and

245. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.9, above.

12. [9K] providing the modified version of the web page to a web browser application on the remote computer for rendering and display.

246. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.A.10, above.

J. Claim 10

1. The system of claim 9, wherein the method further comprises supplementing the web content with indicators regarding a vulnerability of one or more links within the web content.

247. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.B, above.

K. Claim 11

- 1. The system of claim 9, wherein the method further comprises identifying the malware threat within the web content by performing one or more of a URL (Uniform Resource Locator) analysis, an IP (Internet Protocol) analysis, an image analysis, and a script/HTML analysis.**

248. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.C, above.

L. Claim 12

- 1. The system of claim 9, wherein the method further comprises determining whether a request for the web content from the web browser application is a first request.**

249. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.D, above.

M. Claim 13

- 1. The system of claim 9, wherein the method further comprises performing one or more of pre-process data decryption, de-chunking, and decompressing.**

250. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.E, above.

N. Claim 14

- 1. The system of claim 9, wherein the method further comprises identifying the malware threat utilizing a protocol-level analysis.**

251. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.F, above.

O. Claim 15

- 1. The system of claim 9, wherein the method further comprises caching the web content in allocated memory.**

252. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.H, above.

P. Claim 16

- 1. The method of claim 1, wherein the malware threat comprises a Uniform Resource Locator (URL).**

253. Devarajan in view of McAfee teaches the “method of claim 1, wherein the malware threat comprises a Uniform Resource Locator (URL).” I include by reference my analysis of claim 16 with respect to Devarajan. As discussed above, McAfee’s system comprises a “URL processor” that is “operative to encode and decode the URLs on a page.” Ex.1005 (McAfee) at [0063]. These URLs may be “encoded by the URL processor 314 to ensure each of the URLs redirects the user A 120A through the service provider server 240,” so as to analyze whether the URL comprises malware. Ex.1005 (McAfee) at [0063]. Devarajan similarly teaches an intrusion detection and prevention module, which analyzes the content on a page if it “identifies a link to a known malware website in the content, the content sanitizer module 909 may either block the content or remove the link from the content.” Ex.1004 (Devarajan) at [0104]. A POSITA would understand a “link” to be a URL. As such, Devarajan in view of McAfee discloses or renders this claim obvious.

Q. Claim 17

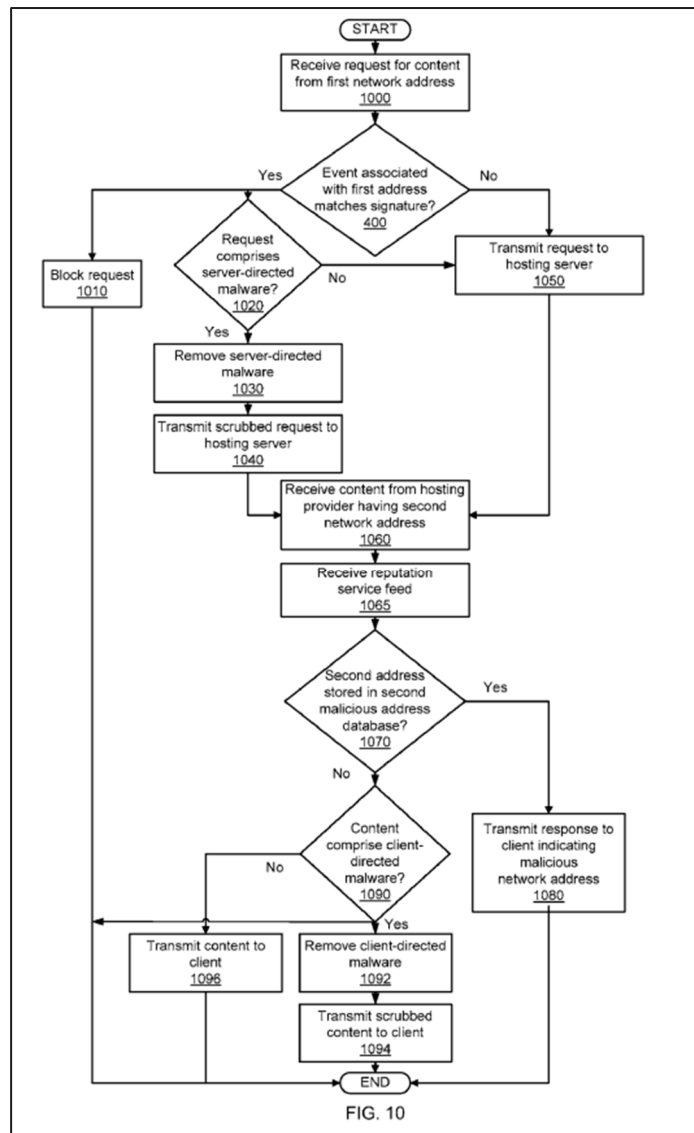
- 1. The system of claim 9, wherein the method further comprises determining if a malware threat comprises a Uniform Resource Locator (URL).**

254. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.P, above.

R. Claim 18

- 1. The method of claim 1, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.**

255. Devarajan in view of McAfee discloses the “method of claim 1, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.” I include by reference my analysis for claim 18 with respect to Devarajan. For example, Devarajan teaches sending requests to one or more web servers in the event malware is not detected. One embodiment of Devarajan depicted below in figure 10 illustrates a “method for bi-directional detection and removal of network traffic malware”. Ex.1004 (Devarajan) at [0109].

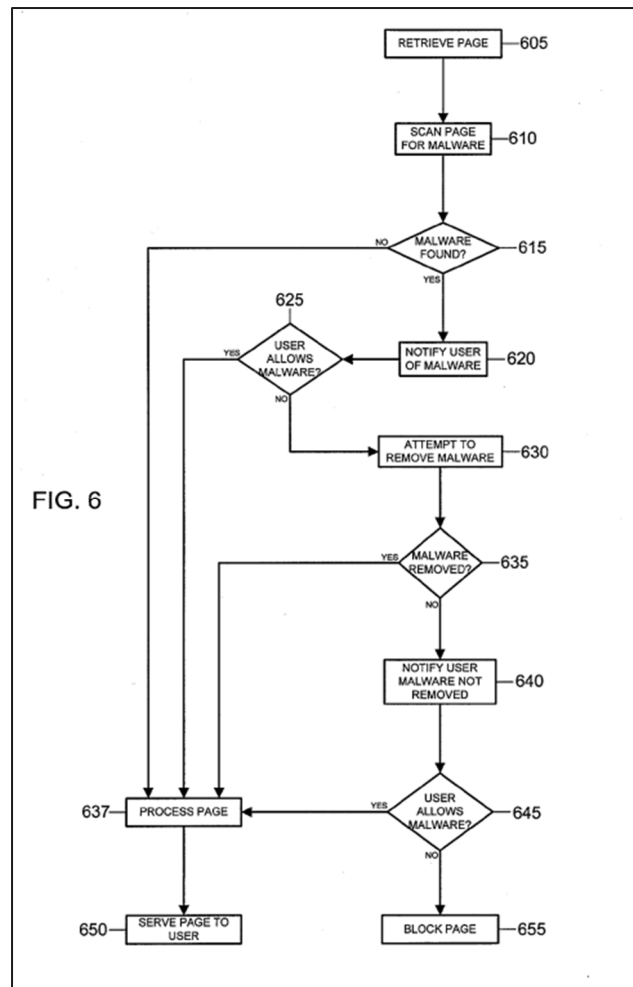


Ex.1004 (Devarajan) at Fig. 10.

256. As shown in figure 10, the process begins when the system “receive[s]...a request for content” as shown in block 1000. Ex.1004 (Devarajan) at [0109]. At the next block (400), the system determines “whether an event associated with the client 900 or its first network resource address 901 matches one or more of a plurality of event signatures 102.” Ex.1004 (Devarajan) at [0111].

These “signatures 102” are “simply known attack patterns” and “comprise a plurality of malware signatures including, but not limited to a virus signature, a worm signature, a trojan horse signature, a rootkit signature, a backdoor signature, a spyware signature, a keystroke logger signature, or a phishing application signature.” Ex.1004 (Devarajan) at [0046]. If at step 400 the system determines that “the event does not match an event signature 102, the request for content may be transmitted ...to the hosting server 903” as depicted in step 1050. Ex.1004 (Devarajan) at [0111]. A POSITA would understand this step to mean the request for content is sent iteratively until the content is received by the server.

257. To the extent Devarajan does not disclose this limitation, McAfee, as depicted in Figure 6, below teaches that if no malware is found at block 615, the request is moved down to block 637, where “the page may be further processed by the page processor.” Ex.1005 (McAfee) at [0082].



Ex.1005 (McAfee) at Fig. 6.

258. McAfee teaches that the page processor “may retrieve the page from the content provider server,” which constitutes sending a request to the server in order to retrieve the page’s content. Ex.1005 (McAfee) at [0077]; *see also*, [0005]; [0082]. A POSITA would understand that such requests are sent iteratively until all of the web content is received.

259. Thus, Devarajan in view of McAfee discloses or renders this claim obvious.

S. Claim 19

- 1. The method of claim 1, further comprising processing the web content prior to identifying the malware threat within the web content.**

260. Devarajan in view of McAfee discloses the “method of claim 1, further comprising processing the web content prior to identifying the malware threat within the web content.” I include by reference my analysis of claim 19 with respect to Devarajan. Further, McAfee discloses decoding URLs on a page prior to analysis of the content linked to the URLs for malware. Ex.1005 (McAfee) at [0063]. Similarly, Devarajan discloses processing the web content by analyzing its IP address prior to determining whether or not it comprises malware. Ex.1004 (Devarajan) at [0117]. Thus, this claim is disclosed or rendered obvious by Devarajan in view of McAfee.

T. Claim 20

- 1. The system of claim 9, wherein sending the request to one or more web servers comprises iteratively sending a plurality of requests to the one or more web servers.**

261. This claim is rendered obvious by Devarajan in view of McAfee. *See* Section XI.R, above.

XII. SECONDARY CONSIDERATIONS

262. I have not seen any evidence of secondary considerations that would support a determination of non-obviousness of the claimed subject matter in the '928

patent. To the extent the Patent Owners or its expert provides of such secondary considerations, I reserve the right to respond.

XIII. CONCLUSION

263. It is my opinion that all claims of the '928 patent are invalid.

264. I hereby declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I understand that willful false statements are punishable by fine or imprisonment or both. *See* 18 U.S.C. § 1001.

Date: March 30, 2023

Respectfully submitted,



Dr. Michael T. Goodrich