UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

PALO ALTO NETWORKS, INC.
Petitioner,

v.

CENTRIPETAL NETWORKS, INC.
Patent Owner.

IPR2022-00182

Patent No. 9,917,856

_____

DECLARATION OF DR. JON WEISSMAN

TABLE OF CONTENTS

I. INTRODUCTION 1

II. QUALIFICATIONS AND EXPERIENCE 2

III. MATERIALS CONSIDERED AND RELIED UPON 6

IV. LEGAL STANDARDS FOR PATENTABILITY 6

A. Obviousness 7

B. Claim Construction 12

V. DESCRIPTION OF THE RELEVANT FIELD AND THE RELEVANT TIMEFRAME 13

VI. THE PERSON OF ORDINARY SKILL IN THE RELEVANT FIELD IN THE RELEVANT TIMEFRAME 14

VII. TECHNICAL BACKGROUND AND STATE OF THE ART 16

A. Computer Network Architectures and Communication Protocols 17

B. Network Security Systems – Firewalls 19

C. Network Security Systems – Intrusion Detection Systems and Intrusion Prevention Systems 24

D. Network Security Systems – Detailed Concepts 30

1. Applying Rules to Filter Packets in a Packet Filtering System 31

2. Identifying Security Risks Based on Network-Threat Indicators 32

3. Correlating Packets of a Session 35

4. Inspecting Handshake Messages Prior To Establishing Encrypted Sessions. 36

5. Routing Packets Between Functions of A Security Device 38

6. Sending Suspicious Packets to a Proxy System 39

VIII. OVERVIEW OF THE ’856 PATENT 46

A. Patent Overview 46

B. Prosecution History 50

IX. CLAIM INTERPRETATION 51

a. “network-threat indicators” 51

X. GROUND 1: CLAIMS 1, 24-25 ARE RENDERED OBVIOUS BY BURUGANAHALLI 52

A. Overview of Buruganahalli 53

B. Challenged Claims 64

C. Claim Chart 97

XI. GROUND 2: CLAIMS 1, 24-25 ARE RENDERED OBVIOUS BY BURUGANAHALLI IN VIEW OF BAEHR 97

XII. SECONDARY CONSIDERATIONS 107

XIII. CONCLUSION 108

I, Dr. Jon Weissman, hereby declare under penalty of perjury:

I. INTRODUCTION

1. My name is Dr. Jon Weissman, and I have been retained by counsel for Petitioner Palo Alto Networks Inc. ("PAN" or "Petitioner") as an expert witness to provide assistance regarding U.S. Patent No. 9,917,856 (Ex. 1001, "the '856 Patent"). Specifically, I have been asked to consider the validity of claims 1, 24-25 of the '856 Patent (the "Challenged Claims") in view of prior art, obviousness considerations, and understanding of a person of ordinary skill in the art at the time of the invention as it relates to the '856 Patent. I have personal knowledge of the facts and opinions set forth in this declaration and believe them to be true. If called upon to do so, I would testify competently thereto.

2. For my time working on this matter, I am being compensated at my standard hourly consulting rate. I do not have any personal or financial stake or interest in the outcome of the present proceeding, and the compensation is not dependent on the outcome of this IPR and in no way affects the substance of my statements in this Declaration. I do not have any expectation or promise of additional business with the Petitioner in exchange for the positions explained herein. My analysis here is based on my years of education, research and experience, as well as my investigation and study of relevant materials, including those cited herein.

3. I may rely upon these materials, my knowledge and experience, and/or additional materials to rebut arguments raised by the Patent Owner. Further, I may also consider additional documents and information in forming any necessary opinions, including documents that may not yet have been provided to me.

4. My analysis of the materials in this proceeding is ongoing and I will continue to review any new material as it is provided. This declaration represents only those opinions I have formed to date. I reserve the right to revise, supplement, and/or amend my expert opinions stated herein based on new information and on my continuing analysis of the materials already provided.

II. QUALIFICATIONS AND EXPERIENCE

5. My qualifications can be found in my Curriculum Vitae, which includes a complete list of my publications, and is attached as **Appendix A**. Some of my background and experience that qualifies me to offer the opinions offered in this Declaration set forth as an expert in the technical issues in this case are as follows.

6. I am a Full Professor of Computer Science at the University of Minnesota, where I lead the Distributed Computing Systems Group. I received my B.S. in Applied Mathematics and Computer Science from Carnegie-Mellon, and my M.S. and Ph.D. in Computer Science from the University of Virginia.

7. For my Ph.D. thesis, I developed the first automated scheduling system for parallel and distributed applications across heterogeneous local and wide-area networks. I thereafter worked as a software engineer for five years in the area of distributed systems. In 1995, I returned to academia and began my career as a professor in networked distributed systems. My work has included packet analysis for network intrusion, proxy routing in edge networks, and monitoring and analysis of network performance. I have also taught the following topics in my networking, operating systems, distributed systems, and internet programming courses: packet-switched networking, firewalls, security policies, packets filtering, and the use of network-threat indicators to address cyber-attacks.

8. My research has been funded by NASA, the National Science Foundation, the Department of Energy, and the Air Force, and has included the following projects related to the network traffic monitoring and processing, and real-time systems:

- National Science Foundation, "Scaling the IoT with Constellation";
- National Science Foundation, "Location, location, location (L3): Support for Geo-Centric Applications";
- National Science Foundation, "One Thousand Points of Light: Accelerating Data-Intensive Applications By Proxy";
- National Science Foundation, "A Data Mining and Exploration

Middleware for Grid and Distributed Computing";

- National Science Foundation, "Toward Community Services: Putting Parallel Network Services On-line";
- National Science Foundation, "Resource Management for Parallel and Distributed Systems"; and
- Air Force Office of Scientific Research, "Telecommunication Networks for Mobile and Distributed Computing and Communications."

9. I have published over 100 peer-reviewed technical articles, including some awarded or nominated for Best Paper at highly competitive international conferences. Many of my published papers relate to the subject matter of the Challenged Claims, including this small sample (and many more listed on my CV):

- "Rethinking Adaptability in Wide-Area Stream Processing Systems," Albert Jonathan, Abhishek Chandra, and Jon Weissman, 10th USENIX Workshop on Hot Topics in Cloud Computing;
- "Nebula: Distributed Edge Cloud for Data Intensive Computing," Albert Jonathan, Mathew Ryden, Kwangsung Oh, Abhishek Chandra, and Jon Weissman, *IEEE Transactions on Parallel and Distributed Systems*;
- "TripS: Automated Multi-tiered Data Placement in a Geo-distributed Cloud Environment," Kwangsung Oh, Abhishek Chandra, and Jon Weissman, *10th ACM International Systems and Storage Conference*;
- "Redefining Data Locality for Cross-Data Center Storage," Kwangsung Oh, Ajaykrishna Raghavan, Abhishek Chandra, and Jon Weissman, *The 2nd International Workshop on Software-Defined*

Ecosystems;

- "Passive Network Performance Estimation for Large-scale, Data-Intensive Computing," Jinoh Kim, Abhishek Chandra, and Jon B. Weissman, *IEEE Transactions on Parallel and Distributed Systems*;
- "DDDAS/ITR: A Data Mining and Exploration Middleware for Grid and Distributed Computing," Jon B. Weissman, Vipin Kumar, Varun Chandola, Eric Eilertson, Levent Ertoz, Gyorgy Simon, Seonho Kim, and Jinoh Kim, *Workshop on Dynamic Data Driven Application Systems – DDDAS*;
- "Scheduling Parallel Applications in Distributed Networks," Jon B. Weissman and Xin Zhao, *Journal of Cluster Computing*;
- "Adaptive Resource Scheduling for Network Services," Byoung-Dai Lee and Jon B. Weissman, *IEEE 3rd International Workshop on Grid Computing*;
- "Eliminating the Middle-Man: Peer-to-Peer Dataflow," Adam Barker, Jon B. Weissman, and Jano van Hemert, *17th IEEE International Symposium on High Performance Distributed Computing*; and
- "Optimizing Remote File Access for Parallel and Distributed Network Applications," Jon B. Weissman, Mike Gingras, and Mahesh Marina, *Journal of Parallel and Distributed Computing*.

10. Additionally, I have served on the boards of several flagship journals, including IEEE Transactions on Parallel and Distributed Systems and IEEE Transactions on Computers. I have previously served as an investigator for both the Center for Research in Intelligent Storage (sponsored by the National Science

Foundation) and the Digital Technology Center. I am the steering committee chair of the ACM International Symposium on High Performance Parallel and Distributed Systems, the flagship conference in my area.

III. MATERIALS CONSIDERED AND RELIED UPON

11. I have reviewed the ’856 patent, its prosecution history, the prior art and other documents and materials cited herein. For ease of reference, the full list of documents that I have considered is included in **Appendix B**. Each of these materials is a type of document that experts in my field would have reasonably relied upon when forming their opinions and would have had access to either through the applicable patent office and/or well-known libraries, conferences, publications, organizations and websites in the field as further discussed herein.

12. My opinions in this declaration are based on my review of these documents, as well as upon my education, training, research, knowledge, and experience.

IV. LEGAL STANDARDS FOR PATENTABILITY

13. I am not a lawyer and do not provide any legal opinions, but I have been advised that certain legal standards are to be applied by technical experts in forming opinions regarding meaning and validity of patent claims. I have applied the legal standards described below, which were provided to me by counsel for Petitioner.

14. I have been informed that prior art may render a patent claim unpatentable if the prior art can be shown to have made the claim "obvious" to a person of ordinary skill in the art.

A. Obviousness

15. I have been informed that a claim may be unpatentable under (AIA) 35 U.S.C. § 103(a) if the subject matter described by the claim as a whole would have been obvious to a hypothetical person of ordinary skill in the art in view of a prior art reference or in view of a combination of references at the time the claimed invention was made. I have been informed that obviousness is determined from the perspective of a hypothetical person of ordinary skill in the art and that the asserted claims of the patent should be read from the point of view of such a person at the time the claimed invention was made. I have been informed that a hypothetical person of ordinary skill in the art is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent in suit.

16. I have been informed that there are two criteria for determining whether prior art is analogous and thus can be considered prior art: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the art is not within the field of the patentee's endeavor, whether the teachings of the art are still reasonably pertinent to the particular problem with which the patentee is involved. I have also been informed that the field of endeavor of a patent is not

limited to the specific point of novelty, the narrowest possible conception of the field, or the particular focus within a given field. I have also been informed that a prior art reference is reasonably pertinent if, even though it may be in a different field from that of the patentee's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to a patentee's attention in considering his problem.

17. I have also been informed that an analysis of whether a claimed invention would have been obvious should be considered in light of the scope and content of the prior art, the differences (if any) between the prior art and the claimed invention, and the level of ordinary skill in the pertinent art involved. I have been informed as well that a prior art reference should be viewed as a whole.

18. I have also been informed that in considering whether an invention for a claimed combination would have been obvious, I may assess whether there are apparent reasons to combine known elements in the prior art in the manner claimed in view of interrelated teachings of multiple prior art references, the effects of demands known to the design community or present in the market place, and/or the background knowledge possessed by a person having ordinary skill in the art. I have been informed that other principles may be relied on in evaluating whether a claimed invention would have been obvious, and that these principles include the following:

- A combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results;
- When a device or technology is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or in a different one, so that if a person of ordinary skill can implement a predictable variation, the variation is likely obvious;
- If a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill;
- An explicit or implicit teaching, suggestion, or motivation to combine two prior art references to form the claimed combination may demonstrate obviousness, but proof of obviousness does not depend on or require showing a teaching, suggestion, or motivation to combine;
- Market demand, rather than scientific literature, can drive design trends and may show obviousness;

- In determining whether the subject matter of a patent claim would have been obvious, neither the particular motivation nor the avowed purpose of the named inventor controls;
- One of the ways in which a patent's subject can be proved obvious is by noting that there existed at the time of invention a known problem for which there was an obvious solution encompassed by the patent's claims;
- Any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed;
- "Common sense" teaches that familiar items may have obvious uses beyond their primary purposes, and in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle;
- A person of ordinary skill in the art is also a person of ordinary creativity, and is not an automaton;
- A patent claim can be proved obvious by showing that the claimed combination of elements was "obvious to try," particularly when there is a design need or market pressure to solve a problem and there are a

finite number of identified, predictable solutions such that a person of ordinary skill in the art would have had good reason to pursue the known options within his or her technical grasp; and

- One should be cautious of using hindsight in evaluating whether a claimed invention would have been obvious.

19. I have further been informed that, in making a determination as to whether or not the claimed invention would have been obvious to a person of ordinary skill, the Board may consider certain objective factors if they are present, such as: commercial success of products practicing the claimed invention; long-felt but unsolved need; teaching away; unexpected results; copying; and praise by others in the field. These factors are generally referred to as "secondary considerations" or "objective indicia" of nonobviousness. I have been informed, however, that for such objective evidence to be relevant to the obviousness of a claim, there must be a causal relationship (called a "nexus") between the claim and the evidence, and that this nexus must be based on what is claimed and novel in the claim rather than something in the prior art. I have also been informed that even when they are present, secondary considerations may be unable to overcome primary evidence of obviousness (*e.g.*, motivation to combine with predictable results) that is sufficiently strong.

B. Claim Construction

20. I have been informed that for purposes of this Inter Partes Review, the standard for claim construction is the same as the standard used in federal district court litigation: claim terms should generally be given their ordinary and customary meaning as understood by one of ordinary skill in the art at the time of the invention and after reading the patent and its prosecution history.

21. I have been informed that patent claims are construed from the viewpoint of a person of ordinary skill in the art of the patent at the time of the alleged invention. I have been informed that patent claims generally should be interpreted consistent with their plain and ordinary meaning as understood by a person of ordinary skill in the art in the relevant time period (*i.e.*, at the time of the purported invention, or the so called "effective filing date" of the patent application), after reviewing the patent claim language, the specification and the prosecution history (*i.e.*, the intrinsic record).

22. I have further been informed that a person of ordinary skill in the art must read the claim terms in the context of the claim itself, as well as in the context of the entire patent specification. I understand that in the specification and prosecution history, the patentee may specifically define a claim term in a way that differs from the plain and ordinary meaning. I understand that the prosecution history of the patent is a record of the proceedings before the U.S. Patent and

Trademark Office, and may contain explicit representations or definitions made during prosecution that affect the scope of the patent claims. I understand that an applicant may, during the course of prosecuting the patent application, limit the scope of the claims to overcome prior art or to overcome an examiner's rejection, by clearly and unambiguously arguing to overcome or distinguish a prior art reference, or to clearly and unambiguously disavow claim coverage.

23. In interpreting the meaning of the claim language, I understand that a person of ordinary skill in the art may also consider "extrinsic" evidence, including expert testimony, inventor testimony, dictionaries, technical treatises, other patents, and scholarly publications. I understand this evidence is considered to ensure that a claim is construed in a way that is consistent with the understanding of those of skill in the art at the time of the alleged invention. This can be useful for technical terms whose meaning may differ from its ordinary English meaning. I understand that extrinsic evidence may not be relied on if it contradicts or varies the meaning of claim language provided by the intrinsic evidence, particularly if the applicant has explicitly defined a term in the intrinsic record.

V. DESCRIPTION OF THE RELEVANT FIELD AND THE RELEVANT TIMEFRAME

24. I have reviewed the '856 patent and its prosecution history in detail.

25. I understand that the ’856 patent issued from U.S. Patent Application 14/757,638, filed on December 23, 2015.

26. I have been instructed by counsel to assume the relevant timeframe for my analysis in this declaration to be on or before December 23, 2015.

27. Based on my review of this material, I believe that the relevant general field for the purposes of the ’856 patent is network security systems, including firewalls and other packet filtering devices for protecting computer networks from cyber-attacks.

VI. THE PERSON OF ORDINARY SKILL IN THE RELEVANT FIELD IN THE RELEVANT TIMEFRAME

28. My opinions are based on what a person of ordinary skill in the art in the technical field of the alleged invention would have understood at the time of the alleged invention of the challenged claims. As I explained above, the relevant timeframe in this Declaration is on or before December 23, 2015. I have been informed and understand that the content of a patent and prior art should be interpreted the way a person of ordinary skill in the art would have interpreted those references at the time of the alleged invention of the patent using the ordinary and customary meanings of the claim terms. I understand that the person of ordinary skill in the art is a hypothetical concept referring to one who thinks along the lines of conventional wisdom at the time.

29. I was asked to provide an opinion as to the level of a person of ordinary skill in the art pertinent to the subject matter set forth in the challenged claims of the ’856 patent at the time of the priority date. I have been informed that the level of one of ordinary skill in the art is evidenced by prior art references. A person of ordinary skill in the art would have had a bachelor’s degree in computer science, computer engineering or an equivalent, as well as four years of industry experience and would have had a working knowledge of communications network, firewalls, security policies, and packets filtering. Additional graduate education could substitute for professional experience, or significant experience in the field could substitute for formal education.

30. The basis for my familiarity with the level of ordinary skill is my own technical experience and my interaction with students and professionals in the computing field who were at this level of skill. In reaching this opinion as to the hypothetical person of ordinary skill in the art, I have considered the types of problems encountered in the art, the prior art solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the educational level and professional capabilities of workers in the field. As summarized above with regards to my qualifications and experience, I was at least the level of a person of ordinary skill in the art at the time of the ’856 patent.

VII. TECHNICAL BACKGROUND AND STATE OF THE ART

31. The following analysis shows that the concept of network security systems and firewalls were well known in the art prior to December of 2015. The following sections describe the historical development of the art for network security systems, including firewalls and other packet filtering devices for protecting computer networks from cyber-attacks, and the user interface systems for controlling them, leading up to the state of the art as of December 2015. The prior art references, and the discussions of what was known to a person of ordinary skill in the art, provide the factual support for the general description of the state of the art at the time of the ’856 Patent. This section provides additional rationale for obviousness over Buruganahalli (Ex. 1004), obviousness over and motivation to combine the teachings of Buruganahalli in view of Baehr (Ex. 1005) with the knowledge, skill, and creativity of a person of ordinary skill in the art, provides support for why a person of ordinary skill in the art would have had a reasonable expectation of success in doing so, rebuts any assertion by the Patent Owner of unpredictability in the art, and rebuts any assertion by the Patent Owner of unexpected results.

32. This overview is based on the “State of the Art” summary provided by Dr. Stuart Staniford in his declaration in support of the petition for IPR of U.S. Pat. No. 9,413,722 (the ’722 patent) (IPR2018-01760), filed by Cisco System, Inc.

The'722 patent is directed to similar basic concepts and technology as the '856 patent. For that reason, the state of the art relevant to the '722 patent is necessarily also relevant to the '856 patent. Any similarity between Dr. Staniford's "State of the Art" overview and my own is intentional. I reviewed his opinions on the state of the art and the supporting documents cited therein, and I agree with those opinions as set forth herein.

33. As of December 2015, a person of ordinary skill in the art working in the field of network security systems, including firewalls and other packet filtering devices for protecting computer networks from cyber-attacks, would have been familiar with certain common concepts and related components including, but not limited to: computer network architectures and communication protocols, firewalls, Intrusion Detection Systems and Intrusion Prevention Systems, the layering of computer network protocols, the use of network threat indicators to prevent cyber-attacks, and filtering unencrypted and encrypted packet data based on such indicators.

A. Computer Network Architectures and Communication Protocols

34. The Internet has its roots in conceptual papers on packet switching in the 1960s. These papers led to the funding by DARPA of an experimental internetwork – the ARPANET – with the first two nodes being connected in late 1969, and then growing steadily through the 1970s as a means of doing research on

computer networking. Ex. 1020 (Doomjuice). The early ARPANET used a protocol, NCP, that is no longer in use and proved not to be scalable.

35. NCP's deficiencies led to the development of two new protocols – Internet Protocol (IP) to provide best effort delivery of data packets across heterogeneous networks (Ex. 1021 (Helix)), and Transmission Control Protocol (TCP) to provide reliable delivery of streams of data across an IP internetwork (Ex. 1023 (RFC 791)). These two protocols were described in the IETF Request for Comment (RFC) documents RFC 791 and 793 respectively in 1981. Ex. 1022 (RFC 793); Ex. 1021 (Helix). The ARPANET cut over to using the new TCP/IP protocol suite on January 1, 1983. That date can be viewed as the birth date of the modern Internet. Ex. 1007 (Leiner).

36. Logically, the Internet is a web of intermediate network nodes and communication paths interconnecting those network nodes that provide multiple data transmission routes from any given point to any other given point on the network (i.e., between any two computers.) Ex. 1008 (Jungck '06), 3:59-4:11. The disparate networks are connected where they join by special devices – IP routers – that decide which way to send each packet. A router typically creates or maintains a table of the available routes and their conditions and uses this information to

determine the best route for a given packet. Ex. 1008 (Jungck '06), 3:32-46, 38:29-39:8.

37. While TCP/IP provided the basic abstract facility to move data across a global network, higher-level application protocols were required to allow data communication to be managed conveniently. Application protocol data would be transmitted in the body of packets with IP and TCP headers. Particularly important applications included email (with protocols such as SMTP, POP and IMAP), with early email systems dating back to 1972. Ex. 1007 (Leiner). Other important protocols included FTP for making files available, first defined in 1980 (Ex. 1024 (RFC 765)), and the Hypertext Transfer Protocol (HTTP) for facilitating the exchange of hypertext markup language (HTML) and World Wide Web (WWW) content, first defined in 1991 (Ex. 1025 (T. B. Lee)).

B. Network Security Systems – Firewalls

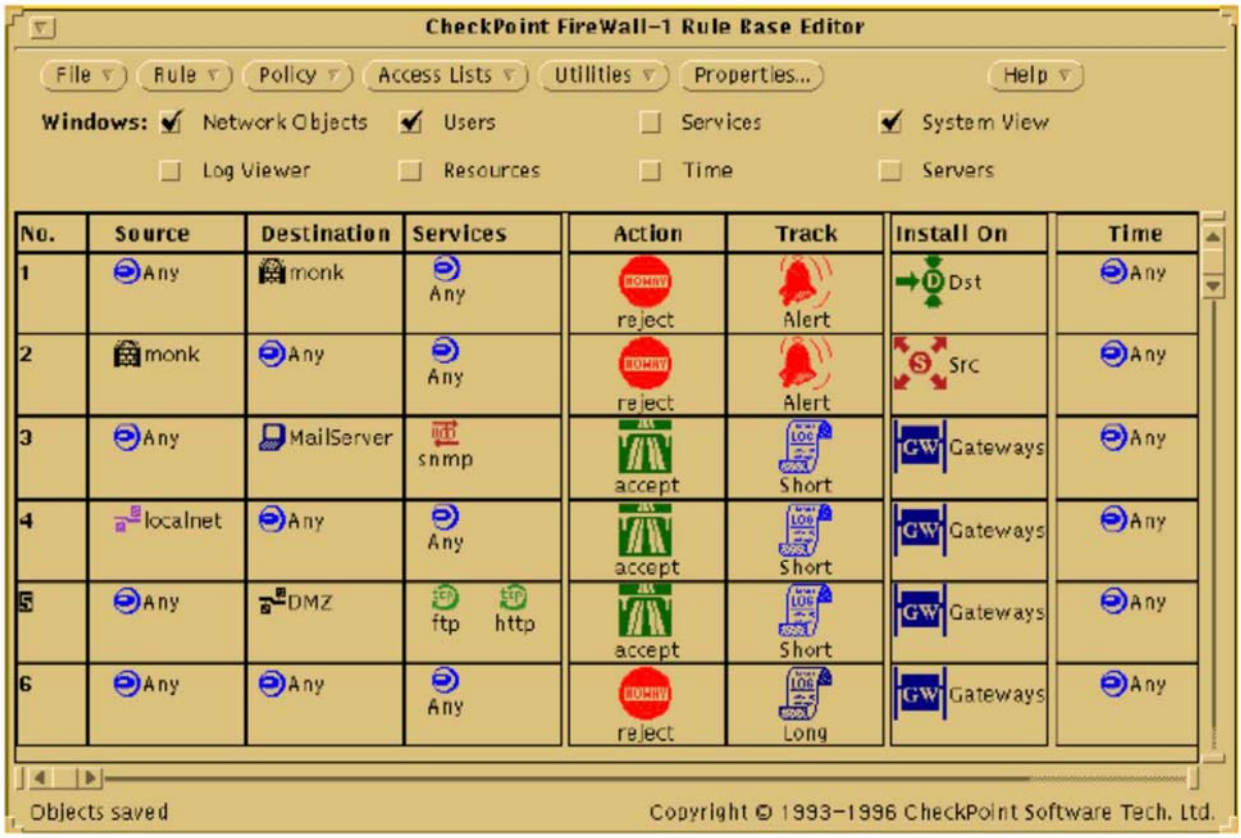
38. Throughout the mid to late 1980s, the Internet grew very rapidly as more and more researchers and developers connected departmental mainframes, minicomputers, Unix workstations, and the like to it. The Internet reflected the openness and information sharing that characterizes the research community that developed it, with few security barriers beyond the requirement to login on remote systems (which often had weak or default passwords). Ex. 1007 (Leiner).

39. This era of complete openness spawned a variety of abuses by computer hackers. A particularly spectacular and famous incident spawned the modern network security industry: the Internet worm of 1988. In this episode, a Cornell graduate student, Robert Morris Jr., released a worm – a self-replicating program capable of infecting other computers across the network via vulnerabilities in their operating system or applications. This worm infected about 6,000 computers across the Internet. Most of these computers were disabled, either by large numbers of copies of the worm clogging their memory, or at least by the need to disinfect and upgrade them. Many of the computers were important shared facilities with numerous users, so much of the work on the Internet was disrupted. The incident caused a great deal of analysis and soul-searching amongst the Internet community. Within a year (by 1989), both the first firewall and the first network intrusion detection system (NIDS) were under development. Ex. 1009 (Spafford), Ex. 1010 (Eichin), Ex. 1011 (Eisenberg).

40. The development of firewalls had its roots in the late 1980s practice of using filtering rules on Internet routers. This development began primarily as a way to prevent misconfigurations on one network that generated a lot of noise packets from affecting an adjacent network. However, it then resulted in the development of security applications, with routers providing simple facilities to block packets based

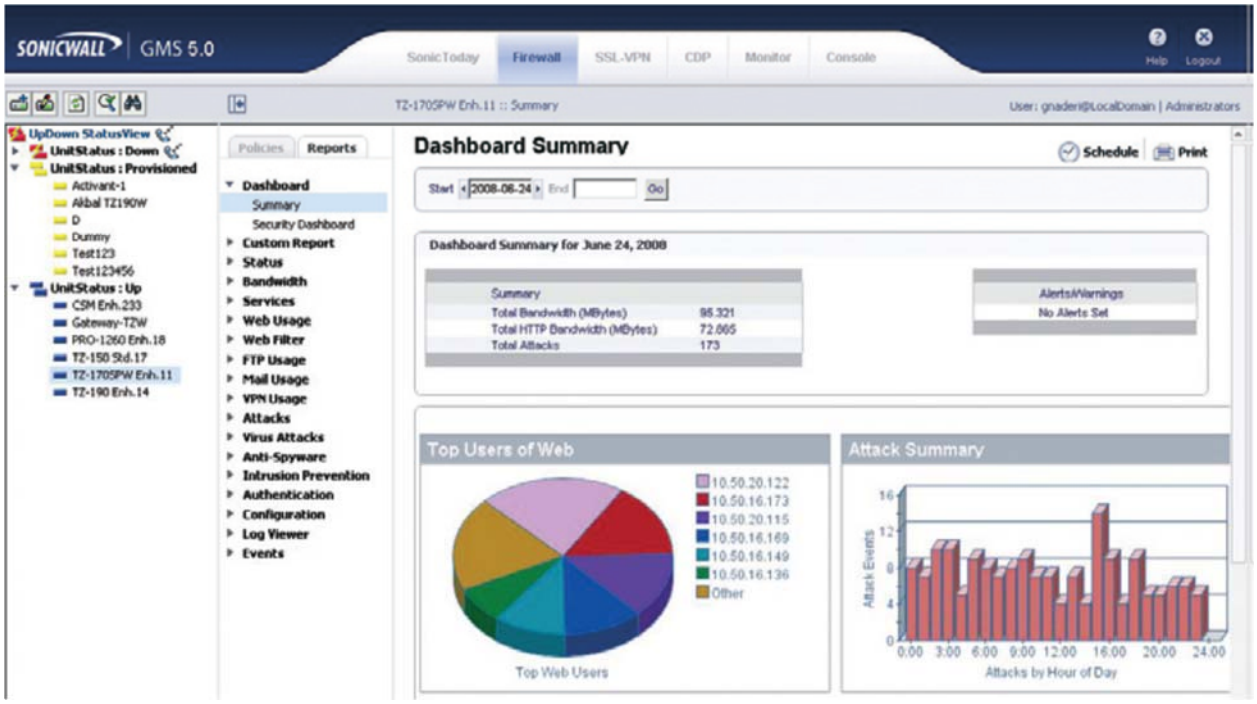
on their source and destination networks and TCP ports. For example, with Cisco routers in 1989: "one could deny access to all but one host on a particular network. The Cisco gateways also allow discrimination based on IP protocol type, and TCP or UDP (User Datagram Protocol) port numbers." Ex. 1012 (Mogul), 3. This example illustrates that the basic functionality of blocking or admitting packets based on a rule was available prior to 1989.

41. User interfaces for firewalls developed soon after the introduction of the firewalls themselves. A particularly influential company was CheckPoint, with their FireWall-1 product. A screenshot dated 1996 of this product is below:



42. This demonstrates a view of the firewall rules, with their five tuple information (source, destination, services, etc.), and the actions the rule will take (such as rejecting or accepting the connection attempt, and alerting the user or not). Ex. 1013 (FireWall-1).

43. Firewall system user interfaces also came to provide dashboards with overview information about the operation of the firewall such as graphs of summary statistics and tables of top things of importance. A typical example from a SonicWALL firewall in 2009 is shown below:



44. Here is shown graphs with the number of attacks detected per hour in the recent time period, as well as the top source IPs using bandwidth over HTTP

connections. Also, there is a table with current bandwidth usage across the firewall. Ex. 1014 (SonicWALL GMS), 2.

45. This system also provided functionality similar to packet logs. For example, the screenshot below shows the display of a listing of connections blocked, which a person of ordinary skill in the art would understand is stored within the system's persistent memory:

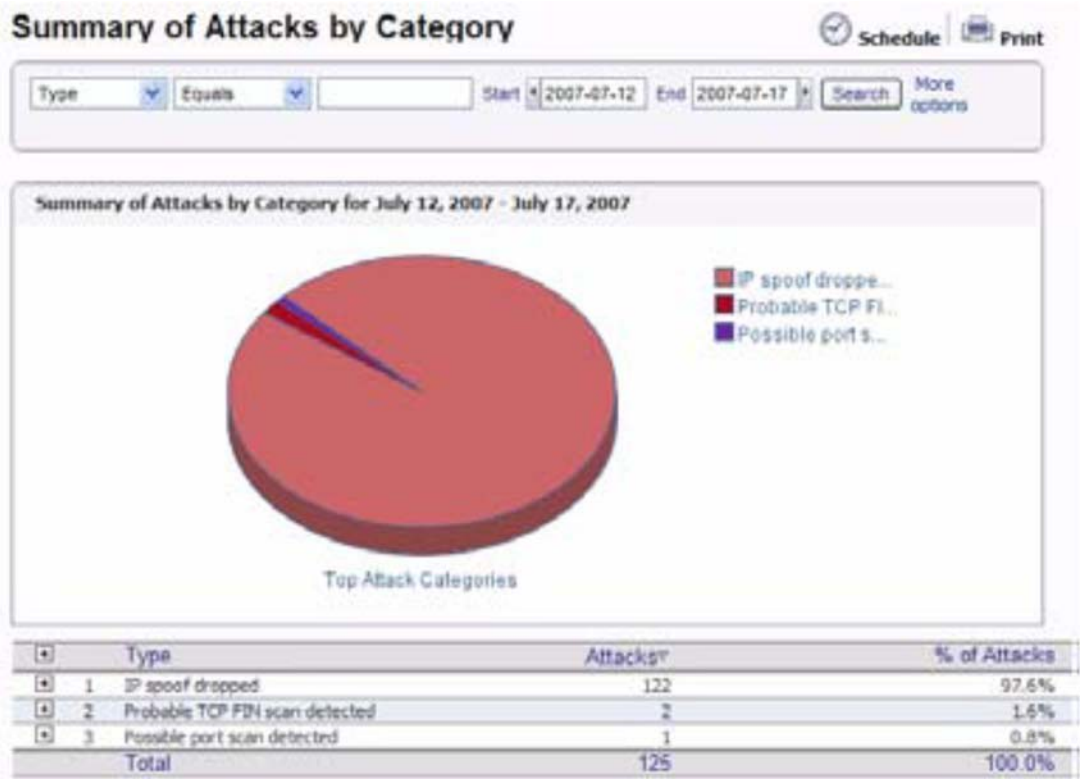
Log Search Results for May 1, 2007 00:00:00 to May 1, 2007 23:59:59 as of 05/01/2007 19:11:47 UTC
Message Category: Connections

Displaying records 1-500 of 25715 Next 500 (1 2 3 4 5 6 7 8 9 10)

Time[1]	Source/User	Destination	Protocol	Src Port	Dst Port	Src Interface	Dst Interface	TX Byt
05/01/2007 12:11:47	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:47	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:47	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:46	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:46	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:44	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:43	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:43	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:43	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:43	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:25	10.0.202.131	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:24	10.0.202.131	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:24	10.0.202.131	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:15	10.0.202.131	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:14	10.0.202.131	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:14	10.0.202.131	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:12	10.0.203.179	255.255.255.255	udp/0	68	0	WAN	LAN	0
05/01/2007 12:11:09	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:09	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:09	0.0.0.0	255.255.255.255	udp/0	68	0	WAN	LAN	0
05/01/2007 12:11:09	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:08	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0
05/01/2007 12:11:08	0.0.0.0	255.255.255.255	udp/0	68	0	WAN	LAN	0
05/01/2007 12:11:08	10.0.56.39	224.0.0.22	igmp	34	0	WAN	N/A	0

46. The same system was capable of providing functionality akin to packet logs or flow logs. Ex. 1015 (SonicWALL ViewPoint), 292. For example, the screenshot below (Ex. 1015 (SonicWALL ViewPoint), 253) shows a table of types

of attacks the firewall is capable of detection and the counts of them occurring over a time period.



		Type	Attacks	% of Attacks
	1	IP spoof dropped	122	97.6%
	2	Probable TCP FIN scan detected	2	1.6%
	3	Possible port scan detected	1	0.8%
		Total	125	100.0%

C. Network Security Systems – Intrusion Detection Systems and Intrusion Prevention Systems

47. In an independent line of development from firewalls, the Internet worm and other less serious incidents of the time inspired researchers to begin exploring systems specifically designed to look for intrusions on networks by examining the data packets crossing the network. Initially, network intrusion detection was done by placing network interfaces on broadcast networks into "promiscuous" mode to inspect all the traffic, which could then be copied into a

user-mode application where it could be inspected and reasoned about. The earliest such system was the Network Security Monitor (NSM) developed at University of California, Davis in 1991. Ex. 1016 (Heberlein). This evolved into the U.S. Air Force's Automated Security Incident Measurement (ASIM) system, and a large part of the development team on that effort split off and formed a company, WheelGroup, which launched the first commercial network intrusion detection system (IDS) NetRanger in 1996.

48. NetRanger had a general facility to write rules that matched arbitrary strings in the application data inside packets, in addition to matching on IP and TCP packet header values. For example, a 1997 user guide for the NetRanger system illustrated how to log every HTTP GET request passing by in packets on the monitored network:

The following example shows how to log every HTTP GET on a network. The context info provided with the string will show what URL was used in the GET.

Web	SIGID	Port	Direction	Occurrences	String Match
RecordOfStringName	8001	80	1	1	"GET[ ]+"

Ex. 1017 (NetRanger), 189.

49. NetRanger also had a GUI For example, the same user guide from 1997 provides a Figure 4.10 (Ex. 1017 (NetRanger), 127) which shows a view of an

individual 'packet log' type display with the source and destination addresses, port numbers and so forth, and the 'Threat-ID' (here 'Alarm Name' Sweep-Echo):

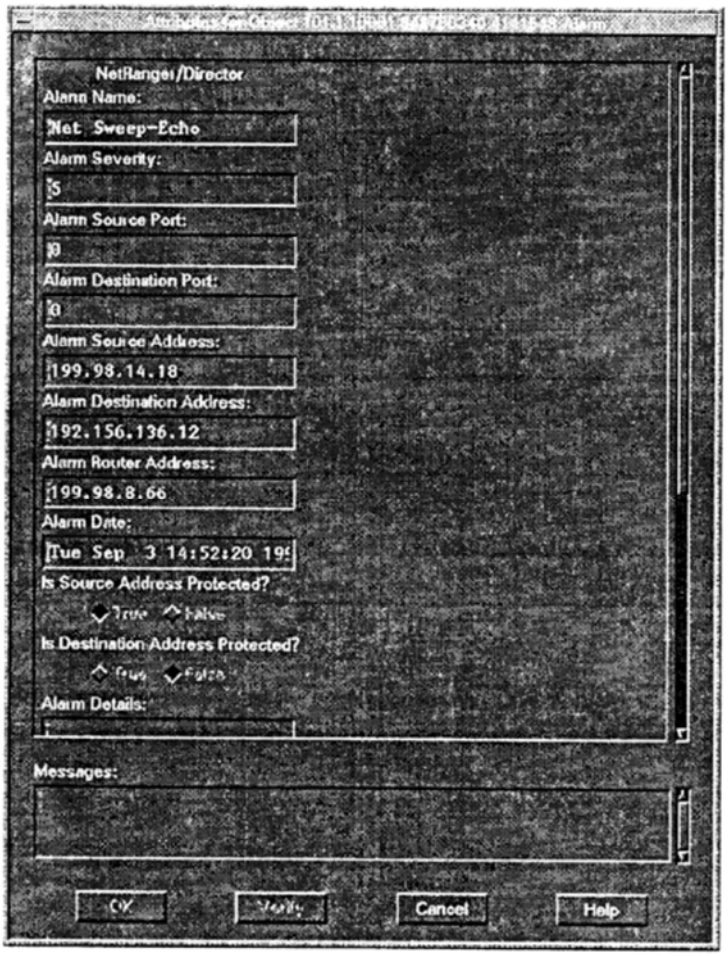


Figure 4.10: Alarm Event Attributes

50. Additionally, NetRanger had user-interface functionality for a user to block at least some traffic based on detection of a rule firing. For example, the following (Ex. 1017 (NetRanger), 135):

Shunning IP Addresses and Class C IP Networks

The **Security→Shun** functions can be used to manually shun (block) incoming IP traffic. To shun traffic that emanates from an Alarm's Source IP Address, select the Alarm (or Alarm Set), and choose **Security→Shun→Source IP**. To shun traffic that emanates from any IP address within the Class C Network that contains an Alarm's Source IP Address, select the Alarm (or Alarm Set), and select **Security→Shun→Source Net**.

In either case, an xnmappmon window will display the output of the nrexec command that was used to shun the traffic.

This user action would reconfigure a packet filtering router to block all packets from the source IP address used in a detected attack. Ex. 1017 (NetRanger), 135.

51. The first widely adopted open source network intrusion detection system (NIDS or just IDS), Snort, was first released in 1998, and was documented in a conference paper in 1999. Ex. 1018 (Roesch). Like other NIDS before it, Snort allowed the user to write rules for detecting particular network packets based on both the 5-tuple information in the packet IP and TCP headers, as well as strings found in packet application data. For example:

```
alert tcp any any -> 10.1.1.0/24 80 (content: "/cgi-bin/phf"; msg: "PHF
probe!";)
```

This rule (see Ex. 1018 (Roesch), 232, Fig. 5) shows that alerting was to be done based on TCP protocol packets from any source IP address and port, and destination IP addresses in the range 10.1.1.0/24, and with destination port 80 (characteristic of HTTP traffic). This particular content keyword (/cgi-bin/phf) matches a uniform

resource locator (URL) used in a popular attack of the time, but content keywords in general could be used to find and match any text in application data in packets transiting the network.

52. The earliest NIDS were strictly involved in detecting attacks; these systems worked off a copy of the packets and could only alert or log suspected intrusions, not directly block them. However, it was quickly realized that customers had a need to take automated action in some cases. While the first instance of this involved communication to firewalls or filtering routers, the need to block attacks quickly, as well as the need to resolve various complex ambiguities in the meaning of packet traffic that computer intruders used to get around the first NIDS, meant that the industry quickly developed systems that were "inline" in the traffic flow, and which could choose to drop or pass any given packet. These systems were known as intrusion prevention systems (IPS), with multiple companies launching such products in 2002, including Intruvert (later acquired by McAfee), Tipping Point (3Com then HP), and OneSecure (later acquired by NetScreen then Juniper).

53. These inline intrusion prevention systems were capable of applying intrusion detection rules that looked both at 5-tuple header data and application data (e.g. content) in every packet and making pass/alert/block decisions on each such packet. For example, the Zuk patent (Ex. 1019), filed in 2002, describes an IPS

architecture in which packets were inspected, and the 5-tuple in the packet header was used to lookup the correct rules in a flow table (hash table of all TCP connections or UDP pseudo-connections currently being tracked by the system). The rules described in Zuk then allowed regular expression matching of any packet content in the application layer, and packet application data content matches would result in triggering the rule and dropping the packet. Ex. 1019 (Zuk), 18:4-32, Fig. 11. Zuk described in outline terms the use of user-interface functionality for controlling the IPS:

54. "The information provided by MMIDP sensors 45a-c and MMIDP server 100 is organized in reports that provide access to a list of all the detected attacks and intrusions as well as network statistics that otherwise would be difficult to gather, such as the top IP addresses used in attacks, the top attacks, the number of alarms and incidents generated, and whether an alarm is real or false, among other statistics. In addition, network security administrators may specify which signatures from the set of signatures stored in MMIDP database 105 will be used to detect and prevent attacks, as well as create new signatures." Ex. 1019 (Zuk), 14:14-25.

55. Similarly, the Jungck patent from 2006 is directed to a "packet interceptor/processor apparatus" that "applies one or more rules to ... intercepted packets which execute one or more functions on a dynamically specified portion of

the packet and take one or more actions with the packets." Ex. 1008 (Jungck '06), Abstract. Actions taken on packets include "releasing the packet unmodified, deleting the packet, modifying the packet, logging/storing information about the packet or forwarding the packet to an external device for subsequent processing." Ex. 1008 (Jungck '06), Abstract. Further, "the rules may be dynamically modified by the external devices." Id.

D. Network Security Systems – Detailed Concepts

56. As discussed further below, each element of the Challenged Claims was well-known to a person of ordinary skill in the art, and disclosed and rendered obvious by the prior art, well before the earliest possible priority date of the '856 patent—December 23, 2015. Moreover, each combination recited by the Challenged Claims would have been, at least, obvious and was at most an obvious implementation choice.

57. The Challenged Claims are directed to well-known concepts of rule-based network-threat detection whereby a packet filtering system, such as a firewall, identifies packets, correlates unencrypted packets with encrypted packets, determines whether they are associated with a network threat indicator, filters them using rules based on predetermined criteria, and routes potentially threatening packets to a proxy system. The Challenged Claims further recite generic and well-known functionalities and components for performing these well-known concepts,

including a "processor," a "memory," "computer-readable media," "instructions," "packet-filtering rules," "network-threat indicators," "unencrypted data," "encrypted data," and "a proxy system." '856 patent, claims 1, 24-25.

58. Each of these concepts was well-known in the art and disclosed in the prior art.

1. Applying Rules to Filter Packets in a Packet Filtering System

59. It was well-known and conventional in the art before the claimed priority date of the '856 patent for a network security system to include a rule-based packet filtering system, e.g., a firewall or a screen, to protect a network from outside communications. A packet-filtering system, as its name indicates, filters packets to and from a system or network. It does so by applying one or more rules that specify certain criteria, *e.g.*, based on source, destination, protocol version, etc., to identify packets that may present potential security risks to the network.

60. For example, U.S. Patent Appl. Pub. No. 2014/0283004 (Moore) (Ex. 1040) discloses a "packet security gateway" configured to receive packets and apply one or more filters to the packets. Moore, [0019]. The gateway is also configured to receive a security policy, which may include one or more rules, from a security policy management server. Moore, [0019], [0021]. The rules are then applied to filter packets. *Id.*

2. Identifying Security Risks Based on Network-Threat Indicators

61. It was further well-known and conventional in the art before the claimed priority date of the ’856 patent for a network security system to identify security risks based on network-threat indicators, e.g., domain name. It was a common technique for a packet-filtering system to use indicators to identify potential network threats that may indicate security risks to a network. These indicators were known to be based on data collected through threat monitoring and to include source and destination information, such as a domain name.

62. For example, U.S. Pat. App. Pub. No. 2015/0207809 (Macaulay) (Ex. 1041) discloses that for “threat monitoring,” the operator of “a given customer network” subscribes to a service that provides “cyber threat intelligence data.” Macaulay, [0025]. Figure 4 shows the process of collecting, creating, and providing cyber threat intelligence data to a customer network where the customer network “receives customized threat report and takes actions” in step 460. Macaulay, [0099].

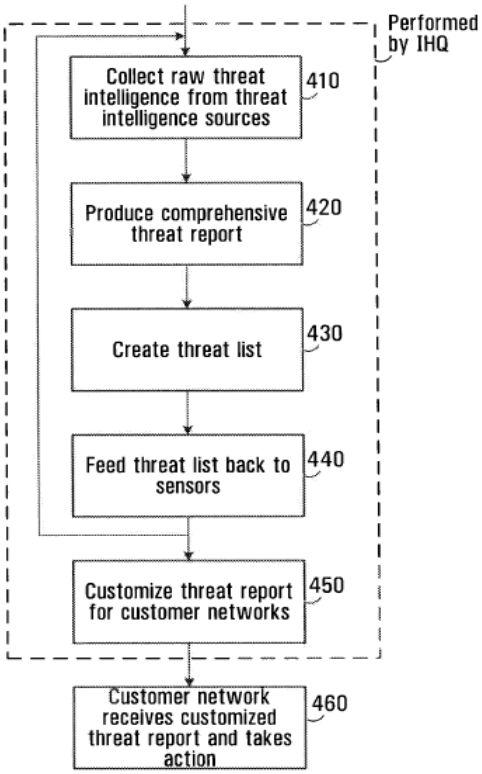


FIG. 4

The "cyber threat intelligence data" include "traffic attributes that are indicative of a threat." Macaulay, [0024]. For example, the data may "indicate, from the perspective of a given customer network [] threats that arise from an external entity and also threats against internal assets…" Macaulay, [0024]. The threat-indicators include "domain names (both as source and as destination)," "autonomous system numbers (ASNs, both as source and as destination)," "URL strings," and "protocol." Macaulay, [0024], [0030], [0037] ("network communication… protocols… are then closely observed for relatively simple criteria, such as… protocols…"), [0045] ("basic tools for… understanding the threats… include… IP protocol…"), [0047]-

[0048], [0056], [0060]-[0062], [0066], [0072]-[0073], [0075] ("it has been observed that P2P protocols are frequently used… by malware and botnets"), [0086], [0091], [0101], [0052] ("domain name identified as a potential threat (i.e., on a threat list [])"). They can be "cross-referenced with a list of … domains and ASNs with which malware is known to communicate." Macaulay, [0036]. Macaulay discloses that the customer network, after receiving the data comprising the threat-indicators from the subscription service, "may then apply the … data directly to traffic entering or existing… the customer network." Macaulay, [0025]. For example, the threat-indicators may be "incorporated into a filter" such that the data communications are monitored. Macaulay, [0047], [0052], [0056], [0061].

63. As another example, Moore discloses a filter that compares "packet header information (e.g., source and destination IP address(s), source and destination port(s), and protocol type(s)) with one or more packet filtering rules, which may define a network usage policy or network security policy." Moore, [0005]. Moore further discloses that "[e]ach of these rules may specify criteria and one or more operators that may be applied to packets associated with (e.g., matching) the specified criteria. The specified criteria may…specify[] a protocol type of the data section of an IP packet (e.g., TCP, User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), or one or more other protocols), one or more source IP

addresses, one or more source port values, one or more destination IP addresses, and one or more destination ports." Moore, [0025].

3. Correlating Packets of a Session

64. It was well-known and conventional for a firewall to determine whether packets, unencrypted or encrypted, belong to a given communications session using IP addresses and/or port numbers. As I discussed above, a network may use a transport-layer protocol called Transmission Control Protocol ("TCP"). TCP provides a data delivery service that monitors transmissions using control information in a header that is encapsulated within an IP packet. The header of the IP packet contains source and destination IP addresses. The TCP header contains port numbers that act like addresses and that may provide an indication of which underlying communication protocols are associated with a packet (e.g., HTTP, FTP). Combined, the IP source address, IP destination address, source port number, destination port number, along with an identifier of the transport protocol in use (e.g., TCP), uniquely identify all the packets exchanged during a communication session between a source computer (e.g., a client) and a destination computer (e.g., a server). Kim (Ex. 1045), 660 ("Each session can be identified by 5-tuple packet information which is source IP address, destination IP address, protocol, source port and destination port."); Zuk, 15:32-53, 16:13-26, Fig. 8; Boivie (Ex. 1044), 4:42-50; Moore, Fig. 3, [0025]. This 5-tuple, called a flow id, has been used to identify

packets within a single end-to-end (original sender to ultimate destination) communication session since the creation of the Internet. Zuk, 15:33-53, 16:13-26, Fig. 8; Boivie, 5:11-18; Moore, [0039]. Secure protocols such as SSL and TLS run on top of the IP and TCP layers, and are used to encrypt application layer data. Wang (Ex. 1034), 3:47-50. The IP addresses, port numbers, and other TCP session data associated with a packet are typically unencrypted and may be inspected without decrypting the packet. Moore, [0036].

4. Inspecting Handshake Messages Prior To Establishing Encrypted Sessions.

65. It was further well-known and conventional in the art before the claimed priority date of the ’856 patent for a network security system to inspect unencrypted handshake messages prior to establishing encrypted sessions. It was a common technique to evaluate the security risk of establishing an encrypted session by inspecting the session’s handshake message to determine whether it comprises any data that may correspond to a network threat indicator.

66. For example, U.S. Patent Appl. Pub. No. 2014/0089661 (Mahadik**)** (Ex. 1042) discloses a method for selectively filtering internet traffic. Mahadik, Abstract. Mahadik recognizes that “[f]or SSL/HTTPS based website access, the network traffic is encrypted and thus cannot be monitored with the same tools used in unencrypted scenario.” Mahadik, [0029]. Mahadik proposes a method to detect the

encrypted traffic by examining its handshake. *Id.* Specifically, Mahadik discloses, "[t]his preferably occurs when a site is being accessed over HTTPS using a SSL certificate of a server during a handshake. A domain is preferably detected during the handshake through a server name attribute or through some alternative parameter. The web proxy server may subsequently determine if the domain is restricted, permitted, or partially restricted. If the domain is restricted, the access may be blocked entirely. If the domain is permitted, the web proxy preferably hands client requests to the server and the server responses back to the client without making any modification to the tunneled SSL traffic." *Id.*

67. Moore discloses that an enterprise may "enforce one or more network security policies…to restrict network communications between networks 104 and 106 (e.g., … encrypted web sessions, and the like)." Moore, [0031]. Further, rules are applied to determine whether a session can be conducted: "rule 1 302 may allow… hosts attached to network 104 to conduct Secure Shell (SSH) sessions…; rule 2 304 may allow any e-mail servers…to conduct SMTP sessions (e.g., e-mail transfer sessions) with any e-mail servers…; rule 3 306 and rule 4 308, may respectively allow e-mail clients attached to network 104 to conduct POP3 and IMAP sessions (e.g., e-mail download sessions into a webmail browser application)…; and rule 5 310 may allow web browsers attached to network 104 to

conduct HTTPS sessions (e.g., secure web sessions)…but may…utilize the REQUIRE-TLS-1.1-1.2 operator to ensure that only HTTPS secure web sessions using version 1.1 or 1.2 of the Transport Layer Security (TLS) protocol to secure such HTTPS sessions are allowed (e.g., because the popular TLS version 1.0 protocol has a known security vulnerability that attackers may exploit to decrypt HTTPS sessions)." Moore, [0031].

5. Routing Packets Between Functions of A Security Device

68. It was further well-known and conventional in the art before the claimed priority date of the '856 patent for a network security system to route packets between interfaces of functional components of a security device such as a firewall. This technique was widely implemented using a software routing function.

69. For example, J. Wack et al., *Guidelines on Firewalls and Firewall Policy*, National Institute of Standards and Technology ("NIST") (Ex. 1036) describes the characteristics of an application-proxy gateway firewall. NIST discloses, "[a]pplication-proxy gateway firewalls do not require a Layer 3 (Network Layer) route between inside and outside interfaces of the firewall; the firewall software performs the routing. In the event the application-proxy gateway software ceases to function, the firewall system is unable to pass network packets through the firewall system. All network packets that traverse the firewall must do so under software (application-proxy) control." NIST, 12. The application-proxy gateway

firewalls use a software application running on a firewall that is "capable of filtering a protocol and routing it [*sic*] between the interfaces of the device." NIST 52. NIST further discloses using a firewall rule set, which is a table of instruction, "for determining how packets should be routed between its interfaces." NIST 51.

6. Sending Suspicious Packets to a Proxy System

70. It was further well-known and conventional in the art before the claimed priority date of the '856 patent for a network security system to send packets that may present a risk to a proxy system for further processing. This technique was widely implemented to isolate a network from direct access or attacks from an outsider and uses a proxy system in its place to handle packets that may present a network threat. Proxies also provide more efficient use and handling of limited network resources. Ebrahimi (Ex. 1035), 1:25-28

71. There are two types of proxies that are generally known to handle network traffic between a client and a server: a forward proxy and a transparent proxy. Ebrahimi, 1:37-1:45. For example U.S. Patent No. 7,793,342 ("Ebrahimi") describes the characteristics of a forward proxy and a transparent proxy:

> A forward proxy is known to and configured by each of the clients, before the clients can take advantage of the forward proxies capabilities. In contrast, a transparent proxy is neither known to nor configured by the clients; rather, the transparent proxy intercepts outgoing client requests to access origin

> servers/sites and transparently processes the requests on behalf of the clients. Therefore, transparent proxies can be more secure than forward proxies, since the clients do not know about the transparent proxies. Furthermore, transparent proxies can be more easily integrated into a network than forward proxies, since no client configuration is required.

Ebrahimi, 1:45-1:56.

72. A transparent proxy system, as its name suggests, is "transparent" to two endpoints (*e.g.*, a client and a server) in a communication. U.S. Patent No. 8,504.822 ("Wang") discloses a transparent proxy between a server and clients in a computer network, as shown in Figure 1. Wang, 2:32-58.

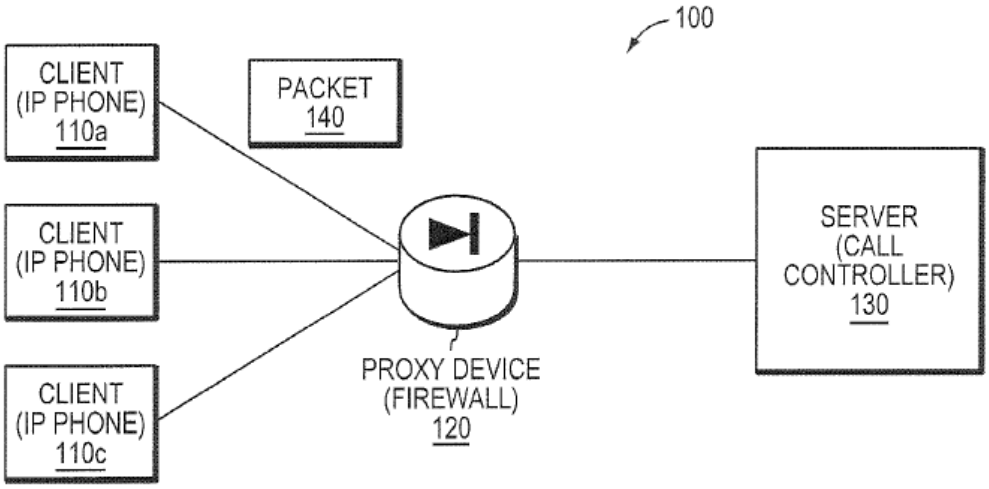


FIG. 1

Wang explains:

> because the proxy device 120 is a transparent intermediary to the client-server security session, the proxy device is able to decrypt the received traffic/packets (*e.g.*, using the session key negotiated for the client-proxy or proxy-server session, as appropriate) from each sending end device, and may re-encrypt the traffic prior to forwarding it on using the respective session key with the receiving end device.

Wang, 8:12-19. Wang explains that "neither the client nor the server is required to be provisioned to specifically communicate with the proxy device, and instead may simply attempt to directly connect to the peer (server or client, respectively) over an apparent client-server security session." Wang, 7:53-57. Furthermore, "[w]ith the established client-proxy and proxy server security sessions acting transparently as a client-server security session, the proxy device may be allowed to inspect the traffic between the client and the server [], such as by decrypting and re-encrypting the traffic accordingly." Wang, 8:67-9:4.

73. A proxy system may be implemented internally to, or implemented externally from, a packet filtering device such as a firewall or a screening system. For example, NIST discloses that a proxy is "a software application running on a firewall or on a dedicated proxy server…" NIST 52. "Dedicated proxy servers differ from application-proxy gateway firewalls in that they retain proxy control of traffic

but they do not contain firewall capability" and they are "typically deployed behind traditional firewall platforms for this reason." NIST, 14. Furthermore, a main firewall "might accept inbound traffic, determine which application is being targeted, and then hand off the traffic to the appropriate proxy server." NIST, 14. NIST further discloses that "[d]edicated proxy servers perform "filtering or logging operations on the traffic and then for forward it to internal systems." NIST, 14. They are typically used to "decrease the work load on the firewall and to perform more specialized filtering and logging that otherwise might be difficult to perform on the firewall itself." NIST 14.

74. As discussed, diverting traffic to a proxy is beneficial to protect against potential attacks. For example, U.S. Patent Application Publication No. 2007/0271605 (Le Pennec) (Ex. 1033) discloses routing packets to a proxy for security reasons. It illustrates the functional blocks of a firewall device in Figure 3:

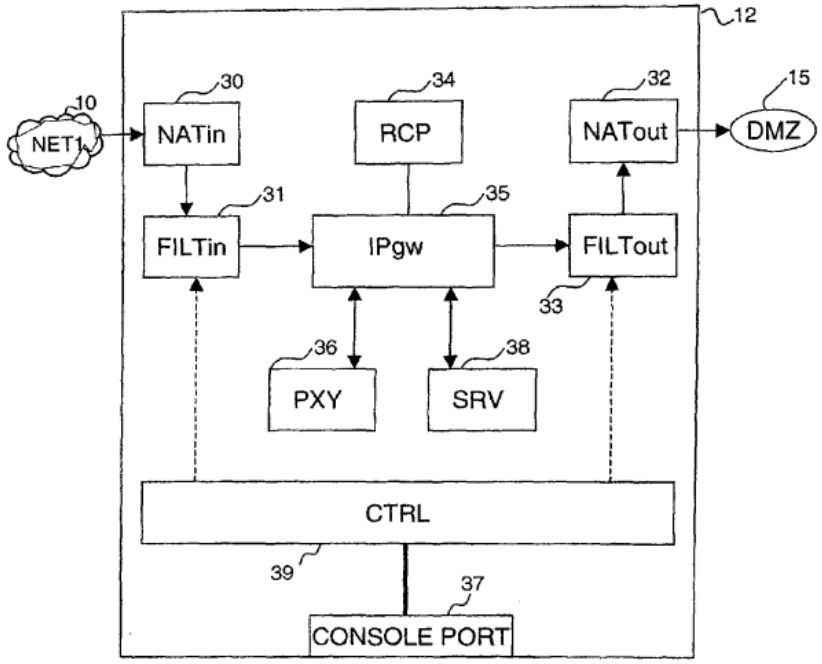
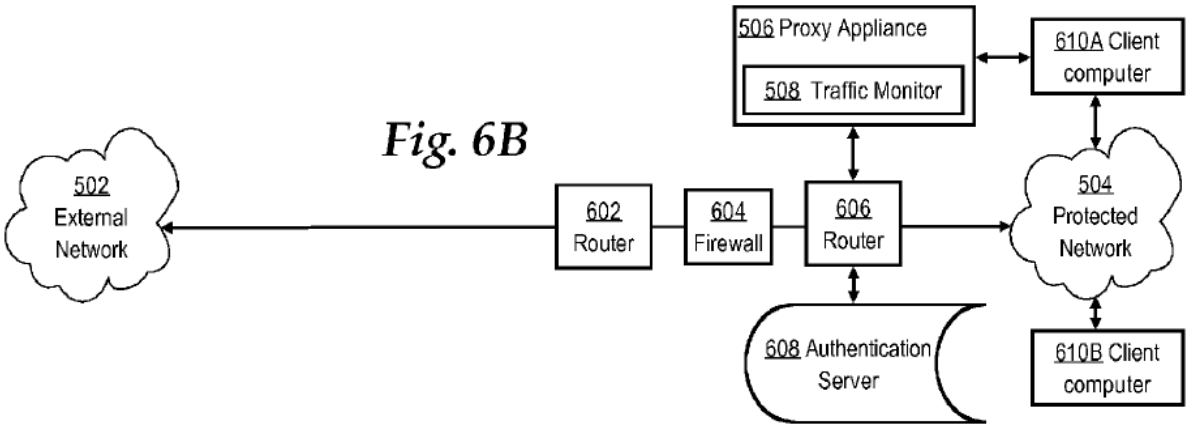


FIG. 3

The firewall filters incoming packets using "FILTin 31" and then forwards the packets to "IPgw 35" that is a forward engine. The forward engine uses a "routing control point RCP" that makes routing decisions based on certain rules. Based on these rules, the packets may be sent to "an integrated function, such as an application PXY 36" or a server SRV 38 for further processing. Le Pennec, [0022]; *see also id.*, [0006]-[0008], [0023]-[0044].

75. As another example, U.S. Patent No. 7,849,502 (Bloch) (Ex. 1043) discloses that a proxy "provides protection against spyware and web-based malware, including managing network traffic into and out of an internal network to block or

redirect the traffic to avoid malware, such as spyware…" Bloch, 3:35-41. Bloch illustrates an example of using a proxy appliance to protect a network. Figure 6B illustrates that traffic that has been filtered by a firewall is routed to a proxy appliance for further processing. Bloch, 6:52-67.



"Referring to FIG. 6B, external network 502 is coupled to proxy appliance 506 through an edge router 602 and firewall 604 that protect the protected network 504. A second router 606, which may be located anywhere within network 504, is coupled to the proxy appliance 506 and authentication server 608. The second router 606 may comprise a WCCP router or a Layer 4 switch. Client computers 610A, 610B may be coupled to protected network 504 and one or more of the client computers may connect to the proxy appliance 506 in forward mode." Bloch, 8:29-39.

76. As yet another example, as disclosed above, Mahadik discloses "[r]edirecting the client to a web proxy server based on the user identification requirements." Mahadik, Fig. 8.

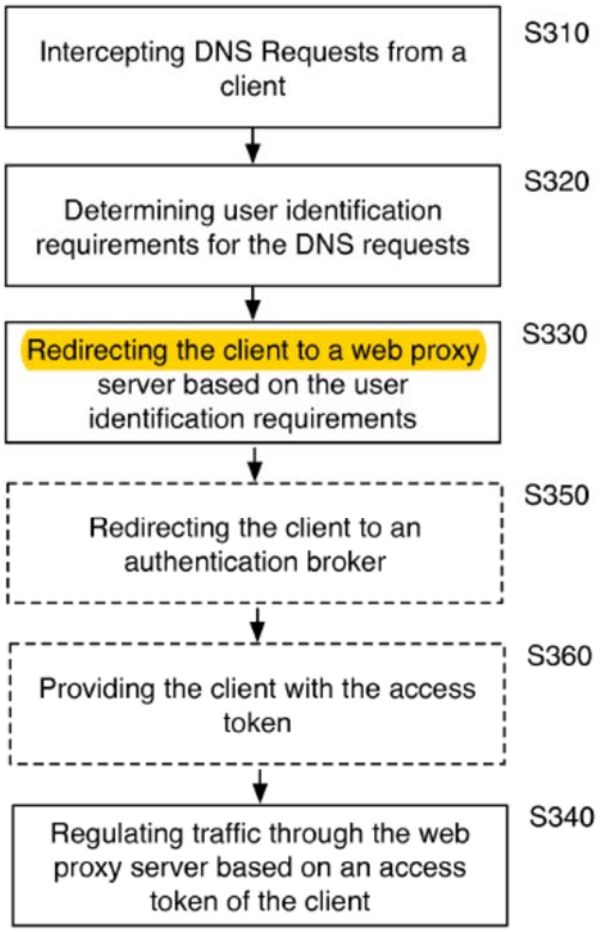


FIGURE 8

Mahadik discloses that the proxy "functions to provide a form of traffic monitoring for resources not fully trusted." Mahadik, [0017]. It is further configured to process the traffic such that suitable actions would be taken. *Id.*

VIII. OVERVIEW OF THE ’856 PATENT

A. Patent Overview

77. The title of the ’856 patent is “Rule-Based Network-Threat Detection for Encrypted Communications.” ’856, *Title.* It is directed to a “packet-filtering system configured to filter packets in accordance with packet-filtering rules may receive data indicating network-threat indicators and may configure the packet-filtering rules to cause the packet-filtering system to identify packets comprising unencrypted data, and packets comprising encrypted data. A portion of the unencrypted data may correspond to one or more of the network-threat indicators, and the packet-filtering rules may be configured to cause the packet-filtering system to determine, based on the portion of the unencrypted data, that the packets comprising encrypted data correspond to the one or more network-threat indicators.” ’856, *Abstract*, 1:32-45. Figure 7 illustrates this method for rule-based network threat detection:

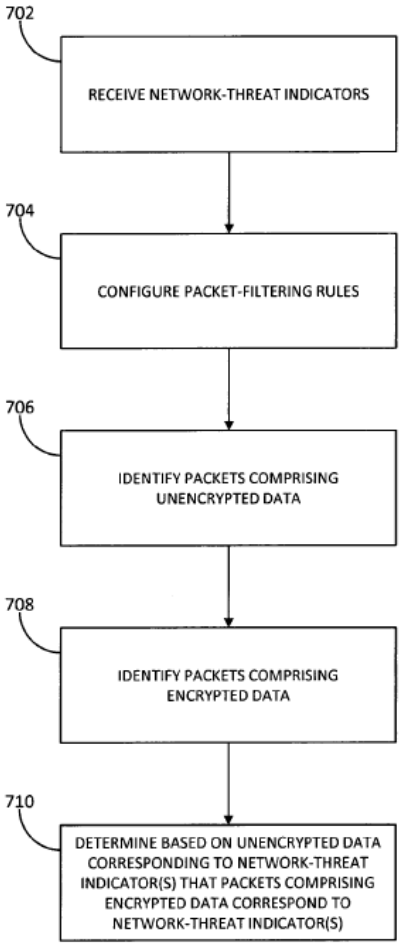


FIG. 7

’856, 23:63-65. Referring to step 702 of Fig. 7, the packet-filtering system “may receive data indicating network-threat indicators,” such as “network addresses, domain names, uniform resource identifiers (URIs), and the like.” ’856, 1:11-15. 3:67-4:3. This data received by the system are rules from “rule providers 138 based on network-threat indicators provided by threat-intelligence providers 140.” ’856, 23:66-24:4, 4:3-7. In step 704, the packet-filtering system configures rules to

identify packets corresponding to the network-threat indicators. ’856, 24:4-7, 4:3-7, 4:18-21.

78. Again referring to Figure 7, “[i]n step 706, the packet-filtering system may identify packets comprising unencrypted data. For example, packet-filtering system 200 may identify packets comprising a DNS query, a reply to a DNS query, or a handshake message configured to establish an encrypted communication session.” ’856, 24:8-12, 9:19-28. Further, “[i]n step 708, the packet-filtering system may identify packets comprising encrypted data. For example, packet-filtering system 200 may identify packets encrypted in accordance with one or more parameters of sessions…” ’856, 24:14-17, 6:42-45. “In step 710, the packet-filtering system may determine based on a portion of the unencrypted data corresponding to the network-threat indicators that the packets comprising encrypted data correspond to the network-threat indicators. For example, packet-filtering system 200 may determine that a domain name included in the DNS query, the reply to the DNS query, or the handshake message corresponds to the network-threat indicators, and packet-filtering system 200 may determine that one or more of the packets encrypted in accordance with the parameters of sessions 306, 308, 408, 410, 504, or 606 correlate to one or more packets comprising the DNS query, the reply to the DNS query, or the one or more handshake messages.” ’856, 24:18-30, 9:29-33. The

filtered packets are routed to a proxy system based on a determination that the packets comprise data corresponding to one or more of the network-threat indicators. ’856, 6:45-59. The proxy system may decrypt encrypted data, which then may be filtered using the unencrypted data in the packets, e.g., URI, data indicating a protocol version, request, or command, before the packets are re-encrypted. ’856, 10:56-11:40.

79. Representative Claim 1 of the ’856 patent recites:

A method comprising:

receiving, by a packet-filtering system comprising a hardware processor and a memory and configured to filter packets in accordance with a plurality of packet-filtering rules, data indicating a plurality of network-threat indicators, wherein at least one of the plurality of network-threat indicators comprises a domain name identified as a network threat;

identifying packets comprising unencrypted data;

identifying packets comprising encrypted data;

determining, by the packet-filtering system and based on a portion of the unencrypted data corresponding to one or more network-threat indicators of the plurality of network-threat indicators, packets comprising encrypted data that corresponds to the one or more network-threat indicators;

filtering, by the packet-filtering system and based on at least one of a uniform resource identifier (URI) specified by the plurality of packet-filtering rules, data indicating a protocol version specified by the plurality of packet-filtering rules, data indicating a method specified by the plurality of packet-filtering rules, data indicating a request specified by the plurality of packet-filtering rules, or data indicating a command specified by the plurality of packet-filtering rules:

> packets comprising the portion of the unencrypted data that corresponds to one or more network-threat indicators of the plurality of network-threat indicators; and

> the determined packets comprising the encrypted data that corresponds to the one or more network-threat indicators; and

routing, by the packet-filtering system, filtered packets to a proxy system based on a determination that the filtered packets comprise data that corresponds to the one or more network-threat indicators.

B. Prosecution History

80. The ’856 patent was matured from a U.S. Patent Application No. 14/757,638, which was filed on December 23, 2015.

81. During prosecution and on April 12, 2017, Examiner issued a non-final rejection on the pending claims under §101 and §112, and further under §103 as obvious in view of Spies (U.S. Patent Publication 2005/0138353), Sorensen (U.S. Patent Publication 2012/0023576), and in further view of Bomer (U.S. Patent Publication 2004/0199629). EX1002, 1064-1083. In response and on October 11, 2017, Applicant amended the claims and argued that Spies does not teach the determining step of claim 1. EX1002, 1365. Applicant argued that Spies teaches filtering unencrypted data and processing decrypted data, and does not teach “any determination regarding any encrypted data based on any unencrypted data corresponding to one or more network-threat indicators.” *Id.* In addition, Applicant argued that Spies fails to teach network-threat indicators comprise a domain name as recited in the claims because, according to Applicant, Spies teaches a domain

name but that domain name is not identified as a network threat. *Id.* Lastly, Applicant argued that neither Sorenson nor Bomer remedy the defects of Spies. EX1002, 1351-1367.

82. Examiner allowed the claims in a Notice of Allowance issued on December 20, 2017, indicating that the amended claims were sufficient to overcome the prior art of record. EX1002, 1374-1390. On January 23, 2018, Applicant requested additional amendments to the claims prior to issuance. EX1002, 1413-1422. Further on February 5, 2018, the amendment was entered. EX1002, 145-1450. The ’856 issued on 03/13/2018. EX1002, 1453.

IX. CLAIM INTERPRETATION

83. I have been informed that only terms necessary to resolve the controversy need to be construed. For the purposes of the present declaration, I have applied the legal principles outlined above in Section IV.B and propose construction for one claim term, as indicated below.

84. I understanding that the following term is being construed in this proceeding:

a. “network-threat indicators”

85. Independent claims 1, 24 and 25 of the ’856 patent recite “receiv[e/ing . . .] data indicating a plurality of network-threat indicators.” ’856, cls. 1, 24-25. I

understand the term "network-threat indicator" has been interpreted to mean "indicators that represent the identity of a resource associated with a network threat." I have reviewed the patent specification and prosecution history. I agree with this interpretation.

86. The '856 patent specification provides that "[n]etwork-threat services provide information associated with network threats, for example, reports that include listings of network-threat indicators (e.g., network addresses, domain names, uniform resource identifiers (URIs), and the like). '856, 1:11-15. Each of those example indicators in parentheses represents the identity of a resource. Similarly, the '856 patent specification provides that "[t]he threat-intelligence reports may include one or more network-threat indicators, for example, domain names (e.g., fully qualified domain names (FQDNs)), URIs, network addresses, or the like." '856, 3:65-4:4.

87. I understand that this construction was adopted for this same term in IPRs filed against other of patent owner's patents in which this same claim term appears.

X. GROUND 1: CLAIMS 1, 24-25 ARE RENDERED OBVIOUS BY BURUGANAHALLI

88. The '856 patent has three independent claims. Claim 1 is a method claim; Claim 24 is a system claim; and Claim 25 is a non-transitory computer-

readable medium claim. Other than the differences based on the types of claims, the recited limitations in the three claims are largely identical and substantively similar. Claim 1 is representative of Claims 24 and 25 and is analyzed below.

A. Overview of Buruganahalli

89. Buruganahalli (Ex. 1004) is generally directed to firewall technologies. Buruganahalli explains that "[a] firewall generally protects networks from unauthorized access while permitting authorized communications to pass through the firewall." Buruganahalli, 1:20-22. Buruganahalli discloses that there are many ways to implement a firewall, for example as a device, a set of devices or software executed on a device that provides a firewall function for network access; and there are also a number of ways to implement Buruganahalli's disclosed techniques, including as a process, apparatus, a computer program and/or a processor. Buruganahalli, 1:21-31, 2:14-32, 2:48-60. Furthermore, Buruganahalli's firewall may be implemented in a security device that includes many functions, such as a proxy function (which is a type of security function) and a routing function. *Id.*, 1:36-38, 3:5-15. Buruganahalli further discloses that the "routing functions can be based on source information (*e.g.*, IP address and port), destination information (*e.g.*, IP address and port), and protocol information." *Id.*

90. Buruganahalli discloses "a firewall can filter inbound traffic by applying a set of rules or policies to prevent unwanted outside traffic from reaching

protected devices." Buruganahalli, 2:61-66, 1:32-36. These rules specify certain matching criteria or heuristics, such as domain (*e.g.*, destination domain). Buruganahalli, 14:60-67, 4:23-30, 5:35-43. As an example, a security policy may be a "whitelist/blacklist policy" where a destination domains may be listed on a "blacklist" such that destination domains extracted from packets would be compared against the domains listed on the blacklist. Buruganahalli, 7:39-67, 10:66-11:2. The firewall then filters packets of a session between a client and a remote server associated with the domain name. Buruganahalli, 7:39-51, 7:60-67. Also, there are several ways to identify a domain, such as using a URL or an IP address. Buruganahalli, 3:5-24, 5:6-23, 7:39-8:44, 9:25-28, 10:21-28, 13:52-14:1, 15:37-45.

91. Figure 3A illustrates a firewall that filters packets in communications between a client and a remote server. More specifically, it illustrates an SSL/TLS session passing through a firewall between the client and the remote server. Buruganahalli, 4:31-40.

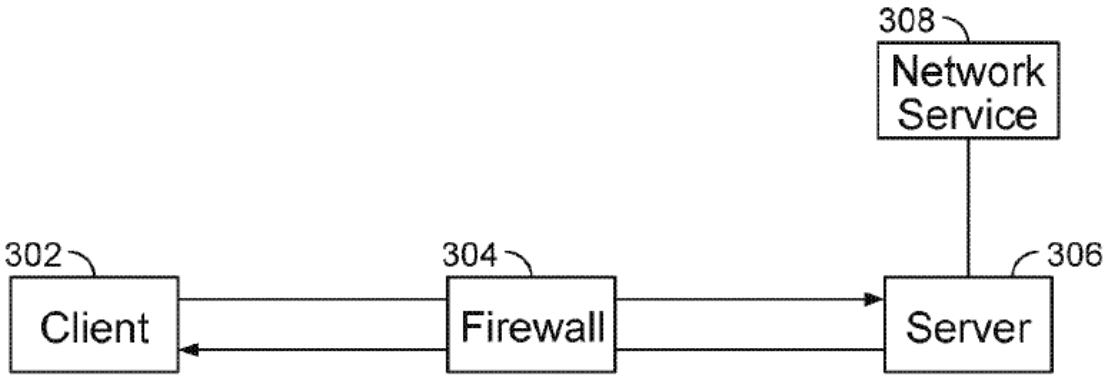


FIG. 3A

In the SSL/TLS session, "a typical firewall can initially just inspect the handshake traffic (*e.g.*, the initial negotiation to setup a secure communication channel/tunnel between a client and a remote server), as all the application data is encrypted." Buruganahalli, 10:54-61, 11:7-42. In Buruganahalli's system, the firewall extracts information, such as a destination domain, in unencrypted packets from the handshake message. Buruganahalli, 5:24-51, 7:24-25, 11:7-13. It then applies rules to the extracted information, such as determining whether it corresponds to the domains listed on a blacklist. Buruganahalli, 5:24-36, 7:42-67, 11:7-19, 11:30-54. The firewall then filters the unencrypted packets from the handshake message and may take additional actions (*e.g.*, decryption) that apply to the encrypted packets in the secure session based on the determination whether they correspond to the blacklist. *Id.*, *Abstract*, 4:16-22, 4:31-44, 6:12-39, 6:62-7:9. The firewall may

intercept a client's initial request for an encrypted session. *Id.*, 15:4-21, Fig. 7. One example of an initial request is a TLS "hello message sent from the client to the remote server." *Id.* The firewall then extracts a domain name from the unencrypted request. *Id.* Based on the information in the initial request, the firewall applies rules by which the packets of the session between the client and the remote server are filtered. *Id.*

92. Buruganahalli discloses that the firewall uses a decrypt engine to decrypt encrypted traffic in a secure session associated with the handshake message. Buruganahalli, 7:42-51, 7:60-67, 8:45-54, 9:61-10:9. Figure 3B depicts establishing an SSL session between a client and a server and the corresponding description explains the decryption of encrypted packets in accordance with this process. Buruganahalli, 9:61-10:9, 12:4-18. In response to intercepting and detecting a request to establish a secure session, Figure 3B shows that the firewall splits the SSL session between the client and the remote server into two half sessions as "Session A" and "Session B". Buruganahalli, 12:12-18.

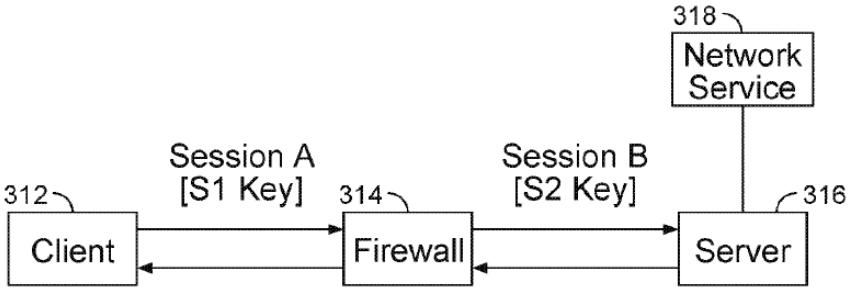


FIG. 3B

Specifically, Buruganahalli discloses, "[i]n Session A, the firewall 314 acts as the remote server 316 such that it is transparent to the client 312 that it is not communicating directly with the remote server 316. Session A traffic is encrypted using the session key S1 associated with the firewall device. In Session B, the firewall 314 acts as the client 312 such that it is transparent to the remote server 316 that it is not communicating directly with the client 312. Session B traffic is encrypted using the session key S2 associated with the firewall device (e.g., the firewall device can store the fingerprint from the remote server in association with that remote server IP address)." Buruganahalli, 12:14-28. Accordingly, the firewall uses "S1 key" to decrypt data communicated from the client to the firewall, and uses "S2 key" to decrypt data communicated from the server to the firewall. Buruganahalli, 12:29-32, 12:52-55. In this manner, the firewall processes the packets "transparent" to the client and the server (that are the end points of the

communication) such that neither the client or server would detect the presence of the firewall that applies that trusted man-in-the-middle techniques for decryption.

93. Buruganahalli uses a "self-signed certificate" in applying "trusted man-in-the-middle" techniques. Buruganahalli, 9:61-67. This means that the firewall replaces the public key in the original server certificate with its own key, and then generates a new certificate signed with its own public key, *i.e.*, "S1 key," to replace the server certificate. Buruganahalli, 5:6-23, 5:52-61, 8:25-32, 9:61-67, 12:18-36. Buruganahalli further discloses that "[t]hese decryption techniques can be used in addition to and/or in combination with the various techniques described herein for destination domain extraction," which includes the example provided in Figure 3A above. *Id.*, 11:59-12:3.

94. In Buruganahalli's system, a firewall intercepts and inspects packets sent from and to a client. Buruganahalli, 12:4-55. In accordance with one or more rules implementing a blacklist, the firewall identifies a request as part of a session-initiating handshake from the client to establish a secure session with a destination domain. Buruganahalli, 12:4-55, 7:39-67. As described in connection with Figure 3B, the firewall filters the unencrypted packets establishing the session, as well as the subsequent encrypted packets of the secure session in accordance with the one or more rules. Buruganahalli, 12:4-55. The firewall determines that the encrypted

packets are part of the session established by the client request using unencrypted information, such as IP addresses in the packets. *Id.* at 9:25-28. It was well-known and conventional to use IP addresses and other unencrypted header data to determine if packets belong to a session. For example, Buruganahalli explains to use "IP address… for a monitored traffic flow (*e.g.*, a session) based on packet analysis." Buruganahalli, 9:25-28. IP addresses generally reside in the unencrypted portion of a packet and can be analyzed without first decrypting the packet. Therefore it would be routine to use IP addresses to determine if a packet is associated with a session.

95. Returning to Figure 3B, the firewall passes the client's secure session request to the server. Buruganahalli, 12:4-55. The server responds, and the firewall intercepts the response, determines that it is a response to the request, and then decodes and filters it. *Id.* The firewall extracts a key from the server's response and substitutes it with its own key. *Id.* The firewall passes the response, along with the substituted key, to the client. *Id.* The key substitution effectively splits the original session into two half sessions: Session A and Session B. In Session A, the client encrypts packets using the substituted key (S1 key), and the firewall filters and then decrypts the encrypted packets from the client (having identified those packets as being part of the secure session) using that key. *Id.* And in Session B, the firewall re-encrypts the same information using the server's key (S2 key), and sends the

encrypted packets to the server. *Id.* at 12:4-55. Similarly, when information is sent from the server back to the client, a reverse process of filtering and decryption/re-encryption takes place. *Id.*

96. Buruganahalli describes this process of session decryption and encryption using an intermediary interface as "transparent" to the client and server (which are end points of the communications). This description essentially describes the functions of a proxy, and more specifically, a "transparent" proxy, which was well-known in the art. For example, Wang, discloses, "because the proxy device 120 is a transparent intermediary to the client-server security session, the proxy device is able to decrypt the received traffic/packets (*e.g.*, using the session key negotiated for the client-proxy or proxy-server session, as appropriate) from each sending end device, and may re-encrypt the traffic prior to forwarding it on using the respective session key with the receiving end device." Wang, 8:12-19. As another example, Ebrahimi discloses that "a transparent proxy is neither known to nor configured by the clients; rather, the transparent proxy intercepts outgoing client requests to access origin servers/sites and transparently processes the requests on behalf of the clients." Ebrahimi, 1:48-52. *See also*, NIST, 14. It was also well-known that proxies would operate in a manner that is invisible to an outsider, just as described as "transparent" in connection with Figure 3B of Buruganahalli. *See, e.g.*,

Baehr, 4:50-63, 8:36-47, 8:65-9:7. Accordingly, although the description of Fig. 3B in Buruganahalli does not explicitly refer to implementing its decryption and encryption functionality as a proxy, a person of ordinary skill would have recognized the intermediary nature of that functionality (*i.e.*, implemented in a firewall that is in line between the client and the server), and that functionality's transparent operation relative to communicating endpoints, as a proxy. This is in further view of Buruganahalli's disclosure stating that "security devices were known to include both firewall and proxy functions." Buruganahalli, 3:5-8. The prior art also recognizes that a software application running on a firewall is "a proxy agent." NIST, 52.

97. Buruganahalli may not explicitly disclose "routing" filtered packets (*i.e.* encrypted packets of a secure session directed to a blacklisted domain after having been filtered as being part of the secure session established by the initial client request) to a proxy system, but a person of ordinary skill would have understood that the filtered encrypted packets are "routed" to Buruganahalli's decrypt engine. As I have mentioned above, a person of ordinary skill would have understood that decrypt engine would be implemented as performing the functions of a proxy. This is because the function of a decrypt engine is decryption and Buruganahalli discloses the packets corresponding to a blacklist would be decrypted.

Buruganahalli, 7:39-51, 9:61-67. Buruganahalli also explains that a proxy is a well-known component of a security device. Buruganahalli, 3:5-15. A person of ordinary skill would have understood that transmitting packets between interfaces of functional components of a security device such as a firewall, using a software routing function, was also well-known. For example, NIST discloses that "[a]pplication-proxy gateway firewalls do not require a Layer 3 (Network Layer) route between inside and outside interfaces of the firewall; the firewall software performs the routing." NIST, 12. NIST also discloses, "[a] firewall ruleset is a table of instructions that the firewall uses for determining how packets should be routed between its interfaces"; and "[a] proxy agent is a software application running on a firewall or on a dedicated proxy server that is capable of filtering a protocol and routing it [*sic*] between the interfaces of the device." NIST, 51-52.

98. Alternatively, a person of ordinary skill would have at least found it obvious to route the filtered encrypted packets in Buruganahalli to a proxy because proxies were known and routine components of firewalls and other network security devices. This is explicitly disclosed in Buruganahalli: "[s]ecurity devices… can include various security functions (*e.g.*, firewall, … proxy…)… routing functions can be based on source information… destination information…" Buruganahalli, 3:5-15. Proxies were also known to be used for SSL/TLS decryption. *See, e.g.*,

Wang, 8:6-27, 3:44-57, 4:21-28, 9:30-35. And as I have explained in Section VII.D.6, it was also known to route filtered packets to a proxy implemented internally to, or implemented externally from, the packet filtering device. The two configurations were disclosed extensively in prior art. *See, e.g.*, NIST, 12-14; Baehr, 2:25-30, 5:2-8, 8:12-18, Figs. 6-7; Wang, Fig. 1, 2:32-58; Le Pennec, Fig. 3, [0021]-[0023].

99. A person of ordinary skill would have been further motivated to route filtered packets to a proxy system in implementing the split session disclosed by Buruganahalli for security purposes. It was well known in the art is to isolate potentially dangerous traffic using a proxy, which is isolated from a network, to perform inspection, filtering, logging and/or virus detection. For example, NIST describes dedicated proxy servers as providing filtering, logging and virus detection. NIST, 14. The advantage of a proxy is that it performs these important security functions without the need to alter the network, as was known in the art. Baehr, 2:38-42, 8:13-17, 8:37-47; Wang, 7:53-57, 8:12-21, 8:67-9:4 (no special client/server provisioning required, and decryption and inspection is isolated to the proxy device).

100. Furthermore, a person of ordinary skill would be motivated to perform the session splitting operation by routing filtered packets to a proxy system for

performance reasons. The use of dedicated proxy servers improves the overall performance of a system by handing off decryption and encryption that are known to be computationally intensive processing tasks. For example, NIST discloses that dedicated proxy servers were used to decrease the work load of a firewall. NIST, 14. Indeed, Buruganahalli recognizes the need for "workload balancing of network related resources." Buruganahalli, 3:5-15. Buruganahalli further discloses using "special purpose hardware" to implement firewalls "as dedicated appliances" that would "generally provide higher performance levels for application inspection than software executed on general purpose hardware." Buruganahalli, 4:7-15. Buruganahalli also discloses implementing firewalls as a set of devices such that the proxy would be implemented as its own device. 1:22-25. Accordingly, a person of ordinary skill would have been motivated to use dedicated proxy servers for session splitting to reduce load on a firewall and improve performance.

B. Challenged Claims

1. Preamble[1]

[1] To the extent they are limiting, the preambles of all Challenged Claims are nonetheless met by the art of record.

(a) [1.pre] A method comprising:

(b) [24.pre] A packet-filtering system comprising:

(c) [25.pre] One or more non-transitory computer-readable media comprising instructions that when executed by at least one hardware processor of a packet-filtering system cause the packet-filtering system to:[2]

101. **Regardless of whether the preamble is limiting, Buruganahalli discloses a method.**

102. Buruganahalli discloses "a firewall… [that] protects networks from unauthorized access while permitting authorized communications to pass through the firewall." Buruganahalli, 1:20-22.

103. Buruganahalli further discloses "policy based filtering on… communication flows (e.g., sessions) prior to the set-up of the encrypted data communication," such as "to monitor the initial unencrypted/clear text data communications exchanged as part of an initial handshake to setup a secure connection for a new session between a client and a remote server…" to determine if the establishment of the new session should be allowed for transmitting encrypted packets. Buruganahalli, 4:34-40.

[2] Please see limitation [1.a] for additional mapping.

104. Buruganahalli's techniques may be implemented in many ways. For example, Buruganahalli discloses that "a firewall is typically a device, such as a computer… [and] can also be… executed as software on computer servers, gateways, network/routing devices (e.g., network routers), or data appliances…" 1:22-30. The firewall "can be implemented…as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a process or…[which] may be referred to as techniques." Buruganahalli, 2:14-22. Buruganahalli further discloses that the firewall is "a device, a set of devices, or software executed on a device that provides a firewall function for network access." Buruganahalli, 2:50-52.

2. Limitation a

(a) [1.a] receiving, by a packet-filtering system comprising a hardware processor and a memory and configured to filter packets in accordance with a plurality of packet-filtering rules, data indicating a plurality of network-threat indicators, wherein at least one of the plurality of network-threat indicators comprises a domain name identified as a network threat;

(b) [24.a] at least one hardware processor; and memory storing instructions that when executed by the at least one hardware processor cause the packet-filtering system to: receive data indicating a plurality of network-threat indicators, wherein at least one of the plurality of network-threat indicators comprise a domain name identified as a network threat;

(c) [25.a] receive data indicating a plurality of network-threat indicators, wherein at least one of the plurality of network-threat indicators comprise a domain name identified as a network threat;

105. **Buruganahalli discloses receiving, by a packet-filtering system** (*e.g.,* "packet-filtering firewall," "a security device") **comprising a hardware processor** (*e.g.,* a processor) **and a memory** (*e.g.,* a memory) **and configured to filter packets in accordance with a plurality of packet-filtering rules** (*e.g.,* "set of rules… referred to as policies," "firewall policies/rules"), **data indicating a plurality of network-threat indicators** (*e.g.,* "blacklist," data that identify "threats" or "vulnerabilities"), **wherein at least one of the plurality of network-threat indicators comprises a domain name** (*e.g.,* "domain name," "destination domain") **identified as a network threat.**

106. Buruganahalli discloses that the techniques of its firewall or security device "can be implemented in numerous ways…embodied on…a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor." Buruganahalli, 2:14-31, 2:55-60, 3:16-24, 4:23-30.

107. Buruganahalli discloses, "firewalls….deny or permit network transmission based on a set of rules. These sets of rules are often referred to as policies (e.g., network policies or network security policies)." Buruganahalli, 2:61-64. Furthermore, a firewall "can filter inbound traffic by applying a set of rules or policies to prevent unwanted outside traffic from reaching protected devices"; and it "can also filter outbound traffic by applying a set of rules or policies…" Buruganahalli, 2:64-3:4. These "policies are received and stored in the management plane [of the security device]," as shown in Figure 6. Buruganahalli, 14:59-60.

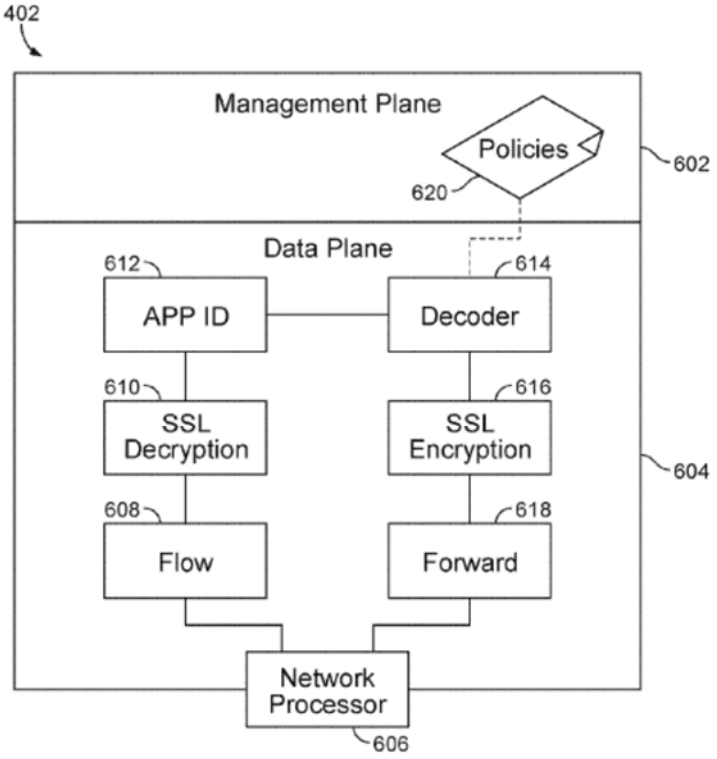


FIG. 6

Buruganahalli, Fig 6.

108. Buruganahalli further discloses that "policies can include one or more rules, which can be specified using domain and/or host/server names, and rules can apply one or more signatures or other matching criteria or heuristics…" Buruganahalli, 14:61-64. A firewall may "apply[] a policy (e.g., a security policy) based on the destination domain to filter traffic at a security device." Buruganahalli, 7:27-29, 8:1-44, 10:66-11:2. For example, Buruganahalli discloses "blocking the session traffic using the security device is performed if a violation of a first firewall policy is determined, in which the first firewall policy includes a policy for verifying

that the destination domain name extracted from a server name indication (SNI) matches…" Buruganahalli, 15:38-42, 5:37-42 (disclosing that the "domain name can be extracted by parsing the client hello message to extract the host name…which can then be used to apply firewall policies or take responsive actions based on this information without having to wait for data to be decrypted…"). Buruganahalli emphasizes that "it is generally desirable for a security device…to extract information about the destination domain for any given new flow (e.g., new session) as early as possible…information about the destination domain can be critical as well as relevant for security processing based on which policies and a set of actions that can be applied to the flow." Buruganahalli, 4:23-30.

109. Buruganahalli discloses that "the security policy can include…a whitelist/blacklist policy" and "the destination domain [can be] included in [a] blacklist…" Buruganahalli, 7:42-43, 7:49-50. Buruganahalli further discloses that "the security device 402 can extract destination domain information… that can facilitate identification of new, zero-day threats, new vulnerabilities, prevent false positives, and/or provide a feedback loop for any of such activities or trends aggregated and correlated." Buruganahalli, 13:44-51. Taken together, Buruganahalli discloses packet-filtering rules (*e.g.*, policies or rules) that are based

on whether certain data (*e.g.*, a destination domain) is identified as a network threat (*e.g.*, by matching the data on a blacklist).

3. Limitation b

(a) [1.b] identifying packets comprising unencrypted data;

(b) [24.b] identify packets comprising unencrypted data;

(c) [25.b] identify packets comprising unencrypted data;

110. **Buruganahalli discloses identifying** (*e.g.*, "identifies") **packets comprising unencrypted data** (*e.g.*, "unencrypted/clear text data communications exchanged as part of an initial handshake," "handshake traffic," "handshake exchange," "hello message").

111. Buruganahalli discloses a security device having a "Flow 608" that "identifies packets as being part of a new session and creates a new session flow. Subsequent packets will be identified as belonging to the session based on a flow lookup." Buruganahalli, 14:25-28. Buruganahalli further discloses that "an initial TLS handshake exchange of that flow can be parsed to extract a hostname to facilitate destination domain extraction for secure protocols…" Buruganahalli, 14:28-31. Buruganahalli further discloses "monitor[ing] the initial unencrypted/clear text data communications exchanged as part of an initial

handshake to setup a secure connection for a new session between a client and a remote server…" Buruganahalli, 4:37-40. Therefore, the system is able to "intercept[] a request to establish an encrypted session from the client to the remote server..." and "extract the hostname from the client hello message in order to apply various policies based on the destination domain." Buruganahalli, 8:26-28, 11:49-51; *see also id.*, 11:59-12:64, 15:22-26.

112. Buruganahalli discloses that a "firewall can…inspect the handshake traffic (e.g., the initial negotiation to setup a secure communication channel/tunnel between a client and a remote server)…" Buruganahalli, 10:55-60. This is discussed in connection with Figure 3A.

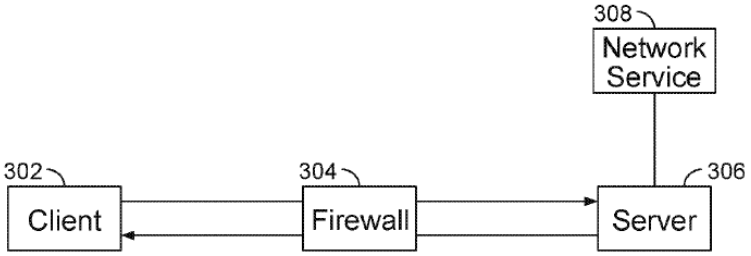


FIG. 3A

Buruganahalli also discloses "parsing the handshake traffic (e.g., parsing a client hello message to extract a hostname that identifies the destination domain…)". Buruganahalli, 11:9-11; *see also id.*, 12:32-36.

113. Buruganahalli discloses that the "handshake exchange (e.g., handshake/negotiation)" contains and transmits unencrypted packet data including "the client hello message is an unencrypted communication." Buruganahalli, 7:24-25, 11:31-41. This allows early extraction of relevant information by "appl[ying] sooner based on this early extraction of the hostname…such that the rest of the stream can be forwarded without any decryption resulting in….enhanc[ed] performance (e.g., of the security device, such as a firewall)." Buruganahalli, 5:47-51. Therefore, to set up a new session, the system identifies unencrypted packet data within the initial handshake.

4. Limitation c

(a) [1.c] identifying packets comprising encrypted data;

(b) [24.c] identify packets comprising encrypted data;

(c) [25.c] identify packets comprising encrypted data;

114. **Buruganahalli discloses identifying** (*e.g.,* "identifies") **packets comprising encrypted data** (*e.g.,* "encrypted data communication").

115. Buruganahalli discloses a security device having a "Flow 608" that "identifies packets as being part of a new session and creates a new session flow. Subsequent packets will be identified as belonging to the session based on a flow lookup." Buruganahalli, 14:25-28. Buruganahalli further discloses that "[v]arious

techniques…can be used to determine whether a new session…violates a policy… if a new flow is determined to violate a policy…the flow can be blocked…" Buruganahalli, 5:62-6:1. The new session using a secure protocol is an encrypted data communication; furthermore, the encrypted data communication would be evaluated based on handshake traffic configured to set up the session. Buruganahalli, 5:62-6:22, 4:31-44, 5:43-52, 12:33-48. For example, Buruganahalli discloses "monitor[ing] the initial unencrypted/clear text data communications exchanged as part of an initial handshake… prior to the establishment of the secure communication channel/tunnel after which… would be encrypted." Buruganahalli, 4:37-44. A person of ordinary skill would have recognized that the encrypted data communication comprises data packets. Buruganahalli, 3:16-24, 3:40-50, 8:25-27. Specifically, Buruganahalli discloses "various techniques for destination domain extraction for secure protocols (e.g., by ***facilitating deep packet inspection of the session traffic, which can include…encrypted data communications*** associated with the session using a secure protocol, such as SSL/TLS)." Buruganahalli, 8:55-60. This is accomplished by "monitored traffic flow…based on packet analysis..." (Buruganahalli, 9:27-28) and "monitor[ing] encrypted data communications associated with the session." Buruganahalli, 6:15-16, 8:11-44 (referring to a "session" as "encrypted data communications."). Therefore, Buruganahalli teaches

identifying data packets comprising encrypted data during set up of a new session, and during the session.

5. Limitation d

(a) [1.d] determining, by the packet-filtering system and based on a portion of the unencrypted data corresponding to one or more network-threat indicators of the plurality of network-threat indicators, packets comprising encrypted data that corresponds to the one or more network-threat indicators;

(b) [24.d] determine, based on a portion of the unencrypted data corresponding to one or more network-threat indicators of the plurality of network-threat indicators, packets comprising encrypted data that corresponds to the one or more network-threat indicators;

(c) [25.d] determine, based on a portion of the unencrypted data corresponding to one or more network-threat indicators of the plurality of network-threat indicators, packets comprising encrypted data that corresponds to the one or more network-threat indicators;

116. **Buruganahalli discloses determining, by the packet-filtering system and based on a portion of the unencrypted data** (*e.g.*, "unencrypted/clear text data communications exchanged as part of an initial handshake," "handshake traffic," "handshake exchange," "hello message") **corresponding to one or more network-threat indicators of the plurality of network-threat indicators** (*e.g.*, destination domain on a blacklist), **packets comprising encrypted data** (*e.g.*,

"encrypted data communication") **that corresponds to the one or more network-threat indicators** (*e.g.*, destination domain on a blacklist).

117. Buruganahalli discloses that "a firewall can intercept and monitor data communications from a client and a remote server in order to extract the hostname data from the SNI field…which is sent as an unencrypted data communication from the client…at the start of the handshaking process for setting up a secure TLS communication channel/session…" Buruganahalli, 5:27-34. Buruganahalli further discloses "various techniques…for destination domain extraction for secure protocols…by parsing the handshake traffic (e.g., parsing a client hello message to extract a hostname that identifies the destination domain being requested by the client to the remote server)" Buruganahalli, 11:7-12. Buruganahalli discloses "a TLS handshake exchange (e.g., handshake/negotiation) between a client (C) and a remote server (s) using the SNI approach:

> Step 1. C: (TLS Handshake) Hello, I support XYZ Encryption, and I am trying to connect to 'website.example-.com'.
> Step 2. S: (TLS Handshake) Hi there, here is my Public Certificate, and let's use this encryption algorithm.
> Step 3. C: (TLS Handshake) Agreed.
> Step 4. C: (Encrypted) HTTP Request.
> Step 5. S: (Encrypted) HTTP Reply.

... the firewall can intercept and decode (*e.g.*, parse) this TLS handshake exchange to extract the hostname from the client hello message in order to apply various policies based on destination domain." Buruganahalli, 11:31-51.

118. Buruganahalli discloses applying a security policy to compare extracted information from a handshake traffic with information on a blacklist to determine whether the handshake traffic and the encrypted data communication it seeks to set up present any security threats. For example, Buruganahalli discloses "applying a security policy based on the destination domain to filter traffic at the security device, in which the security policy includes a… blacklist policy, and… if the destination domain is included in a blacklist…" Buruganahalli, 7:61-67. Therefore, the firewall inspects the handshake exchange to verify the presence of network-threat indicators by comparing against the blacklist. Buruganahalli further discloses that "the firewall 304 can apply one or more firewall policies/rules… (e.g., a policy that includes requirements or rules related to destination domains that may be used for secure protocol communications)" Buruganahalli, 11:13-17.

119. Buruganahalli correlates packets associated with a session using certain identifying information. For example, Buruganahalli discloses using "[a]n IP address and port engine [] determines [that] an IP address and port number for a monitored traffic flow (e.g., a session) based on packet analysis." Buruganahalli, 9:26-28. It was well-known and conventional that IP addresses for source and destination associated with a packet, or other TCP session data, would be located in an unencrypted portion of a packet (*e.g.,* a clear header) that can be inspected without

decrypting the packet so that the packet can be routed using the IP addresses. Buruganahalli further discloses having "storage stores tables that include host names/identifiers (e.g., destination domains) and associated IP addresses and possibly other information…for implementing policies using destination domain extraction for secure protocols." Buruganahalli, 13:63-14:1.

120. Based on these disclosures, Buruganahalli teaches correlating encrypted packets in a session with the hello message in TLS handshake traffic that initiates the session. A person of ordinary skill would have understood that this correlation is accomplishing using at least address information (*e.g.,* IP address) associated with the destination domain in the hello message. Accordingly, a person of ordinary skill would have understood that determining encrypted packets comprising data corresponding to one or more network-threat indicators (e.g., a blacklisted destination domain) based on data included in the hello message (e.g., destination domain and associated IP address) configured to establish the encrypted session, because both the encrypted session and the hello message would be associated with the same destination IP address of the remote server.

121. Indeed, Buruganahalli's use of a handshake message in performing the determining step is similar to those in the specification of the '856 patent. The '856 patent discusses to "determine that the packets comprise data corresponding to the

network-threat indicators based on data included in the one or more handshake messages configured to establish session" and to "determine that one or more of the packets encrypted… correlate to one or more packets comprising … the one or more handshake messages." '856, 9:30-33, 24:18-30.

6. Limitation e

(a) [1.e] filtering, by the packet-filtering system and based on at least one of a uniform resource identifier (URI) specified by the plurality of packet-filtering rules, data indicating a protocol version specified by the plurality of packet-filtering rules, data indicating a method specified by the plurality of packet-filtering rules, data indicating a request specified by the plurality of packet-filtering rules, or data indicating a command specified by the plurality of packet-filtering rules:

(b) [24.e] filter, based on at least one of a uniform resource identifier (URI) specified by a plurality of packet-filtering rules, data indicating a protocol version specified by the plurality of packet-filtering rules, data indicating a method specified by the plurality of packet-filtering rules, data indicating a request specified by the plurality of packet-filtering rules, or data indicating a command specified by the plurality of packet-filtering rules:

(c) [25.e] filter, by the packet-filtering system and based on at least one of a uniform resource identifier (URI) specified by a plurality of packet-filtering rules indicating one or more of the plurality of network-threat indicators, data indicating a protocol version specified by the plurality of packet-filtering rules, data indicating a method specified by the plurality of packet-filtering rules, data indicating a request specified by the plurality of packet-filtering rules, or data indicating a command specified by the plurality of packet-filtering rules:

122. **Buruganahalli teaches filtering, by the packet-filtering system and based on data indicating a request** (*e.g.*, "a request to create a secure connection") **specified by the plurality of packet-filtering rules.**

123. Please see my discussion above in [1.a]. Buruganahalli discloses that "[a] firewall can filter inbound traffic by applying a set of rules or policies." Buruganahalli, 1:32-35, 2:61-66. Buruganahalli further discloses that "a network service activity that is in violation of one or more firewall policies/rules implemented by the firewall device [] (e.g., a policy that includes requirements or rules related to destination domains…)"). Buruganahalli, 10:63-11:2.

124. Please see my discussion above in [1.d]. Buruganahalli discloses parsing a client's SSL/TLS request, such as packets in a hello message or as part of a handshake traffic, to establish a secure connection with a remote server to determine whether the destination domain extracted from the request corresponds to information on a blacklist. For example, Buruganahalli discloses "destination domain extraction for secure protocols includes… initiating a setup for a secure protocol-based connection… destination domain is extracted from the server name indication (SNI) of a client hello message" and this destination domain extraction approaches further includes "applying a policy (e.g., a security policy)." Buruganahalli, 7:12-28, 5:42-52. Buruganahalli also discloses "applying a security policy based on the destination domain to filter traffic at the security device, in which the security policy includes a whitelist/blacklist policy, and [determines] if the destination domain is included in a blacklist of the whitelist/blacklist policy."

Buruganahalli, 7:63-67; *see also*, 7:39-51. Buruganahalli further discloses "extracting a destination domain from the *request* to create the secure connection" and "applying a policy (e.g., a security policy) based on the destination domain to filter traffic using a security device." 15:8-21, 8:25-44. Taken together, a person of ordinary skill would have understood that Buruganahalli teaches filtering packets based on a SSL/TLS request specified by security rules or policy.

125. **Buruganahalli discloses filtering, by the packet-filtering system and based on a uniform resource identifier (URI)** (*e.g.*, a uniform resource locator (URL)) **specified by the plurality of packet-filtering rules.**

126. Please see my discussion above in [1.a]. Buruganahalli discloses that "[a] firewall can filter inbound traffic by applying a set of rules or policies." Buruganahalli, 1:32-35, 2:61-66, 10:63-11:2.

127. Buruganahalli discloses "applying a security policy based on the destination domain to filter traffic at the security device," such as "a uniform resource locator (URL)/category filtering policy." Buruganahalli, 7:39-7:51, 10:24-25. Buruganahalli also discloses "blocking the session traffic using the security device is performed if a violation…is determined, in which…the destination domain name extracted from a server name indication (SNI) matches the domain associated with a uniform resource locator (URL) for a session associated with the network

communications between the client and the remote server." Buruganahalli, 15:38-45. A person of ordinary skill would have recognized, as it was well-known and conventional in the art, that URLs are examples of uniform resource identifiers. For example, U.S. Patent 8,813,228 ("Magee") (Ex. 1037) discloses that "[t]he module will also gather a taxonomy of the uniform resource identifiers (URIs) (e.g., URLs…) associated with the target host." Magee, 12:26-27. Accordingly, Buruganahalli teaches filtering based on a URI (*e.g.*, URL) specified by a rule. Buruganahalli further discloses the ability to "perform application layer filtering…" (Buruganahalli, 3:25-26) and "real-time content scanning…" (Buruganahalli, 4:2) through "state-based firewall…[to] monitor traffic flows…" (Buruganahalli, 9:18) as well as utilizing "application identification, user identification, and/or other information to match signatures (e.g., file-based, protocol-based, and/or other types/forms of signatures) for detecting malware or suspicious behavior." Buruganahalli, 10:15-20, 9:29. In view of these disclosures, a person of ordinary skill would have further understood that a firewall filters packets by scanning content for the URL in the application layer where a URL would be found.

128. Corresponding limitation [e] of Challenged Claim 25 has an additional limitation reciting "filter[ing], by the packet-filtering system and based on at least one of a uniform resource identifier (URI) specified by a plurality of packet-filtering

rules ***indicating one or more of the plurality of network-threat indicators.***" Please see my discussion above in [1.a]. Buruganahalli teaches packet-filtering rules that indicate network-threat indicators. As an example, Buruganahalli discloses that firewalls "deny or permit network transmission based on a set of rules" and "filter inbound traffic by applying a set of rules or policies to prevent unwanted outside traffic from reaching protected devices" and thus these rules are implemented for security purposes to protect the network. Buruganahalli, 2:61-66, 1:32-36. Buruganahalli further discloses the policies or rules include requirements related to destination domains, which are listed on a blacklist if they are deemed to present a security risk to the system. Buruganahalli, 10:66-11:2, 7:27-29, 7:42-67, 8:1-44, 13:43-52. A person of ordinary skill would have understood that destination domains that are included on a blacklist are network-threat indicators as they present a potential security risk.

129. **Buruganahalli also teaches filtering, by the packet-filtering system and based on data indicating a protocol version** (*e.g.*, "protocol information," version of TLS protocol with "server name indication (SNI)" extension) **specified by the plurality of packet-filtering rules.**

130. Buruganahalli discloses that firewalls "inspect the individual packets themselves and apply rules based on the inspected packets[']…source and

destination address information, ***protocol information***, and a and port number…" Buruganahalli, 3:16-24. Buruganahalli further discloses "monitored network traffic can include… traffic using ***other types of known or unknown protocols…***" Buruganahalli, 9:18-19. A person of ordinary skill would have understood that protocol information would includes a version number that identifies the specific version of the protocol because it was widely known that certain SSL and TLS versions have significant security vulnerabilities and risk assessment would be properly conducted only if the protocol version is known. For example, such security vulnerabilities were described in a Request for Comments for "Prohibiting Secure Sockets Layer (SSL) Version 2.0" (Turner) (Ex. 1031), which lists various deficiencies in SSL version 2.0: "Message authentication uses MD5. Most security-aware users have already moved away from any use of MD5"; "Handshake messages are not protected…"; "Message integrity and message encryption use the same key…"; and "Sessions can be easily terminated…" Turner, 2. Similarly, a "SSL/TLS and PKI History" article (Ristic) (Ex. 1032) also identifies security issues with various versions of SSL and TLS, e.g., "SSL v2 is shot down because of serious security issues…" Ristic, 1. Accordingly, a person of ordinary skill would have been motivated and found it obvious to filter packets based on SSL/TLS versions specified by packet-filtering rules.

131. Buruganahalli further discloses "monitoring network communications…and extracting a destination domain from the request to create the secure connection. …the secure protocol is a secure sockets layer (SSL) protocol or transport layer security (TLS) protocol and the destination domain is extracted from the server name indication (SNI) of a client hello message…" Buruganahalli, 7:13-23, *Abstract*, 4:45-5:4, 15:4-21. A person of ordinary skill would have understood that this refers to a version of the TLS protocol using the SNI extension.

7. Limitation f

(a) [1.f] packets comprising the portion of the unencrypted data that corresponds to one or more network-threat indicators of the plurality of network-threat indicators; and

(b) [24.f] packets comprising the portion of the unencrypted data corresponding to one or more network-threat indicators of the plurality of network-threat indicators; and

(c) [25.f] packets comprising the portion of the unencrypted data corresponding to one or more network-threat indicators of the plurality of network-threat indicators; and

132. **Buruganahalli discloses filtering packets comprising the portion of the unencrypted data** (*e.g.*, "unencrypted/clear text data communications exchanged as part of an initial handshake," "handshake traffic," "handshake exchange," "hello message") **that corresponds to one or more network-threat**

indicators of the plurality of network-threat indicators (*e.g.*, destination domain on a blacklist).

133. Please see my discussion above in [1.b], [1.d] to [1.e]. Buruganahalli discloses that it is "desirable for a security device…to extract information about the destination domain for any given new flow… as early as possible." Buruganahalli, 4:23-26. Buruganahalli further discloses that it is also "desirable and efficient to…facilitate policy-based filtering on such data communication flows (e.g., sessions) prior to the set-up of the encrypted data communication for such flows (e.g., to monitor the initial unencrypted/clear text data communications exchanged as part of an initial handshake to setup a secure connection for a new session between a client and a remote server, which is prior to the establishment of the secure communication channel/tunnel after which any data communications intercepted between the client and the remote server would be encrypted)." Buruganahalli, 4:34-44. Buruganahalli discuss as an example that "intercepting a request to establish an encrypted session from the client to the remote server…" Buruganahalli, 8:26-28, 15:24-25. Furthermore, Buruganahalli discloses "filter[ing] traffic at the security device… if the destination domain is included in a blacklist..." based on inspecting the handshake traffic," which is known to be the unencrypted portion of the encrypted secure tunnel session traffic. Buruganahalli, 7:41-66, 10:54-11:19.

Accordingly, Buruganahalli teaches filtering packets comprising unencrypted data in the handshake traffic that correspond to a blacklisted destination domain.

8. Limitation g

(a) [1.g] the determined packets comprising the encrypted data that corresponds to the one or more network-threat indicators; and

(b) [24.g] the determined packets comprising the encrypted data that corresponds to the one or more network-threat indicators; and

(c) [25.g] the determined packets comprising the encrypted data that corresponds to the one or more network-threat indicators; and

134. **Buruganahalli teaches filtering the determined packets comprising the encrypted data** (*e.g.,* "encrypted data communication") **that corresponds to the one or more network-threat indicators** (*e.g.,* destination domain on a blacklist).

135. Please see my discussion above in [1.c] to [1.e]. Buruganahalli discloses filtering "encrypted data communications" based on a security policy or rule as discussed above. Buruganahalli, 4:20-22. The encrypted data communications are filtered after it is determined that information extracted from the handshake traffic matches with those on a blacklist, and thus corresponds to a network-threat indicator. Buruganahalli, 6:12-39 (encrypted data communications associated with a secure session monitored for filtering based on a firewall policy); *see also id*., Abstract (applying a policy based on the destination domain to filter

traffic), 4:31-44 (policy-based filtering on a session with encrypted data communications), 6:62-7:9 (applying policy to a resumed session), 7:12-29 (applying a policy based on the destination domain to filter traffic), 7:39-51 (same), 7:60-67 (same), 15:4-21 (same). To the extent it is argued that Buruganahalli does not explicitly disclose filtering encrypted data using network-threat indicators, a person of ordinary skill would have understood that packets transmitted as part of a secure session are filtered after determining whether the handshake message that initiates the session corresponds to a network threat indicator, such as a blacklisted destination domain. This is in view of the disclosure that the handshake message and the encrypted data of the session are correlated to the same session using network address information. A person of ordinary skill would have understood that both correspond to the same network-threat indicator because any security risks presented by the handshake message would also apply to the encrypted session based on the correlation. A person of ordinary skill, would at a minimum, would have been motivated and found it obvious to filter the encrypted data of a secure session that was initiated by a handshake message with data corresponding to a network-threat indicator in order to monitor transmission, into or out of a protected network, of data that potentially presents a security risk.

9. Limitation h

(a) [1.h] routing, by the packet-filtering system, filtered packets to a proxy system based on a determination that the filtered packets comprise data that corresponds to the one or more network-threat indicators.

(b) [24.h] route, by the packet-filtering system, filtered packets to a proxy system based on a determination that the filtered packets comprise data that corresponds to the one or more network-threat indicators.

(c) [25.h] route, by the packet-filtering system, filtered packets to a proxy system based on a determination that the filtered packets comprise data that corresponds to the one or more network-threat indicators.

136. **Buruganahalli teaches routing, by the packet-filtering system, filtered packets to a proxy system** (*e.g.*, a "decrypt engine… applying trusted man-in-the middle techniques," a "proxy" that is one of the "security functions") **based on a determination that the filtered packets comprise data that corresponds to the one or more network-threat indicators** (*e.g.*, destination domain on a blacklist).

137. Buruganahalli discloses that "if the destination domain is included in the blacklist" (*i.e.*, if the filtered packets correspond to a network-threat indicator), then the encrypted data communications between the client and the server are "decrypted." Buruganahalli, 7:39-51, 7:60-67. A person of ordinary skill would have understood that the filtered encrypted data are decrypted by the decrypt engine

of the firewall. As shown in Figure 1, Buruganahalli discloses that firewall 100 has a decrypt engine 110 that decrypts monitored traffic. Buruganahalli, 9:61-67.

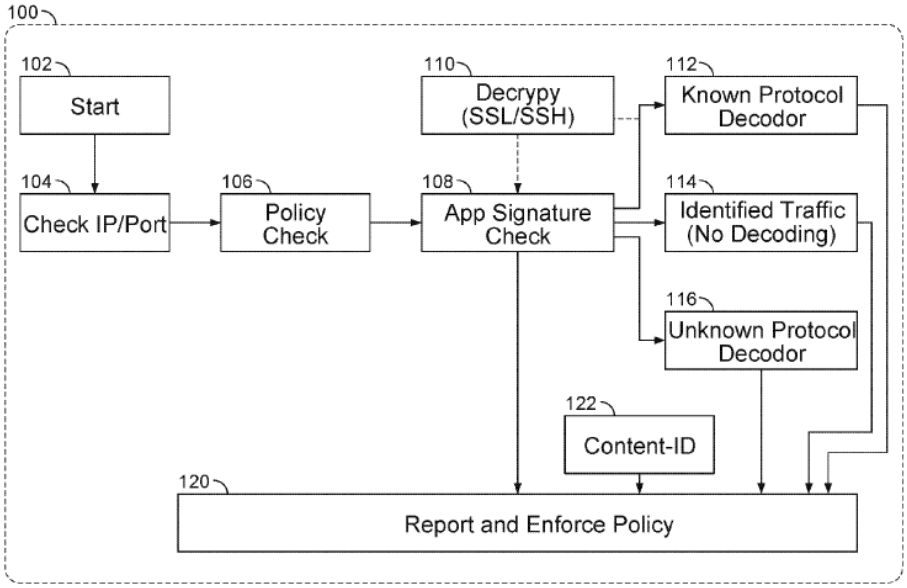


FIG. 1

138. Buruganahalli describes the process of decrypting the encrypted data in more details. Figure 3B of Buruganahalli illustrates a process of establishing an SSL session. Buruganahalli, 12:4-18.

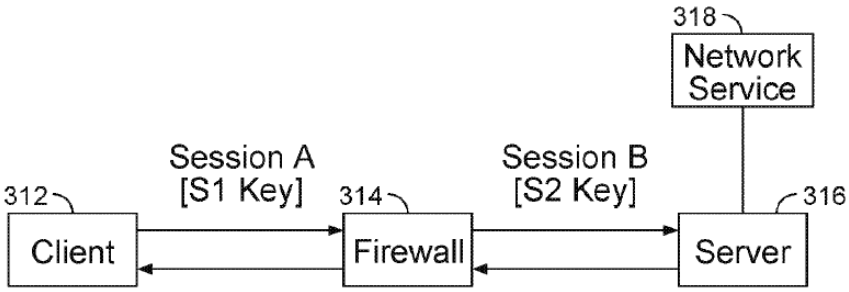


FIG. 3B

Here, a SSL session request from a client to a server is detected and intercepted by a firewall. Buruganahalli, 12:12-14. The session is split in half with Session A between the client and the firewall and Session B between the firewall and the server.

139. When the client makes an SSL session request, Buruganahalli discloses that, "the SSL session request is intercepted and detected by the firewall." The firewall handles the request as follows: "In Session A, the firewall 314 acts as the remote server 316 such that it is transparent to the client 312 that it is not communicating directly with the remote server 316. Session A traffic is encrypted using the session key S1 associated with the firewall device. In Session B, the firewall 314 acts as the client 312 such that it is transparent to the remote server 316 that it is not communicating directly with the client 312. Session B traffic is encrypted using the session key S2 associated with the firewall device…" Buruganahalli, 12:14-28. Buruganahalli describes the packets of a secure session are processed in a manner transparent to the end points (*i.e.*, the client and the server) of the communications such that they would not detect the presence of the firewall applying the "trusted man-in-the-middle technique." Buruganahalli, 12:14-15.

140. Buruganahalli describes the decryption engine of the firewall as applying "trusted man-in-the the-middle techniques using a self-signed certificate." Buruganahalli, 9:61-67. As I have mentioned in the overview section of

Buruganahalli above, Buruganahalli uses a "self-signed certificate" in applying "trusted man-in-the-middle" techniques. Buruganahalli, 9:61-67. A person of ordinary skill would have understood that the firewall replaces the public key in the original server certificate with its own key, and then generates a new certificate signed with its own public key ("S1 key") to replace the server certificate. Buruganahalli, 5:6-23, 5:52-61, 8:25-32, 9:61-67, 12:18-36.

141. Buruganahalli further discloses that "[t]hese decryption techniques can be used in addition to and/or in combination with the various techniques described herein for destination domain extraction," which includes the example provided in Figure 3A above. *Id.*, 11:59-12:3. Accordingly, a person of ordinary skill would have understood that these techniques described in connection with Figure 3A and Figure 3B are taught to be combined in implementing Buruganahalli's security system.

142. Also, as I discussed in the overview section of Buruganahalli above, a person of ordinary skill would have understood Buruganahalli to describe a firewall containing functionality that acts as a proxy based on its disclosure in connection with Figure 3B where a function of the firewall stands in line of communications between a client and a server in a "transparent" manner. This understanding is consistent with the description of a proxy in the '856 specification, which describes

"proxy devices" as "the man in the middle." *See* '856, 10:62-63. This understanding, as I have discussed in the overview section of Buruganahalli above, is further consistent with the prior art's description of a proxy device. *See, e.g.*, Wang, 8:12-19; NIST, 14; Baehr, 4:50-63, 8:36-47, 8:65-9:7.

143. Buruganahalli also discloses "[s]ecurity devices (e.g., security appliances, security gateways, security services, and/or other security devices) can include various security functions (e.g., firewall, anti-malware, and intrusion prevention/detection, proxy, and/or other security functions), networking functions (e.g., routing, Quality of Service (QoS), workload balancing of network related resources, and/or other networking functions), and/or other functions. For example, routing functions can be based on source information (e.g., IP address and port), destination information (e.g., IP address and port), and protocol information." Buruganahalli, 3:5-15; *id.*, 1:36-38. In view of these disclosures, a person of ordinary skill would have understood, or at minimum would have been motivated and found it obvious, to route filtered packets to the decrypt engine of the firewall that is used as part of the proxy function, as discussed above. In this manner, the filtered encrypted packets would be decrypted using a substituted key (*i.e.*, S1 key) and inspected by the firewall; and this process would take place "[a]fter the session set-up handshaking is completed." Buruganahalli, 12:29-32, 15:22-44. A person of

ordinary skill would have further understood that routing may be implemented as a software process that may be internal within a firewall. For example, as I discussed in the overview section of Buruganahalli above, NIST discloses that "[a]pplication-proxy gateway firewalls do not require a Layer 3 (Network Layer) route between inside and outside interfaces of the firewall." NIST, 12. Accordingly, a person of ordinary skill would have understood, or would have been motivated and found it obvious, to route filtered encrypted packets to a proxy system, *i.e.*, the decrypt engine of the firewall used as part of the proxy function, which applies the "trusted man-in-the-middle" techniques as taught in Buruganahalli.

144. To the extent one argues that Buruganahalli's disclosure of session splitting described along with Figure 3B does not explicitly teach routing filtered packets to a proxy system, at minimum a person of ordinary skill would have been motivated and found it obvious to perform session splitting by routing filtered packets to a proxy system using software or the routing function as taught in Buruganahalli. This would be done for security purposes such that potentially dangerous packets would be routed to a proxy system for separate handling. Furthermore, implementing a proxy system with other functions of a firewall in this manner (same device or separate devices) was already well-known and conventional in the art, as I have discussed in Section VII.D.6. *See, e.g.*, NIST, 12-14; Baehr,

2:25-30, 5:2-8, 8:12-18, Figs. 6-7; Wang, Fig. 1, 2:32-58; Le Pennec, Fig. 3, [0021]-[0023].

145. Indeed, it would be advantageous to divert potentially dangerous traffic for separate inspection (*e.g.*, actions such as filtering and logging) by a proxy system such that the security functions performed by the proxy system are isolated from the rest of the interconnecting network. *Id.* A person of ordinary skill would have recognized that it was well known and conventional to isolate a network from direct access by an outsider by using a proxy system to intercept communications to and from the network. *See, e.g.*, Baehr, Abstract, 8:37-40. Furthermore, Buruganahalli recognizes to implement a firewall as a set of devices such that a proxy function would be implemented on its own device separate from other firewall functions. Buruganahalli, 1:22-25; 2:50-52. Such implementation would be advantageous because it would offload intensive processes, such as decryption and encryption, to a proxy system that would be dedicated to perform these tasks, in order to reduce the overall workload of a firewall. *See* NIST, 14. Moreover, isolating a proxy system from firewall functions would further Buruganahalli's security purpose to protect a system against malware or other security attacks. Buruganahalli, 3:5-8, 7:39-51, 7:60-67, 8:20-24, 10:15-20, 13:43-52. For example, as I have discussed in connection with the overview section of Buruganahalli above, such purpose is

achieved by routing potentially dangerous packets to a proxy function acting as a trusted man-in-the-middle to decrypt the encrypted data. *Id.*

C. Claim Chart

146. Furthermore, I have reviewed, had input to, and endorse, the claim charts of the accompanying Petition showing that each element of claims 1, 24-25 are rendered obvious by Buruganahalli.

XI. GROUND 2: CLAIMS 1, 24-25 ARE RENDERED OBVIOUS BY BURUGANAHALLI IN VIEW OF BAEHR

147. As explained in connection with Ground 1, the Challenged Claims 1, 24-25 are rendered obvious by Buruganahalli. To the extent it is argued that routing, by the packet-filtering system, filtered packets to a proxy system based on a determination that the filtered packets comprise data that corresponds to the one or more network-threat indicators (*e.g.*, claim limitations [1.h], [24.h], and [25.h]) is not explicitly taught by Buruganahalli, that limitation is rendered obvious by Buruganahalli in view of Baehr (Ex. 1005), which explicitly discloses a packet filtering system that sends packets aside to a proxy network.

148. Buruganahalli teaches firewall techniques based on securities policies to filter packets in communications exchanged between a client and a remote server. Buruganahalli, 4:31-40. This includes intercepting an initial clear handshake message to set up an encrypted session between the client and the server. *Id.*

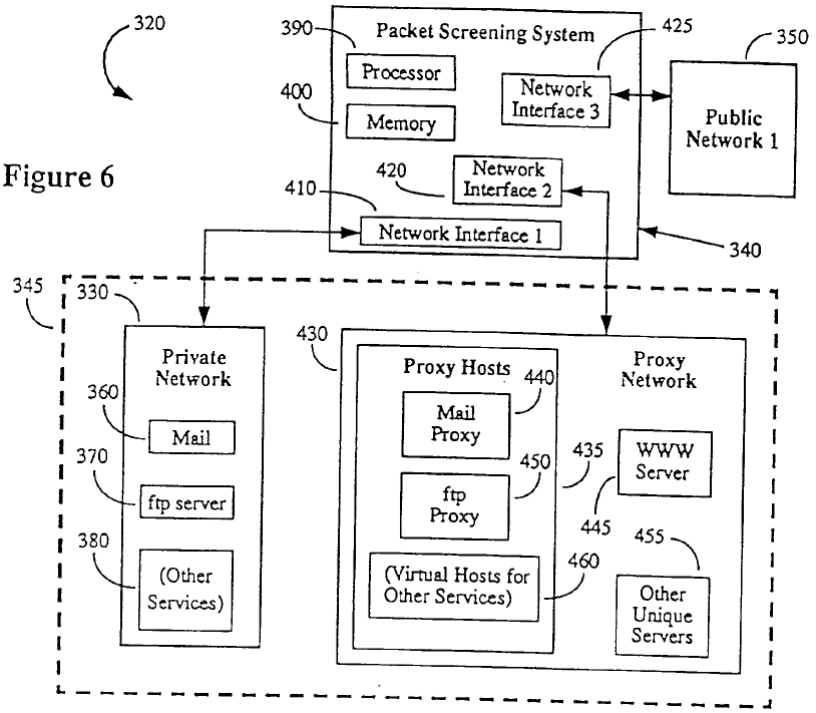
Buruganahalli also teaches that if data from the handshake message corresponds to information on a blacklist, *e.g.*, such as via a comparison of destination domains, then the packets from the encrypted session would be decrypted. Buruganahalli, 7:39-51, 7:60-67, 12:29-32. The firewall has a decrypt engine to perform the decryption by applying "trusted man-in-the-middle techniques." Buruganahalli, 9:7-8, 9:61-67. In applying these techniques, "the firewall 314 acts as the remote server 316 such that it is transparent to the client 312 that it is not communicating directly with the remote server 316," and vice versa. Buruganahalli, 12:13-33.

149. Buruganahalli further discloses "[s]ecurity devices… can include various security functions (*e.g.*, firewall, proxy…)… and/or other functions. For example, routing functions can be based on source information (*e.g.*, IP address and port), destination information (*e.g.*, IP address and port), and protocol information." Buruganahalli, 3:5-15. As I have explained above, a person of ordinary skill would have recognized that the disclosures in connection with Figure 3B of Buruganahalli describes functions for a proxy system that stands between a client and server and acts on behalf of a server to process a client's request. This understanding is further apparent by Buruganahalli's disclosure to pass packets between software functions of a firewall and route filtered packets to the proxy function. Buruganahalli, 3:16-24.

150. Buruganahalli discloses the need for "workload balancing of network related resources." Buruganahalli, 3:5-15. Buruganahalli also states that "special purpose hardware for next generation firewalls implemented, for example, as dedicated appliances, generally provide higher performance levels for application inspection than software executed on general purpose hardware…" Buruganahalli, 4:7-15. Buruganahalli further discloses that "[a] firewall is typically a device or a set of devices." Buruganahalli, 1:22-25. To reduce the workload of a firewall, a person of ordinary skill would have known, as was conventional, that the firewall would pass filtered traffic to a dedicated proxy server, rather than performing the functions of a proxy internally within a firewall. NIST, 14 ("a main firewall might accept inbound traffic, determine which application is being targeted, and then hand off the traffic to the appropriate proxy server."). Therefore, a person of ordinary skill would have been motivated and found it obvious to incorporate the teachings of Baehr, which discloses sending filtered packets to a proxy system outside of a packet screening system acting as a firewall, in implementing Buruganahalli's security system.

151. Baehr discloses "[a] system for screening data packets transmitted between a network to be protected, such as a private network, and another network, such as a public network" and the system has a "screen" that acts as a firewall and

filters packets. Baehr, Abstract, 2:10-12. Such system is illustrated in Figure 6 of Baehr. Baehr, 2:66-67.



Baehr discloses that, "[a] screen is positioned on the network connection between, for example, a public network and a private network that is to be protected from targeting for attack." Baehr, 2:10-17. Baehr discloses that the screen "includes a packet filtering subsystem or module, which inspects each incoming packet and sends it to an engine, which determines, based upon the packet inspector and other information, what actions should be taken on the packet." Baehr, 2:18-23. Baehr further discloses "[t]he screen 340 includes instructions stored in memory 400 governing its control of actions to be taken on the incoming (and outgoing) data

packets." Baehr, 6:37-39. Specifically, the screen "includes instructions stored in memory… [and] include a predetermined set of criteria based upon the… contents of the data packets (source and destination addresses…)" that allow the screen to determine whether a packet comes from or goes to a network location that may indicate a security problem. Baehr, 6:37-7:7.

152. To protect its private network, Baehr discloses routing filtered packets to a proxy network, rather than to the private network. Baehr discloses a host sending a packet to a public network that is intercepted at the interface of a screen 340. Baehr, 10:2-5, 9:63-67. The screen 340 uses a set of predetermined "screening criteria" for filtering packets. Baehr, 6:37-7:10. The screen 340 then takes actions on the packets, based on the screening, in order to prevent attacks on the system. Baehr, 7:8-25. Baehr discloses that "an important action for security purposes is that of sending packets aside to the proxy network…" Baehr, 8:12-18. This is done so for "security purposes" even if the packets' intended destination is the private network. Baehr, 2:25-30, 10:43-49. The proxy network would then execute "appropriate operations." Baehr, 2:25-30. Furthermore, "[a] proxy network may [] include proxy hosts representing actual hosts, and/or proxy hosts with unique servers, in any combination…" Baehr, 4:26-63.

153. Baehr provides reasons to explain why routing filtered packets to a proxy network, rather than sending them to a private network, would be advantageous. Baehr explains that "[t]he proxy network is isolated from the private network, so it cannot be used as a jumping off point for intruders." Baehr, *Abstract*. Therefore, it has the advantage of "preventing outsiders from ever actually entering the private network." Baehr, 8:37-40. This is in contrast to allowing a user to access the network directly, because "once a user has been allowed access or a connection to a private network, it is much more difficult to restrict his/her actions than if no access at all is allowed." Baehr, 8:37-40. Furthermore, "the private network 330 and the proxy network 430 together form a single logical or apparent network 345, i.e. a single apparent domain from the point of view of outsiders, such as users on the public network 350, so that when a user attempts to access a service or host of the private network, the request may be shunted aside to the proxy network to either a mirroring proxy host or a unique proxy host, without any indication being given to the user that this has occurred." Baehr, 4:50-63, 8:36-47, 8:65-9:7. This approach allows the proxy network to communicate seamlessly with the outsiders as if the outsiders are communicating directly with the private network. *Id.*

154. In terms of implementations, Baehr discloses that the proxy network may be a separate computer system—this is shown in Figure 6; alternatively, the

proxy network does not have to be physically separate from the screen, but rather a separate logical entity—this is shown in Figure 7. Baehr, 4:26-31, 5:2-8. Regardless of whether there is a physical separation between the screen and the proxy network, the proxy is implemented as a dedicated function separate from the screen that filters packets. Baehr further discloses that the proxy network may be implemented "entirely in program instructions stored in the memory" of the packet screening system, "or as additional processor(s) and memory(-ies) controlled by program instructions stored in one or more of the memories." Baehr, 4:64-5:2.

155. A person of ordinary skill would have been motivated and found it obvious to implement Buruganahalli's system in view of the teachings of Baehr to route the filtered packets to a proxy system separate from or included in a firewall such that Buruganahalli's techniques (that in essence describe a transparent proxy function) would have been implemented in a proxy system separate from and external to a firewall. A person of ordinary skill would have been motivated and found it obvious to implement the techniques describing a proxy in a dedicated function for the advantages of preserving computing resources, achieving load balancing, and maximizing performance, as recognized in Buruganahalli. *See, e.g.*, Buruganahalli, 3:16-24, 4:7-15. For example, having a dedicated function separate from other functions allows the proxy function to manage its own resources

efficiently without having to share resources with another function; and this approach also increases the overall system performance by allowing each function to focus on performing its own specialized tasks. These advantages were well known in the prior art. For example, NIST discloses that "dedicated proxy servers are used to decrease the work load on the firewall…" and "can be used to secure less time-sensitive services…" NIST, 14. Baehr also teaches other advantages of its implementation to isolate a private network from an outsider in order to strengthen security. For example, an isolated proxy network "cannot be used as a jumping off point for intruders" because it prevents outsiders from actually accessing the private network. Baehr, *Abstract*, 8:36-47.

156. When a proxy system is implemented as separate from a screen/firewall, the proxy system and the protected private network do not communicate directly with one another; rather, each would communicate separately with the screen/firewall via a separate interface and provide the opportunity for the screen/firewall to inspect any packets sent to or from the proxy. Therefore, having a separate proxy external to a firewall helps isolate the proxy from the private network by positioning the screen/firewall between the two networks. This configuration is also beneficial to limit any potential security vulnerability to only

one component of a larger system. For example, even if the proxy system was compromised, the screen/firewall would remain unaffected.

157. It would be routine, straightforward and advantageous to modify Buruganahalli's system to incorporate Baehr's teachings of routing to a proxy external to a firewall. A person of ordinary skill would have known that such a combination (yielding the claimed limitations) would predictably work and provide the expected functionality. Both functionalities described in Buruganahalli and Baehr are transparent to an outsider. For example, Buruganahalli discloses the firewall acts as a remote server such that it is "transparent" to the client. Buruganahalli, 12:17-21. Similarly, the description of Baehr's proxy system matches with that of a transparent proxy where the proxy is invisible to an outsider and meets the outsider's request at the same time. Baehr, *Abstract*, 4:50-63 (describing the proxy and network forming "a single apparent domain from the point of view of outsiders… without any indicator being given to the user" that the proxy network is involved in processing the user's request), 8:36-47 ("invisibly preventing him/her from ever actually accessing the private network"), 8:65-9:7 ("no IP address for the proxy host exists, and none is appended to any packets"). A person of ordinary skill would have expected the modified system to predictably work. For example, in accordance with Buruganahalli's teachings, when a firewall receives an

encrypted packet in a session associated with a suspect domain, the firewall would route the packet to an interface, such as "Network Interface 2" shown in Figure 6 of Baehr, for processing by a proxy network that is situated external to the firewall.

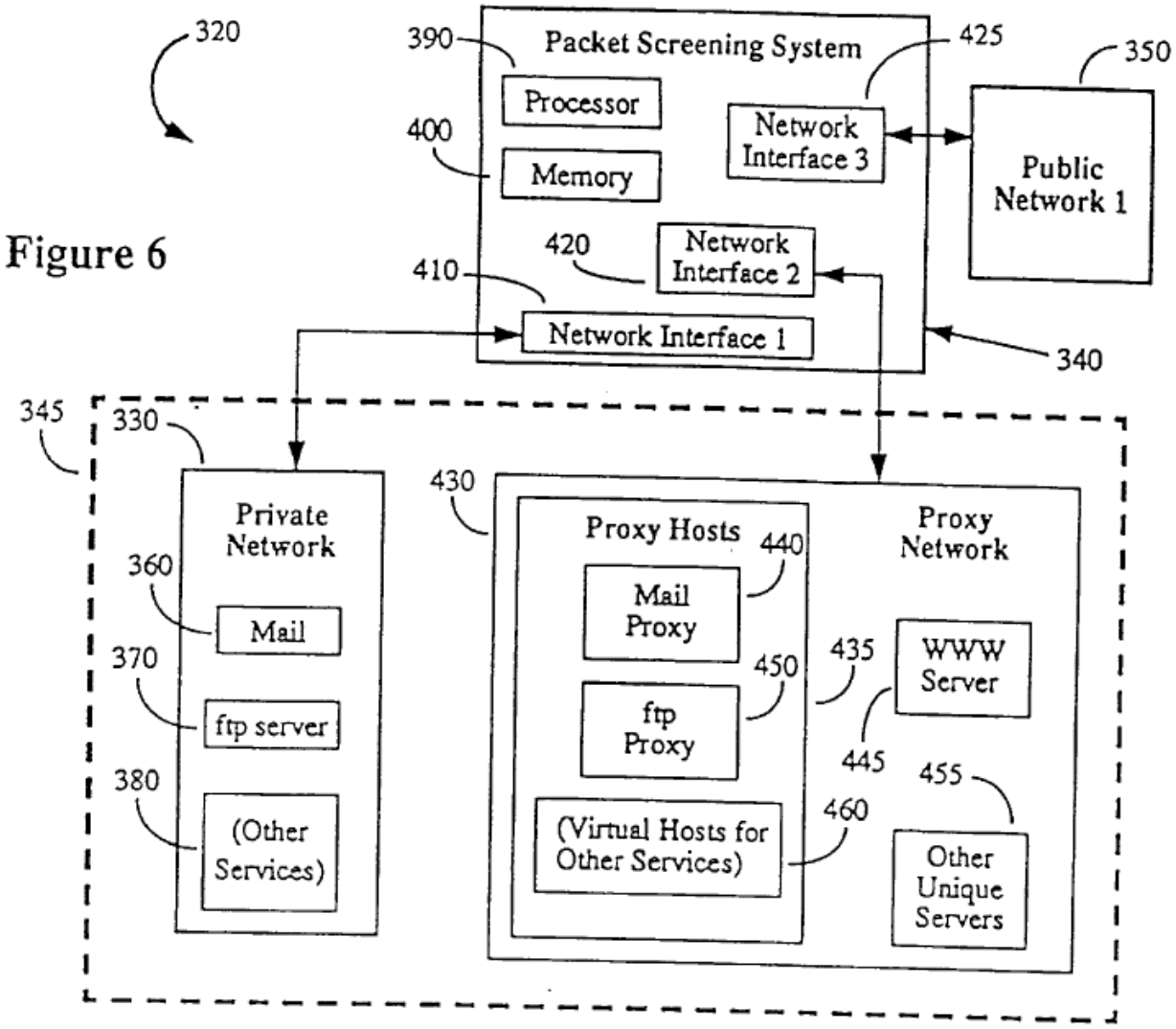


Figure 6

158. Buruganahalli and Baehr are in the same field of endeavor—data communications and security in a computer network—and are analogous art to the '856 patent. Both Buruganahalli and Baehr are directed to a packet-filtering system

to prevent unauthorized access to a network by applying a set of rules for security purposes. For example, is directed to a firewall to prevent unauthorized access to a network by applying a set of rules. Buruganahalli, 1:20-39, 2:48-3:24. Baehr is directed to a system for packet filtering of data packets at a computer network interface for protection against attacks. Baehr, Title, Abstract, 2:10-17.

XII. SECONDARY CONSIDERATIONS

159. I understand that Patent Owner raised evidence of alleged secondary considerations regarding the ’856 patent in a litigation involving Cisco. I understand that the underlying evidence is not publicly available and I have not been provided such evidence.

160. At this time, after review of the documents and materials cited herein, such as the prosecution history of the ’856 patent, I have found no evidence of secondary considerations or objective factors of non-obviousness relevant to the Challenged Claims, and it is my opinion that Challenged Claims of the ’856 patent are obvious over the prior art. To the extent Patent Owner alleges secondary considerations in this proceeding, it is my opinion that they would not outweigh the strong evidence of obviousness of the ’856 patent claims, as set forth herein. I have been informed that if the Patent Owner presents in this proceeding evidence of

objective factors that it argues support a finding of non-obviousness, I may be asked to review such evidence and to formulate an opinion and reserve the right to address any such evidence and/or arguments that Patent Owner may raise in this regard.

XIII. CONCLUSION

161. In summary, I have concluded that each of the Challenged Claims is unpatentable as obvious under 35 U.S.C. § 103 in light of the prior art references as discussed above and the knowledge of one of ordinary skill in the art, as described in this declaration.

162. Specifically, in my opinion, for at least the above reasons:

Buruganahalli alone or in view of Baehr renders obvious claims 1, 24-25.

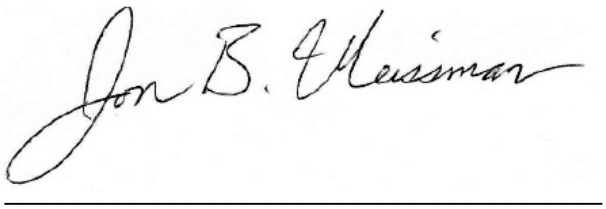
163. Furthermore, as discussed in Sections II-III above, I have used my background and experience in forming my opinions that the Challenged Claims are unpatentable as obvious over the prior art.

164. Even if it were argued that any further disclosure is required for a limitation in the Challenged Claims that I have identified above as having been disclosed by the cited references, a person of ordinary skill in the art would have found that limitation obvious to include based on the same disclosure and analysis I have identified above.

165. I currently hold the opinions expressed in this declaration. But my analysis may continue, and I reserve the right to supplement my opinions in the future to respond to any arguments that Patent Owner or its expert(s) may raise and to take into account new information as it becomes available to me. In signing this declaration, I understand that the declaration will be filed as evidence in a review proceeding before the Patent Trial and Appeal Board of the U.S. Patent and Trademark Office. I acknowledge that I may be subject to cross examination in the case and that cross examination will take place within the United States. If cross examination is required of me, I will appear for cross examination within the United States during the time allotted for cross examination.

166. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of the Title 18 of the United States Code and that such willful false statements may jeopardize the results of these proceedings.

Executed on November 18, 2021 at Minneapolis, MN.



Dr. Jon Weissman

Appendix A

JON B WEISSMAN

CURRICULUM VITAE
April 7, 2021

Work Address:
Department of Computer Science and Engineering
University of Minnesota
4-192 Keller Hall, 200 Union Street SE
Minneapolis, MN 55455-0159

Phone: (612) 626-0044
Fax: (612) 625-0572
Email: jon@cs.umn.edu
URL: http://www.cs.umn.edu/~jon

EDUCATION

Ph.D., Computer Science, University of Virginia, 1995.

M.S., Computer Science, University of Virginia, 1989.

B.S., Applied Mathematics and Computer Science, Carnegie-Mellon University, 1984.

PROFESSIONAL EXPERIENCE

Professor, Computer Science	Univ. of Minnesota (Summer 2012 to date)
Associate Professor, Computer Science	Univ. of Minnesota (Fall 2003 to Spring 2012)
Visiting Researcher and Distinguished Visitor	National e-Science Center, University of Edinburgh (2007-2008)
Assistant Professor, Computer Science	Univ. of Minnesota (1999-2003)
Assistant Professor, Computer Science	Univ. of Texas San Antonio (1995-1999)
Member of Technical Staff	Mitre Corporation, McLean Virginia (1989-1991)
Software Engineer	Software A&E, Arlington Virginia (1984-1987)

PROFESSIONAL ACTIVITIES

Tau Beta Engineering Honor Society, Member ACM, IEEE Senior Member

ACADEMIC AWARDS AND HONORS

- **Best paper nominee**, IEEE International Conference on Cloud Engineering (IC2E) 2015.
- **Best paper**, IEEE Grid conference, 2009.
- **Honorary Fellow**, College of Science and Engineering, University of Edinburgh, 2007-2008.
- **IEEE Senior Member**, promoted 2003.
- **Success Story**, ARL HPC (HPCMO), Virtual Data Grid Project, 2002.
- **CAREER Award**, National Science Foundation, 1996.
- **Supercomputing Award** for "High-Performance Computing with Legion", SC, 1995.
- **Teaching Award**, University of Virginia Teaching Medal of Excellence, 1995.

LEGAL SUPPORT EXPERIENCE

Testifying Expert

- client: Farella Braun + Martel LLP, case: Synkloud Technologies LLC vs. Adobe, area: storage, Web: IPR declaration + litigation, represented: petitioner, status: active..
- client: Lee Sullivan Shea & Smith LLP and Orrick, case: Sonos vs. Google Inc., area: network audio, project: validity expert, represented: plaintiff, *claim construction deposition*, **testifying ITC court**, status: active.
- client: Marton Ribera Schumann& Chang LLP, case: Zoho Corporation vs. Sentius International, LLC, area: web databases, project: patent analysis, expert reports, represented: defense, **claim construction tutorial to district court**, status: active.
- client: Fisch Sigler LLP, case: Packet Intelligence vs. Juniper Networks and Palo Alto Networks, area: networking, project: IPR declaration, represented: petitioner, status: finished, 2020.
- client: DLA Piper LLP, case: TMI Solutions, LLC vs. CVS, Officemax, Home Depot Product Authority LLC, area: security, network services, project: IPR declaration, represented: petitioner, status: finished, 2020.
- client: Reed Smith LLP, case: Uniloc vs. Paychex, area: distributed applications, project: prior art analysis, represented: defense, status: finished, 2020.
- client: Orrick, case: SecureWave vs. Micron Technology, Inc., area: security, project: claim construction, represented defense, status: finished, 2019.
- client: Fenwick & West LLP, case: PersonalWeb vs. Amazon, area: cloud, data storage, project: patent analysis, expert reports, represented: defense, status: settled, 2019.
- client: Jenner & Block LLP, case: Kove IO, Inc., vs. Amazon Web Service, area: cloud, data storage, project: patent analysis, IPR declaration, represented: defense, status: finished.
- client Cooley LLP, case: Zak Inc vs. Facebook, area: web, project: patent analysis, expert reports, represented: defense, status: active.
- client Cooley LLP, case: Blackberry Inc vs. Facebook, area: mobile computing, project: patent analysis, expert reports,*deposition*, represented: defense, status: active.
- client: Faegre Baker Daniels LLP, case: ICN Acquisition, LLC vs. ipDataTel, LLC, area: home alarm systems and networks, IPR declarations, represented: petitioner, *deposition*, status: settled prior to trial, 2019.
- client: Finnegan, Henderson, Farabow, Garrett & Dunner, LLP, case: KOM Software Inc. vs. NetApp, Inc. and HP, area: Distributed file systems, 5 IPR declaration, represented: petitioner, status: finished, 2019.
- client: Cooley LLP, case: Vaporstream Inc vs. Snapchat Inc, area: mobile computing, project: patent analysis, expert reports, *deposition*, represented: defense, status: stay, 2018.
- client: Fenwick & West LLP, case: IBM vs. Groupon, area: security, web, client-server, project: patent analysis, expert reports, *deposition*, **testifying district court**, represented: defense, status: finished, 2017-2018.
- client: DLA Piper LLP, Workspot, Inc. vs. Citrix Systems, INC., area: security, IPR declaration, *deposition*, represented: petitioner, status: active.
- client: Keker, Van Nest & Peters LLP, BMC Software, INC., vs. Cherwell Software, LLC, area: software systems, IPR declaration, represented: petitioner, status: active.
- client: Cooley LLP, case: Sound View Inc vs. Facebook, area: databases, web, project: patent analysis, expert reports, *deposition*, represented: defense, status: settled, 2018.
- client: Fenwick & West LLP, case: IBM vs. Groupon, area: security, IPR declaration, represented: patent owner, status: finished, 2017.

- client: Knobbe Martens, case: Ancora vs. HTC, area: security, CBM declaration, *deposition on claim construction 2019*, represented: petitioner, status: finished, 2017.
- client: Fisch Sigler LLP, case: Broadcom vs. Amazon, area: multimedia, IPR declarations, represented: petitioner, status: finished, 2017. No associated litigation.
- client: Finnegan, Henderson, Farabow, Garrett & Dunner, LLP, area: Internet applications, IPR declaration, represented: petitioner, status: finished, 2017. No associated litigation.
- client: McDonnell Boehnen Hulbert & Berghoff LLP, case: Trading Technologies Inc., area: distributed systems, project: Patent Examination, declaration, represented: patent owner, status: finished, **examiner interview**, 2016.
- client: Fenwick & West LLP, case: Actifio vs. Delphix, area: cloud storage, project: patent analysis, expert reports, represented: defense, status: finished (settled prior to trial), 2016.
- client: Fisch Sigler LLP, case: Kaavo vs. Amazon and Tier3, area: cloud computing, IPR declarations, *deposition*, represented: petitioner, status: finished, 2016.
- client: Morrison and Foerster LLP, case: Ancora vs. Apple Inc, area: security, CBM declaration, represented: petitioner, status: finished (settled prior to trial), 2016.
- client: Finnegan, Henderson, Farabow, Garrett & Dunner, LLP, area: mobile computer systems, 3 IPR declarations, represented: petitioner, status: finished, 2016. No associated litigation.
- client: Cooley LLP, case: BMC Inc vs. ServiceNow Inc, area: computer systems, project: patent analysis, expert reports, *deposition*, represented: defense, status: finished (settled prior to trial), 2016.
- client: Fish & Richardson P.C., case: Ericsson Inc vs. Apple Inc, area: mobile systems, project: patent analysis, represented: defense, status: finished, 2015.
- client: Morrison and Foerster LLP, case: Good Technology Inc vs. Airwatch Inc, area: mobile systems, project: patent analysis, 3 expert reports, *deposition*, represented: defense, status: finished (settled prior to trial), 2015.
- client: Troutman Sanders, Intellectual Ventures I LLC vs. Capitol One Financial Corp, claim construction declarations, area: mobile interfaces, represented: defense, status: finished, 2015.
- client: Bryan Cave LLP, case: Symantec vs. Veeam Software Corporation, area: storage systems, IPR declaration, *deposition*, represented: patent owner, status: finished, 2014.
- client: Finnegan, Henderson, Farabow, Garrett & Dunner, LLP, case: Clouding IP vs. Rackspace, area: storage systems, project: patent analysis, IPR and CBM declarations, *two depositions*, represented: petitioner, status: finished, 2014.
- client: Sidley Austin LLP, case: Motorola vs. Microsoft, area: mobile instant messaging, project: software patent analysis, expert reports, *deposition*, represented: defense, status: finished (settled prior to trial), 2011-2012.
- client: Baker & McKenzie LLP, case: WMR e-PIN vs. Wells Fargo, area: Internet e-commerce applications, project: software, patent analysis, expert reports, *deposition*, **testifying 3 judge panel**, represented: defense, status: closed, 2008-2009.

Consulting Expert

- client: William G. Osborne, Esq., Superior Edge vs. Monsanto, area: distributed software, project: code analysis, report, represented: plaintiff, status: closed, 2015.
- client: Arnold & Porter LLP, Rosebud v. Adobe, area: distributed software, project: patent analysis, represented: defense, status: closed, 2015.
- client: Jones Day, Summit 6 vs. Apple, area: web technologies, project: patent analysis, represented: defense, status: closed, 2014.

- client: Ropes & Gray LLP, case: Parallel Iron vs. EMC, area: storage systems, project: patent analysis, represented: defense, status: closed, 2012-2013.
- client: Bridges & Mavrakakis LLP, case: Nokia/HTC vs. Apple, area: operating systems, project: software, patent analysis, represented: defense, status: closed, 2011.

Professional references

- Andrew Mace (Cooley): www.cooley.com/people/andrew-mace
- Larissa Bifano: (DLA Piper): www.dlapiper.com/en/us/people/b/bifano-larissa/
- Timothy Sullivan: (Faegre Baker Daniels LLP): www.faegrebd.com/en/professionals/s/sullivan-timothy-m
- Phil Haack: (Martin Ribera Schumann LLP): https://www.martonribera.com/haack
- Diek Van Nort (Morrison & Foerster LLP): www.mofo.com/people/v/van-nort-diek-o
- Joshua Goldberg (Finnegan LLP): http://www.finnegan.com/

MAJOR RESEARCH AFFILIATIONS

1. **Distributed Computing Systems Group, University of Minnesota, Founder and Co-Director.**. The Distributed Computing Systems Group (DCSG) is collection of faculty and students working on topics in parallel and distributed computing, Internet/Web, mobile, cloud, edge, IoT, big data and machine learning. Cross-cutting topics include storage, networking, operating systems/middleware, security, and applications. Research centers on system building with a focus on resource management, efficiency, reliability, and enabling newly emerging applications.

2. **Digital Technology Center, University of Minnesota, Investigator**. The Digital Technology Center (DTC)'s goal is to create, promote, and coordinate cooperative interdisciplinary advanced technology initiatives between University, government, industry, and to serve as a point of entry into research and development partnerships with various partners. The DTC is a hub of innovation and excellence at the University of Minnesota in the digital technologies serving the industrial, educational, and public needs of the state of Minnesota and the nation. The DTC integrates research, education, and outreach in digital design, computer graphics and visualization, telecommunications, intelligent data storage and retrieval systems, multimedia, datamining, scientific computation, and other digital technologies. See `dcsg.cs.umn.edu` for details.

3. **CRIS:Center for Research in Intelligent Storage, University of Minnesota, Investigator**. The Center for Research in Intelligent Storage (CRIS) is a partnership between universities and industry, featuring high-quality, industrially relevant fundamental research, strong industrial support of collaboration in research and education, and direct transfer of university developed ideas, research results, and technology to U.S. industry. Research areas include storage and memory technologies, big data, cloud computing, backup/restore, data deduplication, and file systems. See `cris.cs.umn.edu`.

GRANT FUNDING

1. **Principal Investigator**, CNS Core: Small: Scaling the IoT with Constellation, NSF, 2019-2022, $500K (PI Chandra).

2. **Co-Principal Investigator**, I/UCRC Phase II: Center on Intelligent Storage, NSF, 2016-2019, $895,516 (PI Du, Co-PIs Chandra, Lilja, Mokbel).

3. **Co-Principal Investigator**, CSR: Small: Location, location, location (L3): Support for Geo-Centric Applications, NSF, 2016-2019, $528K (PI Chandra).

4. **Co-Principal Investigator**, II-NEW: One Cloud Does Not Fit All: Minnesota Integrated cloud Systems research Testbed (MIST), NSF, 2013-2016, $350K, (PI DU, co-PIs Chandra, Zhang).

5. **Principal Investigator**, An Integrated Middleware Framework to Enable Extreme Collaborative Science, DOE, 2012-2015, $342K UMn share, collaborative with Rutgers and University of Chicago.

6. **Principal Investigator**, CSR: Medium: Enriching Mobile User Experience Through The Cloud, NSF, 2012-2015, $700K (co-PI Chandra, Karypis).

7. **Principal Investigator**, DC: Small: One Thousand Points of Light: Accelerating Data-Intensive Applications By Proxy, NSF, 2009-2012, $482K, including REU supplement $36K (co-PI Chandra).

8. **Co-Investigator**, Collaborative Research: A Multi-University I/UCRC Center on Intelligent Storage, NSF, 2009-2014, $400K, (DU PI, many co-PIs).

9. **Principal Investigator**, Grid Computing for E-Science, National E-Science Centre, University of Edinburgh, 2007-2008, $52K.

10. **Co-Investigator**, ePCRN: Electronic Primary Care Research Network, NIH Roadmap, 2006-2008, $50K (CSE share annual), (Peterson PI, Delaney PI, several co-investigators).

11. **Co-Principal Investigator**, ITR: A Data Mining and Exploration Middleware for Grid and Distributed Computing, NSF ITR, 2003-2007, $1.5M, (Kumar PI, Weissman co-PI).

12. **Investigator**, Intelligent Storage Consortium, DTC, 2003-2009, $250K (Du PI, Weissman, Lilja, Tewfik, Kim co-PIs).

13. **Investigator**, MRI: Development of a System for Interactive Analysis and Visualization of Multi-Terabyte Datasets, NSF MRI, 2004-2006, $300K (Woodward PI).

14. **Contributor**, NIH NCRR, Intelligent Data Storage Support for Microarray data, equipment, (Mayo Clinic PI, with Du), 2004, $222K.

15. **Co-Principal Investigator**, Intelligent Storage Consortium - Engenio, Engenio Information Technologies, 2004-2005, $45K, (Du PI, Weissman co-PI).

16. **Principal Investigator**, Making Parallel Computing Easy, Department of Energy (Office of Advanced Scientific Computing Research), 2002-2007, $250K, sole PI.

17. **Principal Investigator**, A Framework for Adaptive Grid Services, NSF ACR-CNS, 2003-2008, $240K, sole PI.

18. **Principal Investigator**, Towards Community Services: Putting Parallel Network Services On-line, NSF EIA-NGS Program, 2002-2003, $30K, sole PI.

19. **Co-Principal Investigator**, Collaborative Data Analysis and Visualization, NSF EIA Research Resources (Collaborative Research), 2002-2005, $500K (incl. 2 RA's shared with Du), (Woodward PI, Weissman, Du, Retzel, Wetherby co-PIs.)

20. **Investigator**, Metacomputing: Enabling Technologies and the Virtual Data Grid, Army High Performance Computing and Research Center (AHPCRC), 2000-2004, $260K, original contributor to center proposal.

21. **Principal Investigator**, Resource Management for Parallel and Distributed Systems, NSF CAREER award ACR, 1996-2001, $200K, REU supplement $5K (2000), sole PI.

22. **Principal Investigator**, Smart File Objects: An Application-directed File Access Paradigm, Texas Advanced Research Program ARP-010115-226, 1997-2000, $106K, sole PI.

23. **Participant**, Telecommunication Networks for Mobile and Distributed Computing and Communications, AFOSR-F49620-96-1-0472, 1996-2000, $2M, (PI Hiromoto).

24. **Co-Principal Investigator**, Building a Pipeline for Minority Scholars, NSF CISE MI-I CDA-9633299, 1996 - 1999, $1.3M. (Hiromoto PI, Weissman, Das, Boppana, Psarris, Jeffery, Rosen co-PIs).

25. **Fellowship**, Applying Parallel Object-Oriented Computing to Scientific Applications, NASA Graduate Student Researchers Program Fellowship, Jet Propulsion Laboratory, 1991-1995, $66K.

PUBLICATIONS

Journal Articles: for up-to-date-list, see www.cs.umn.edu/˜jon

1. Kwangsung Oh, Nan Qinm Abhishek Chandra, and **Jon Weissman**, Wiera: Policy-Driven Multi-Tiered Geo-Distributed Cloud Storage System, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 31, No. 11, Feb 2020.

2. Albert Jonathan, Mathew Ryden, Kwangsung Oh, Abhishek Chandra, and **Jon Weissman**, Nebula: Distributed Edge Cloud for Data Intensive Computing, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28, No. 11, Nov 2017.

3. Benjamin Heintz, Abhishek Chandra, Ramesh Sitaraman, and **Jon B. Weissman**, End-to-end Optimization for Geo-Distributed MapReduce, *IEEE Transactions on Cloud Computing*, Vol. 4, No. 3, July 2016.

4. Jinoh Kim, Seonho Kim, and **Jon B. Weissman**, A Security-enabled Grid System for MINDS Distributed Data Mining, *Journal of Grid Computing*, 2014.

5. Murray Cole, Shantenu Jha, Daniel S. Katz, Manish Parashar, Omer Rana, and **Jon B. Weissman**, Distributed Computing Practice for Large-Scale Science and Engineering Applications, *Concurrency: Practice and Experience*, Volume 25, Issue 11, September 2013.

6. Adam Barker, **Jon B. Weissman**, Jano van Hemert, Accelerating Service-Oriented Workflows, *IEEE Transactions on Services Computing*, 5(3), 2012.

7. Jinoh Kim, Abhishek Chandra and **Jon B. Weissman**, Passive Network Performance Estimation for Large-scale, Data-intensive Computing, *IEEE Transactions on Parallel and Distributed Systems*, 22(8), August 2011.

8. Jia Jingxi, Bharadwaj Veeravalli, and **Jon B. Weissman**, Scheduling Multi-source Divisible Loads on Arbitrary Networks, *IEEE Transactions on Parallel and Distributed Systems*, 21(4), April 2010.

9. Jinoh Kim, Abhishek Chandra and **Jon B. Weissman**, Using Data Accessibility for Resource Selection in Large-scale Distributed Systems, *IEEE Transactions on Parallel and Distributed Systems*, 20(6), June 2009.

10. Byoung-Dai Lee, **Jon B. Weissman**, Young-Kwang Nam, Adaptive middleware supporting scalable performance for high-end network services, *Journal of Network and Computer Applications*, 32(3), May 2009.

11. Adam Barker, **Jon B. Weissman** and Jano van Hemert, The Circulate Architecture: Avoiding Workflow Bottlenecks Caused By Centralised Orchestration, *Journal of Cluster Computing*, 12(2), 2009.

12. Jason D. Sonnek, Abhishek Chandra, and **Jon B. Weissman**, Adaptive Reputation-Based Scheduling on Unreliable Distributed Infrastructures, *IEEE Transactions on Parallel and Distributed Systems*, 18(11), November 2007.

13. Darin England, Bharadwaj Veeravalli, and **Jon B. Weissman**, A Robust Spanning Tree Topology for Data Collection and Dissemination in Distributed Environments, *IEEE Transactions for Parallel and Distributed Systems*, 18(5), May 2007.

14. Rahul Trivedi, Abhishek Chandra, and **Jon B. Weissman**, Heterogeity-Aware Workload Distribution in Donation Based Grids, invited to the *International Journal for High Performance Computing and Applications* (IJHPCA), 20(4), 2006.

15. Darin England and **Jon B. Weissman**, A Resource Leasing Policy for On-demand Computing, invited to the *International Journal for High Performance Computing and Applications* (IJHPCA), 20(1), 2006.

16. **Jon B. Weissman**, Darin England, and Lakshman Abburi Rao, "Integrated Scheduling: The Best of Both Worlds," *Journal of Parallel and Distributed Computing*, 63(6), 2003.

17. **Jon B. Weissman** and Byoung-Dai Lee, "The Virtual Service Grid: An Architecture for Delivering High-End Network Services", *Concurrency: Practice and Experience*, Vol. 14, No. 4, April 2002.

18. **Jon B. Weissman**, "Predicting the Cost and Benefit of Adapting Parallel Applications in Clusters", *Journal of Parallel and Distributed Computing*, Vol. 62, No. 8, August 2002.

19. **Jon B. Weissman**, Mike Gingras, and Mahesh Marina, "Optimizing Remote File Access for Parallel and Distributed Network Applications" *Journal of Parallel and Distributed Computing*, Vol. 61, No. 11, November 2001.

20. **Jon B. Weissman**, "Prophet: Automated Scheduling of SPMD Programs in Workstation Networks", *Concurrency: Practice and Experience*, Vol. 11, No. 6, August 1999.

21. **Jon B. Weissman** "Gallop: The Benefits of Wide-Area Computing for Parallel Processing", *Journal of Parallel and Distributed Computing*, Vol. 54, No. 2, November 1998.

22. **Jon B. Weissman** and Xin Zhao, "Scheduling Parallel Applications in Distributed Networks," *Journal of Cluster Computing*, Vol. 1, No. 1, May 1998, invited paper.

23. Andrew S. Grimshaw, **Jon B. Weissman**, and W. Timothy Strayer, "Portable Run-time Support for Dynamic Object-Oriented Parallel Processing," *ACM Transactions on Computer Systems*, Vol. 14, No. 2, May 1996.

24. **Jon B. Weissman** and Andrew S. Grimshaw, "A Framework for Partitioning Parallel Computations in Heterogeneous Environments", *Concurrency: Practice and Experience*, Vol. 7, No. 5, August 1995.

25. Andrew Grimshaw, **Jon B. Weissman**, Emily West, and Edmond Loyot, Jr., "Metasystems: An Approach Combining Parallel Processing and Heterogeneous Distributed Computing Systems", *Journal of Parallel and Distributed Computing*, Vol. 21, No. 3, June 1994.

26. **Jon B. Weissman**, Andrew S. Grimshaw, and Robert R. Ferraro, "Parallel Object-Oriented Computation Applied to a Finite Element Problem," *Journal of Scientific Programming*, Vol. 2, No. 4, 1993.

Books and Book Chapters

1. Benjamin Heintz, Abhishek Chandra and **Jon Weissman**, Cross-Phase Optimization in MapReduce, in Cloud Computing for Data-Intensive Applications, Springer, 2014.

2. **Jon B. Weissman** and Jinoh Kim, Network Awareness in Volunteer Networks, in Desktop Grid Computing, CRC Press, 2013.

3. Steve Chapin and **Jon B. Weissman**, Distributed and Multiprocessor Scheduling (Operating Systems Area), The Computer Science and Engineering Handbook, 2004.

Highly Selective Articles: : for up-to-date-list, see www.cs.umn.edu/˜jon

1. Zach Leidall, Abhishek Chandra, **Jon Weissman**, An Edge-based Framework for Cooperation in Internet of Things Applications, 2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge), Renton, Washington, July 2019.

2. Albert Jonathan, Abhishek Chandra, and **Jon Weissman**, Multi-Query Optimization in Wide-Area Streaming Analytics, *ACM Symposium on Cloud Computing 2018 (SOCC)*, Carlsbad, California, October 2018.

3. Feng Liu, Kate Keahey, Pierre Riteau, and **Jon Weissman**, Dynamically Negotiating Capacity between On-Demand and Batch Clusters, *International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'18)*, Nov 2018, Dallas, TX.

4. Albert Jonathan, Abhsishek Chandra, and **Jon B. Weissman**, Rethinking Adaptability in Wide-Area Stream Processing Systems, *Workshop on Hot Topics in Cloud Computing* (HotCloud '18), June 2018.

5. Kwangsung Oh, Abhishek Chandra and **Jon Weissman**, TripS: Automated Multi-tiered Data Placement in a Geo-distributed Cloud Environment, *10th ACM International Systems and Storage Conference* (SYSTOR), May 2017, Haifa Israel.

6. Kwangsung Oh, Abhishek Chandra, and **Jon Weissman**, Wiera: Towards Flexible Multi-Tiered Geo-Distributed Cloud Storage Instances, *25th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, June 2016, Kyoto, Japan.

7. Matteo Turilli, Feng Liu, Zhao Zhang, Andre Merzky, Michael Wilde, **Jon Weissman**, Daniel S. Katz, and Shantenu Jha, Integrating Abstractions to Enhance the Execution of Distributed Applications, *30th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, May 2016, Chicago, Il.

8. Albert Jonathan, Abhishek Chandra, and **Jon Weissman**, Awan: Locality-aware Resource Manager for Geo-distributed Data-intensive Applications, *IEEE International Conference on Cloud Engineering*, April 2016, Berlin, Germany.

9. Francis Liu and **Jon Weissman**, Elastic Job Bundling: An Adaptive Resource Request Strategy for Large-Scale Parallel Applications, to appear in the *International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'15)*, Nov 2015, Austin, TX.

10. John Kolb, Prashant Chaudhary, Alexander Schillinger, Abhishek Chandra and **Jon Weissman**, Cloud-Based, User-Centric Mobile Application, *IEEE International Conference on Cloud Engineering*, March 2015, Tempe, AZ.

11. Ajaykrishna Raghavan, Abhishek Chandra, and **Jon B. Weissman**, Tiera: Towards Flexible Multi-Tiered Cloud Storage Instances, *ACM Middleware*, 2014.

12. John Kolb, Will Myott, Thao Nguyen, Abhishek Chandra and **Jon B. Weissman**, Exploiting User Interest in Data-Driven Cloud-based Mobile Optimization, *IEEE Conference on Mobile Cloud Computing*, April 2014, Oxford, UK

13. Mathew Ryden, Kwangsung Oh, Abhishek Chandra and **Jon B. Weissman**, Nebula: Distributed Edge Cloud for Data Intensive Computing, *IEEE International Conference on Cloud Engineering*, March 2014, Boston, MA

14. Abhishek Chandra, **Jon Weissman**, and Benjamin Heintz Decentralized Edge Clouds, *IEEE Internet Computing*, 2013.

15. Benjamin Heintz, Chenyu Wang, Abhishek Chandra, and **Jon B. Weissman**, Cross-Phase Optimization in MapReduce, *IEEE International Conference on Cloud Engineering*, March 2013, San Francisco, CA

16. Chonglei Mei, Daniel Taylor, Chenyu Wang, Abhishek Chandra, and **Jon B. Weissman**, Sharing-aware Cloud-based Mobile Outsourcing *IEEE 5th International Conference on Cloud Computing*, 2012.

17. Atul Katiyar and **Jon B. Weissman**, ViDeDup: An Application-Aware Framework for Video De-duplication, *Third Usenix Workshop on Hot Topics in Storage and File Systems* (HotStorage'11), Portland, OR 2011.

18. Louis-Claude Canon, Emmanuel Jeannot, and **Jon B. Weissman**, A Scheduling and Certification Algorithm for Defeating Collusion in Desktop Grids, *31st IEEE International Conference on Distributed Computing Systems* (ICDCS), Minneapolis, MN, June 2011.

19. Louis-Claude Canon, Emmanuel Jeannot, and **Jon B. Weissman**, A Dynamic Approach for Characterizing Collusion in Desktop Grids, *24th IEEE International Parallel and Distributed Processing Symposium* (IPDPS), 2010.

20. Abhishek Chandra and **Jon B. Weissman**, Nebulas: Using Distributed Voluntary Resources to Build Clouds, *Workshop on Hot Topics in Cloud Computing* (HotCloud '09), June 2009.

21. **Jon B. Weissman** and Siddharth Ramakrishnan, Using Proxies to Accelerate Cloud Applications, *Workshop on Hot Topics in Cloud Computing* (HotCloud '09), June 2009.

22. Adam Barker, **Jon B. Weissman** and Jano van Hemert, Eliminating the Middle-Man: Peer-to-Peer Dataflow, *17th IEEE International Symposium on High Performance Distributed Computing* (HPDC), Boston, MA, June 2008.

23. Jinoh Kim, Abhishek Chandra and **Jon B. Weissman**, Accessibility-based Resource Selection in Loosely-coupled Distributed Systems, *28th IEEE International Conference on Distributed Computing Systems* (ICDCS), Beijing, China, June 2008.

24. Krishnaveni Budati, Abhishek Chandra, and **Jon B. Weissman**, RIDGE: Combining Reliability and Performance in Open Grid Platforms, *16th IEEE International Symposium on High Performance Distributed Computing* (HPDC), June 2007.

25. Jason D. Sonnek, Mukesh Nathan, Abhishek Chandra, and **Jon B. Weissman**, Reputation Based Scheduling on Unreliable Distributed Infrastructures, *The 26th International Conference on Distributed Computing Systems* (ICDCS), July 2006.

26. Darin England, Jayashree Sadagopan, and **Jon B. Weissman**, A New Metric for Robustness with Application to Job Scheduling, *Fourteenth IEEE International Symposium on High Performance Distributed Computing* (HPDC), 2005.

27. **Jon B. Weissman** and Xin Zhao, "Run-time Support for Scheduling Parallel Applications in Heterogeneous NOWs," *Proceedings of the Sixth IEEE International Symposium on High Performance Distributed Computing* (HPDC), August 1997.

28. **Jon B. Weissman** and Andrew S. Grimshaw, "A Federated Model for Scheduling in Wide-Area Systems," *Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing* (HPDC), August 1996.

29. **Jon B. Weissman** and Andrew S. Grimshaw, "Network Partitioning of Data Parallel Computations," *Proceedings of the Third IEEE International Symposium on High Performance Distributed Computing* (HPDC), August 1994.

Selective Articles: : for up-to-date-list, see www.cs.umn.edu/~jon

30. Albert Jonathan, Abhishek Chandra, and **Jon Weissman**, Locality-Aware Load Sharing in Mobile Cloud Computing *IEEE/ACM International Conference on Utility and Cloud Computing*, 2017.

31. Albert Jonathan, Abhishek Chandra, and **Jon Weissman**, Ensuring Reliability in Geo-Distributed Edge Cloud, *IEEE RWS* (Resilience Week), Wilmington, Delaware, 2017.

32. Kwangsung Oh, Ajaykrishna Raghavan, Abhishek Chandra and **Jon Weissman**, Redefining Data Locality for Cross-Data Center Storage, *International Workshop on Software-Defined Ecosystems* (BigSystem 2015), June 2015, Portland, OR.

33. William Myott, Thao Nguyen, Abhishek Chandra, George Karypis, and **Jon B. Weissman**, Opportunities for Data-Driven Cloud-based Mobile Optimization, invited to 2014 *International Conference on Collaboration Technologies and Systems*, Minneapolis, MN, May 2014.

34. Siddharth Ramakrishna, Robert Reutiman, Abhishek Chandra, and **Jon B. Weissman**, Accelerating Distributed Workflows With Edge Resources, *Second International Workshop on Workflow Models, Systems, Services and Applications in the Cloud (CloudFlow)*, Boston MA, May 2013.

35. Mike Cardosa, Anshuman Nangia, Chenyu Wang, Abhishek Chandra and **Jon B. Weissman**, Exploring MapReduce Efficiency with Highly-Distributed Data, *The Second International Workshop on MapReduce and its Applications* (MapReduce '11), June 2011.

36. Pradeep Sundarrajan, Abhishek Gupta, Matthew Ryden, Rohit Nair, Abhishek Chandra, and **Jon B. Weissman**, Early Experience with the Distributed Nebula Cloud, *The Fourth International Workshop on Data Intensive Distributed Computing* (DIDC '11), June 2011.

37. Shantenu Jha, Daniel S. Katz, Manish Parashar, Omer Rana, and **Jon B. Weissman**, Critical Perspectives on Large-Scale Distributed Applications and Production Grids, *10th IEEE/ACM Conference on Grid Computing* (Grid 2009), Banff, Canada, 2009, winner best paper award.

38. Adam Barker, Paolo Besanay, David Robertson, and **Jon B. Weissman**, The Benefits Of Service Choreography For Data-Intensive Computing, *Proceedings of the 2009 Workshop in Challenges of Large Applications in Distributed Environments* (CLADE), 2009.

39. Adam Barker, **Jon B. Weissman** and Jano van Hemert, Orchestrating Data-centric Workflows, *IEEE/ACM CCGrid International Symposium on Cluster Computing and the Grid* (CCGrid), Lyon, France, May 2008.

40. Vasumathi Sundaram, Abhishek Chandra and **Jon B. Weissman**, Exploring the Throughput-Fairness Tradeoff of Deadline Scheduling in Heterogeneous Computing Environments, *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS), short paper, Annapolis, MD, June 2008.

41. Krishnaveni Budati, Jinoh Kim, Abhishek Chandra and **Jon Weissman**, NGS: Service Adaptation in Open Grid Platforms, *Proceedings of the IPDPS NSF Next Generation Software Workshop*, March 2007.

42. **Jon B. Weissman**, Vipin Kumar, Varun Chandola, Eric Eilertson, Levent Ertoz, Gyorgy Simon, Seonho Kim, and Jinoh Kim, DDDAS/ITR: A Data Mining and Exploration Middleware for Grid and Distributed Computing, *Workshop on Dynamic Data Driven Application Systems - DDDAS 2007*, Beijing, China, May 2007.

43. Jinoh Kim, Abhishek Chandra, and **Jon B. Weissman**, Exploiting Heterogeneity for Collective Data Downloading in Volunteer-based Networks, *IEEE/ACM CCGrid International Symposium on Cluster Computing and the Grid* (CCGrid), 2007.

44. **Jon B. Weissman**, Seonho Kim, and Darin England, A Framework for Dynamic Service Adaptation in the Grid, *Proceedings of the IPDPS NSF Next Generation Software Workshop Denver*, Colorado, April 2005.

45. Jason D. Sonnek and **Jon B. Weissman**, A Quantitative Comparison of Reputation Systems in the Grid, *6th IEEE/ACM International Workshop on Grid Computing* (GRID).

46. **Jon B. Weissman**, Seonho Kim, and Darin England, Supporting the Dynamic Grid Service Lifecycle, *IEEE/ACM CCGrid International Symposium on Cluster Computing and the Grid* (CCGrid), 2005.

47. Darin England and **Jon B. Weissman**, "A Stochastic Control Model for the Deployment of Dynamic Grid Services", *5th IEEE/ACM International Workshop on Grid Computing* (GRID), 2004.

48. Darin England and **Jon B. Weissman**, "A Load Sharing Model for the Computational Grid," *Workshop on Job Scheduling Strategies for Parallel Processing* (JSSPP) with Sigmetrics 2004.

49. Seonho Kim and **Jon B. Weissman**, "A GA-based approach for scheduling decomposable data grid applications," *International Conference on Parallel Processing* (ICPP) 2004.

50. Byoung-Dai Lee and **Jon B. Weissman**, "Adaptive Resource Selection for Grid-enabled Network Services," *IEEE 2nd International Symposium on Network Computing and Applications* (NCA), 2003.

51. Byoung-Dai Lee and **Jon B. Weissman**, "Adaptive Resource Scheduling for Network Services," *IEEE 3rd International Workshop on Grid Computing*, 2002.

52. Byoung-Dai Lee and **Jon B. Weissman**, "Community Service Toolkit for Rapid Deployment of Network Services," *IEEE 2002 Workshop on Cluster Computing*, 2002.

53. **Jon B. Weissman** and Pramod Srinivasan, "Ensemble Schedulng: Resource Co-Allocation on the Computational Grid", *IEEE 2nd International Workshop on Grid Computing* (GRID), November 2001.

54. Byoung-Dai Lee and **Jon B. Weissman**, "An Adaptive Service Grid Architecture for Dynamic Replication," *IEEE 2nd International Workshop on Grid Computing* (GRID), November 2001.

55. **Jon B. Weissman** and Byoung-Dai Lee, "The Service Grid: Supporting Scalable Heterogenous Services in Wide-Area Networks," *The IEEE 2001 Symposium on Applications and the Internet* (SAINT), January 2001.

56. **Jon B. Weissman**, "Scheduling Multi-Component Applications in Heterogeneous Wide-area Networks," *Heterogeneous Computing Workshop*, International Parallel and Distributed Processing Symposium, May 2000.

57. **Jon B. Weissman**, "Fault Tolerant Wide-Area Parallel Computing," *IEEE Workshop on Fault-Tolerant Parallel and Distributed Systems*, International Parallel and Distributed Processing Symposium, May 2000.

58. Mike Gingras and **Jon B. Weissman**, "Smart Multimedia File Objects," *IEEE Workshop on Internet Applications* (WIA), July 1999.

59. **Jon B. Weissman**, "Smart File Objects: A Remote File Access Paradigm," *Sixth ACM Workshop on I/O in Parallel and Distributed Systems* (IOPADS), May 1999.

60. **Jon B. Weissman**, "The Interference Paradigm for Network Job Scheduling," *Heterogeneous Computing Workshop*, Tenth International Parallel and Distributed Processing Symposium, April 1996.

61. **Jon B. Weissman**, Andrew S. Grimshaw, and Robert R. Ferraro, "Parallel Object-Oriented Computation Applied to a Finite Element Problem," *Object-Oriented Numerics*, 1993.

62. Lisa Sokol, **Jon B. Weissman**, and Paula Mutchler, "The Role of Event Granularity in Parallel Simulation Design," *Proceedings of the Parallel and Distributed Simulation Conference* (PADS), 1992.

63. Richard Weatherly, David Seidel, and **Jon B. Weissman**, "Aggregate Level Simulation Protocol," *Proceedings of the 1991 Summer Computer Simulation Conference*, July 1991.

64. Lisa Sokol, **Jon B. Weissman**, and Paula Mutchler, "An Empirical Comparison of MTW and Time Warp in a Shared Memory Environment," *Proceedings of the SCS Multiconference on Work in Progress (WSMC)*, January 1991.

65. Lisa Sokol, **Jon B. Weissman**, and Paula Mutchler, "MTW: An Empirical Performance Study," *Proceedings of the Winter Simulation Conference (WSC)*, December 1991.

Short Papers

1. Zach Leidall, Abhishek Chandra, **Jon Weissman** Creating an Abstraction to Exploit IoT Synergies, *2nd ACM/IEEE International Conference on Internet-of-Things Design and Implementation* (IoTDI 2017), Extended Abstract/Poster, April 2017.

2. Arvanitis TN, Taweel A, Zhao L, Delaney BC, Peterson KA , Speedie SM, Sim I, **Weissman J**, Fontaine P, Lange C, Janowiec M, and Stone J, Supporting e-trials over distributed networks: a tool for capturing randomised control, *MEDNET* 2007.

3. Seonho Kim, **Jon B. Weissman**, Kevin A. Peterson, A Secure Federated Health Data Query System for Primary Care Clinical Trials on the Grid, *2007 Annual Meeting NIH Cancer Grid* (CABig), February 2007.

4. Speedie SM, Peterson K, Fontaine P, **Weissman J**, Delaney B, Arvanitis T, Taweel A, Sim I, Lange C, Janowiec M, Stone J, Wolff, A., An Infrastructure for Conducting Clinical Trials in Primary Care. "Current Issues in Clinical Research: Latest Trends in Clinical Research" (abstract) October 4-5, 2006, Minneapolis, MN.

5. Rahuk Trivedi, Abhishek Chandra, and **Jon B. Weissman**, Platform-of-Platforms: A Modular Integrated Resource Framework for Large-Scale Services, *2nd Symposium on Networked System Design and Implementation* (NSDI), 2005, (poster paper).

6. **Jon B. Weissman** and Zhi-li Zhang, "Smart Environments: Middleware Building Blocks for Pervasive Network Computing," *NSF workshop on Infrastructure for Mobile and Wireless Systems*, Springer-Verlag Research Notes, 2003.

7. Byoung-Dai Lee and **Jon B. Weissman**, "Dynamic Replica Management in the Service Grid," *10th IEEE International Symposium on High Performance Distributed Computing* (HPDC), August 2001.

8. Mike Karo, Christopher Dwan, John Freeman, **Jon B. Weissman**, Miron Livny, and Ernest Retzel, "Appyling Grid Technology to Bioinformatics", with Mike Karo et. al, *10th IEEE International Symposium on High Performance Distributed Computing* (HPDC), August 2001.

9. **Jon B. Weissman**, "Fault Tolerant Computing on the Grid: What are my options?," *Eighth IEEE International Symposium on High Performance Distributed Computing* (HPDC), August 1999.

10. **Jon B. Weissman**, "Metascheduling: A Scheduling Model for Metacomputing Systems," *Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing* (HPDC), 1998.

11. Andrew Grimshaw, **Jon B. Weissman**, and Emily A. West, "Overhead versus Application Granularity in Mentat," *Proceedings of the Workshop on Efficient Implementation of Concurrent Object-Oriented Programs* (OOPSLA), Washington D.C., September, 1993, extended abstract.

12. Andrew Grimshaw, **Jon B. Weissman**, and Emily A. West, "Experiences with the Mentat Metasystems Testbed," *1992 Workshop on Cluster Computing*, December 1992, extended abstract.

Technical Reports and Other Papers

1. Ajaykrishna Raghavan, Abhishek Chandra, and **Jon B. Weissman**, Tiera: Towards Flexible Multi-Tiered Cloud Storage Instances, University of Minnesota Computer Science TR-14-003, January 2014.

2. Mathew Ryden, Abhishek Chandra, and **Jon Weissman**, Nebula: Data Intensive Computing over Widely Distributed Voluntary Resources, University of Minnesota Computer Science TR-13-07, March 2013.

3. Chonglei Mei, Daniel Taylor, Chenyu Wang, Abhishek Chandra, and **Jon B. Weissman**, Mobilizing the Cloud: Enabling Multi-User Mobile Outsourcing in the Cloud, University of Minnesota Computer Science TR-11-029, November 2011.

4. Dan Katz, Shantenu Jha, Manish Parashar, Omer Rana, and **Jon Weissman**, Analysis of Production Distributed Computing Infrastructures, Technical Report CI-TR-7-0811. Computation Institute, University of Chicago & Argonne National Laboratory, September 2011.

5. Chonglei Mei, James Shimek, Chenyu Wang, Abhishek Chandra and **Jon Weissman**, Dynamic Outsourcing Mobile Computation to the Cloud, University of Minnesota Computer Science TR-11-006, March 2011.

6. Louis-Claude Canon, Emmanuel Jeannot, and **Jon B. Weissman**, A Scheduling Algorithm for Defeating Collusion, INRIA, Research Report RR-7403, Oct. 2010.

7. Siddharth Ramakrishnan, Robert Reutiman, Abhishek Chandra, and **Jon B. Weissman**, Standing on the Shoulders of Others: Using Proxies to Opportunistically Boost Distributed Applications, University of Minnesota Computer Science TR-10-012, 2010.

8. Siddharth Ramakrishnan, Robert Reutiman, Harlan Iverson, Shimin Lian, Abhishek Chandra, and **Jon B. Weissman**, Early Experience with Mobile Computation Outsourcing, University of Minnesota Computer Science TR-10-020, 2010.

9. Vasumathi Sundaram, Abhishek Chandra, **Jon B. Weissman**, Exploring the Throughput-Fairness Tradeoff of Deadline Scheduling in Heterogeneous Computing Environments, University of Minnesota Computer Science TR-08-003, 2008.

10. Jinoh Kim, Abhishek Chandra, and **Jon B. Weissman**, OPEN: Passive Network Performance Estimation for Data-intensive Applications, University of Minnesota Computer Science TR-08-041, 2008.

11. Seonho Kim, Jinoh Kim, and **Jon B. Weissman**, A Security-Enabled Grid Middleware for Distributed Data Mining, University of Minnesota Computer Science TR-07-001, 2007.

12. Abhishek Chandra, Rahul Trivedi, and **Jon B. Weissman**, Hosting Services on the Grid: Challenges and Opportunities, University of Minnesota Computer Science TR-05-026, 2005.

13. Lakshman Abburi Rao, **Jon B. Weissman**, MPI-based Adaptive Parallel Grid Services, University of Minnesota Computer Science TR-03-032, 2003.

14. **Jon B. Weissman**, "The Virtual Data Grid: Just-in-time Data Delivery and Data Production for HPC Applications", DOD High Performance Computing Modernization Program Success Story Book, 2002.

15. **Jon B. Weissman**, "Metacomputing Research at the AHPCRC", AHPCRC Bulletin, 2002.

16. **Jon B. Weissman** and Colin Gan, "The Virtual Data Grid: An Infrastructure to Support Distributed Data-Centric Applications," *DOD High Performance Computing Modernization Conference Program* (Users Group Conference 2002), June 2002.

17. **Jon B. Weissman**, "Grids in the Classroom," *IEEE Concurrency*, education feature, July 2000.

INVITED TALKS

(keynotes in bold)

1. Invited Talk, IUCRC Meeting, talk title: Geo-distributed Cloud Storage, May 2017.
2. Invited **Inaugural Keynote**, Open Cloud Institute, University of Texa s at San Antonio, talk title: Geo-Distributed Clouds, September 2015.
3. Invited Talk, NITRD MAGIC, talk title: AIMES: Abstractions and Integrated Middleware For Extreme-Scale Science, March 2015.
4. Invited Talk, Inria-ENS, Lyon, France, talk title: Job Shaping on the Cloud, July 2014.
5. Invited Talk, 2014 International Conference on Collaboration Technologies and Systems (CTS 2014), talk title: Opportunities for Data-Driven Cloud-based Mobile Optimization, May 2014.
6. Invited **Keynote**, NSF/ARL Workshop on Cloud Robotics: Challenges and Opportunities, Drexel University, Feb 2013.
7. Invited Panelist, HPDC, The Value of HPDC, June 2012.
8. Invited Talk, CMU, talk title: Extending Mobile Devices into the Cloud, 2012.
9. Invited Talk, Inria-PGH, talk title: TBD, June 2012.
10. Invited Talk, NSF US-China Workshop on Collaborative Software Research, talk title: TBD, 2012.
11. Invited Talk, Thomson-Reuters Inc, talk title: A Tour of MapReduce/Hadoop, 2012.
12. Invited Talk, Adventium Labs Inc, talk title: Minnesota Cloud Research, November 2011.
13. Invited Talk, Georgia Tech (Open Cirrus Summit), talk title: Expanding the Horizons of Cloud Computing Beyond the Data Center, October 2011.
14. Invited Talk, UMn CS Colloquium, talk title: Scheduling Edge Resources to Build Better Distributed Systems, October 2011.
15. Invited **Keynote**, The Fourth International Workshop on Data Intensive Distributed Computing (DIDC 2011), San Jose, California, talk title: Wide-Area Clouds, 2011.
16. Invited Talk, Microsoft, Cloud Futures Workshop, talk title: Expanding the Horizons of Cloud Computing Beyond the Data Center, June 2011.
17. Invited Talk, Inria-Grenoble, Grenoble, France, talk title: Cloud Computing Using Edge Resources, May 2011.
18. Invited Talk, Inria-sponsored Workshop Scheduling for Large Scale systems, Aussois, France, talk title: Scheduling in the Cloud, May 2011.
19. Invited Talk, UMn CS Colloquium, talk title: Cloud Computing: Research Perspectives, April 2011.
20. Invited Talk, University of Minnesota, Science & Engineering Day, talk title: Charting the Cloud Landscape, March 2011.
21. Invited Talk, NSF Workshop: The Science of Cloud Computing, talk title: Minnesota Cloud Research Vision, March 2011.
22. Invited Talk, University of Edinburgh, Informatics/CS, talk title: New Cloud Models for Distributed Dynamic Data-Intensive Programming Application Systems, November 2010.
23. Invited Talk, Inria-sponsored Workshop: Scheduling for Large Scale systems, Aussois, France, talk title: Living on the Edge: Scheduling Edge Resources Across the Cloud, May 2010.
24. Invited Talk, UMn IT Dean Series, talk title: CloudSystems@UMn, April 2010.

25. Invited Talk, Louisiana State University, talk title: Living on the Edge: Broadening the Vision of Cloud Computing, Baton Rouge, Louisiana, March 2010.

26. Invited **Keynote**, Third Workshop on Desktop Grids and Volunteer Computing Systems (PCGrid), Rome, Italy, talk title: Cloud@Grid, 2009.

27. Invited Talk, University of Florida, Workshop on Data Mining for Biomedical Applications, talk title: Communication is Key to a Happy Marriage-Linking Biomedical Applications and the Open Grid, September 2008.

28. Invited Talk, Sabbatical lecture: University of Edinburgh, Informatics/CS, talk title: Grid Computing for E-Science, July 2008.

29. Invited Talk, University of Edinburgh, Informatics/CS, Edinburgh, talk title: Challenges in Reliable Grid Computing, July 2008.

30. Invited Talk, CLADE Workshop, Boston, MA, talk title: Scalable Computing on Open Distributed Systems, June 2008.

31. Invited **Keynote**, ASCI conference, Delft, Netherlands, talk title: Challenges in Reliable Grid Computing, June 2008.

32. Invited Talk, NeSC Science Advisory Board, talk title: Grid Computing for E-Science, May 2008.

33. Invited Talk, University of Edinburgh, Informatics/CS, NeSC Research Seminar, talk title: Harnessing Volunteer Grids, September 2007.

34. Invited Talk, University of Edinburgh, E-Science Institute Public Lecture: talk title: Decentralizing Grids, November 2007.

35. Invited Talk: Vrjie University, Amsterdam, Netherlands, talk title: The Dynamic Grid, November 2007.

36. Invited Talk, University of Edinburgh, Informatics/CS, talk title: Achieving Reliability in Open Grids, April 2007.

37. Invited Talk, University of Chicago, talk title: Achieving Reliability in Open Grids, March 2007.

38. Invited Talk, Vrjie University, Amsterdam, Netherlands, talk title: Trust-Sensitive Scheduling on the Open Grid, March 2006.

39. Invited Education Talk, UMn Institute of Technology, talk title: Recruiting Workshop talk, Winter 2006.

40. Invited Talk, Inria/NSF-sponsored Workshop: Scheduling for Large Scale distributed platforms, San Diego, California, talk title: Scheduling Challenges in Hosting Services on the Public Grid, 2005.

41. Invited Education Talk, University of Wisconsin, Eau Claire, talk title: Graduate Studies in Computer Science at the "U", 2005.

42. Invited Talk, DOE-ANL, talk title: Future Technology Challenges for Collaboration: Customizing the Grid, 2004.

43. Invited Talk, Inria/NSF-sponsored Workshop: Scheduling for Large-Scale Distributed Platforms, Aussois, France, talk title: Robust Grid Scheduling, 2004.

44. Invited Talk, Digital Technology Center, Open House, talk title: Emerging Technologies: Grid Computing, 2003.

45. Invited Talk, Minnesota IT Leadership Forum, Minneapolis, MN, talk title: Emerging Technology: Grid Computing, October 2002.

46. Invited Talk, Digital Technology Center, Grid Computing: Collaborative Opportunities within the DTC, December 2002.

47. Invited demonstration, HPDC, title: Community Services Project, July 2002.

48. Invited Talk, University of Indiana, Argonne National Laboratory/University of Chicago, talk title: Community Services: Putting High-end Network Services On-line, University of Tennessee, University of Minnesaota, 2002.

49. Invited Talk, Dagstuhl Seminar on Management of Metacomputers," Saarbrucken, Germany, talk title: Scheduling Computational Grids: A Five-Year Retrospective, June 2001.

50. Invited Talk, UMn CS Colloquium, talk title: Turning Hype into Reality: Achieving High Performance in a Computational Grid, Fall 1999.

51. Invited Talk, North Carolina State University, Virginia Polytechnic Institute, Northwestern, Indiana University, University of Kentucky, University of Oregon, talk title: Smart File Objects, Spring 1999.

52. Invited Talk, Southwest Research Institute, San Antonio, TX, talk title: Metacomputing Systems and Tools, 1998.

53. Invited Talk, Southwest Research Institute, San Antonio, TX, talk title: Tools for Application Scheduling in Parallel and Distributed Systems, 1997.

54. Invited Talk, UT San Antonio Seminar, talk title: A Survey of Wide-Area File Systems: Issues, Challenges, and Solutions, 1997.

55. Invited Talk, National Cancer Institute, NIH, talk title: Legion: The Next Logical Step Toward a World-Wide Virtual Computer, 1995.

56. Invited Talk, Nasa Goddard, talk title: Legion: The Next Logical Step Toward a World-Wide Virtual Computer, 1995.

57. Invited Talk, Nasa Jet Propulsion Laboratory, talk title: Scheduling Parallel Computations in a Heterogeneous Environment, 1994.

58. Invited Talk, Nasa Jet Propulsion Laboratory, talk title: Multigranular Scheduling of Data Parallel Programs, Nasa Jet Propulsion Laboratory, 1993.

TEACHING

Courses Taught

University of Minnesota

- CSci 2021 Machine Organization and Architecture: 2011.
- CSci 8101 Advanced Operating Systems: 2011, 2008, 2007, 2002.
- CSci 4211 Introduction to Computer Networks: 2009.
- CSci 4061 Operating Systems: 2017, 2016, 2014, 2013, 2011, 2010, 2006, 2001.
- CSci 8980 Special Topics in Distributed Systems: 2015, 2012, 2004
- CSci 5103 Graduate Operating Systems: 2015, 2012, 2009, 2000-2005.
- CSci 5105 Introduction to Distributed Systems: 2014, 2013.
- CSci 4131/5131 Internet Programming: 2004, 2001.
- CSci 5980 Special Topics (Metacomputing): 2000.

STUDENT ADVISING

Ph.D:

- Albert Jonathan *Multi-Tenant Geo-Distributed Analytics*, 2019.
- Kwangsung Oh, *Exploiting Heterogeneous Resources in a Multi-cloud Environment*, 2019.
- Francis Liu, *Elastic Scheduling in HPC Resource Management Systems*, 2019.
- Zachary Leidall, area: Internet of Things, (WPE Spring 2019).
- Jinoh Kim, Ph.D. 2010, *Data Dissemination for Distributed Computing*
  Post-doc Lawrence Berkeley Research Lab,
  Associate Professor Texas A&M
- Darin England, Ph.D. 2006, *Robust Design for Distributed Computing Systems*, Assistant Teaching Professor, University of Minnesota.
- Byoung-Dai Lee, Ph.D. 2003, *Adaptive Middleware for High-End Network Services*
  Assistant Professor at Kyonggi University, South Korea.

MS Plan A/B/C:

- Aravind Alagiri Ramkumar, Spring 2019, Plan B
- Rohit Sinhu, Spring 2019, Plan B
- Saurabh Gupta

Graduated:

- Albert Jonathan (co-advised Chandra), Fall 2016, Plan B
- Prashant Chaudhary (co-advised Chandra), Spring 2015, Plan B
- Ajaykrishna Raghavan (co-advised Chandra), Spring 2014, Plan B
- Mathew Ryden (co-advised Chandra), Spring 2013, Plan B
- Chenyu Wang (co-advised Chandra), Spring 2012, Plan A
- Atul Katiyar, Spring 2010, Plan B
- Siddarth Ramakrishnan, Spring 2010, Plan B
- Omer Jilani, Summer 2009, Plan A-Edinburgh
- Vasumathi Sundaram, Summer 2008, Plan B
- Krishnaveni Budati, Summer 2007, Plan B
- Rahul Trivedi, Fall 2006, Plan B
- Jayashree Sadagopan, Summer 2005, Plan B
- Jason Sonnek, Summer 2005, Plan B
- Anusha Iyer, Spring 2004, Plan B
- Murali Sangubhatla, Summer 2003, Plan B
- Zhaoxin Ding, Fall 2003, Plan B

- Lakshman Abburi, Fall 2003, Plan B
- Pramod Srinivasan, Summer 2001, Plan B

SERVICE

Professional Service

Leadership

- General Chair, HPDC, Phoenix Arizona, 2019
- General Chair, HPDC, Washington D.C., 2017
- Co-Organizer The Science of Cyberinfrastructure: Research, Experience, Applications and Models Workshop, e-Science, 2016.
- Co-Organizer The Science of Cyberinfrastructure: Research, Experience, Applications and Models Workshop, HPDC, 2015.
- General Chair, HPDC, New York City, NY, 2013
- Local Chair, PACT, Minneapolis, MN, 2012
- Panel co-organizer and panelist, Cloud Computing, UMn Open House, 2011.
- Steering Committee Chair, HPDC, 2010-to-date.
- International Liaison Co-Chair, ICDCS, 2011.
- Co-Organizer Cloud panel, ICDCS, 2011.
- Co-Organizer Dynamic Distributed Data Programming Abstractions Workshop, HPDC, 2011.
- Member advisory board, NSF FutureGrid, 2010-to-date.
- Global Program Chair, Europar, 2009.
- Co-Organizer, NSF Workshop on Data Mining for Biomedical Applications, University of Florida, 2008.
- Co-Organizer, Distributed Programming Abstractions Workshop at E-Science Conference, 2008.
- Program Co-Chair, HPDC 2008.
- Session Chair, HPDC: 2007, 1999,
- Session Chair, GRID, 2004.
- Panel organizer, Intelligent Storage Workshop, UMn DTC, 2006.
- Publicity Chair, ICPP, 2004, 2000.
- Co-Organizer, Intelligent Storage Workshop, UMn DTC, 2004.
- Co-Organizer, Data Mining and the Grid Workshop, UMn MSI, 2003.
- Exhibits and Poster Chair, HPDC, 2003.
- Program Vice-Chair, ICPP, 2002.
- Tutorial Chair, HPDC: 2001, 2000.
- Session Chair, IPDPS 2000.

Program Committee Member

- The International ACM Workshop on Software-Defined Ecosystem (BigSystems), 2015.
- IEEE International Workshop on Cloud Analytics 2014.
- IEEE International Conference on Cloud Engineering (IC2E) 2016, 2015, 2014, 2013.
- IEEE/ACM International Symposium on High Performance Distributed Computing: 2016-1997.
- IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing: 2016, 2015, 2014, 2012, 2011, 2010, 2009, 2007, 2003.
- Europar 2014.
- 2013 IEEE International Conference on Big Data 2013.
- IEEE International Parallel and Distributed Processing Symposium 2012.
- SC 2012, 2011, 2010, 2009, 2008, 2007, 2006.
- Workshop on Job Scheduling Strategies For Parallel Processing: 2011, 2010, 2007, 2006.
- Workshop on Large-Scale, Volatile Desktop Grids: 2011, 2010, 2009, 2007.
- IEEE International Conference on Grid Computing: 2010, 2009, 2007, 2006, 2005, 2004, 2003, 2002.
- Cluster Computing and the Grid: 2011, 2010, 2009, 2007, 2003.
- Challenges of Large Applications in Distributed Environments: 2010, 2009, 2006, 2005, 2004, 2003.
- International Conference on Parallel Processing: 2009, 2005, 2004, 2003.
- International Conference on Distributed Computing Systems: 2008.
- IEEE International Conference on e-Science and Grid Technologies: 2005.
- IEEE International Conference on Services Computing: 2004.

- Heterogeneous Computing Workshop: 2004, 2003, 2002, 2001, 2000, 1999.
- International Workshop on Active Middleare Services (Autonomic Computing): 2004, 2003, 2002.
- International Workshop on Heterogeneous and Adaptive Computing: 2003.
- International Conference on High Performance Computing: 2002.

Editorships

Current:

- Editor HPDC 1992-2012 special issue
- Editorial board: IEEE Transactions on Computers.
- Editorial board: IEEE Transactions on Parallel and Distributed Systems.
- Editorial board: Cluster Computing.

Appendix B

LIST OF EXHIBITS

EX1001	U.S. Patent No. 9,917,856 ("'856")
EX1002	File History of U.S. Patent No. 9,917,856 ("'856 FH")
EX1004	U.S. Patent No. 9,680,795 ("Buruganahalli")
EX1005	U.S. Patent No. 5,878,231 ("Baehr")
EX1006	Reserved
EX1007	Barry M. Leiner et al., "A Brief History of the Internet," Internet Society, 1997, available at: https://www.internetsociety.org/internet/history-internet/brief-historyinternet/ ("Leiner")
EX1008	U.S. Pat. No. 7,032,031 ("Jungck '06")
EX1009	E. H. Spafford, "The Internet Worm Program: An Analysis," Purdue Technical Report CSD-TR-823, 1988. ("Spafford")
EX1010	Eichin, M.W. and Rochlis, J.A., "With Microscope and Tweezers: The Worm from MIT's Perspective," Communications of the ACM, Volume 32 Number 6, June 1989 ("Eichin")
EX1011	T. Eisenberg et al., "The Cornell Commission: On Morris and the Worm," Communications of the ACM, Volume 32 Number 6, June 1989 ("Eisenberg")
EX1012	J. C. Mogul, "Simple and Flexible Datagram Access Controls for Unixbased Gateways," Digital Western Research Laboratory Technical Report No. 89/4, 1989 ("Mogul")
EX1013	"FireWall-1 User Interface from 3.0," PhoneBoy's Security Theater, archived at the Internet Archive Wayback Machine on January 2, 2014 at https://web.archive.org/web/20140102143305/http://phoneboy.net/ ("FireWall-1")
EX1014	"SonicWALL Global Management System: Centralized Network Monitoring and Management Solution", 2008,

	archived at the Internet Archive Wayback Machine on December 7, 2008 at https://web.archive.org/web/20081207003303/http:/www.sonicwall.com/downloads/EC_GMS_DS_US.pdf ("SonicWALL GMS")
EX1015	"SonicWALL ViewPoint 6.0 Administrator's Guide", 2010, archived at the Internet Archive Wayback Machine on August 21, 2010 at https://web.archive.org/web/20100821022839/http://www.sonicwall.com:80/downloads/SonicWALL_ViewPoint_6_0_Admin_Guide.pdf ("SonicWALL ViewPoint")
EX1016	Heberlein, L. T.L, et al., "A Network Security Monitor," Proceedings, 1990 IEEE Computer Society Symposium on Security and Privacy, 1990 ("Heberlein")
EX1017	"NetRanger Network Security Management System User Guide". v1.3.1 Wheelgroup Corporation, 1997 ("NetRanger")
EX1018	Roesch, M., "Snort: Lightweight Intrusion Detection for Networks," Lisa (Vol. 99, No. 1, pp. 229-238), 1999 ("Roesch")
EX1019	U.S. Pat. No. 8,370,936 ("Zuk")
EX1020	W32.hllw.Doomjuice Technical Description, Symantec Security Center, updated February 13, 2007, available at: https://www.symantec.com/security-center/writeup/2004-020909-2916-99 ("Doomjuice")
EX1021	"(Vendor Issues Fix) Re: Helix Universal Server and RealServer URL Parsing Flaw in View Source Plug-in Lets Remote Users Execute Arbitrary Code with Root Privileges" Security Tracker, September 13, 2003, available at: https://securitytracker.com/id/1007692 ("Helix")
EX1022	"Transmission Control Protocol," IETF RFC 793. J. Postel, ed., September 1981, archived at Internet Archive Wayback Machine on February 2, 2007 at https://web.archive.org/web/20070202201546/https://tools.ietf.org/html/rfc793 ("RFC 793")

EX1023	"Internet Protocol," IETF RFC 791, J. Postel, ed., September 1981, archived at Internet Archive Wayback Machine on February 4, 2007 at https://web.archive.org/web/20070204151303/https://tools.ietf.org/html/rfc791
EX1024	"File Transfer Protocol," IETF RFC 765, J. Postel, ed., June 1980, archived at Internet Archive Wayback Machine on February 6, 2007 at https://web.archive.org/web/20070206005843/https://tools.ietf.org/html/rfc765 ("RFC 765")
EX1025	T. B. Lee, "The Original HTTP as defined in 1991," World Wide Web Consortium, 1991, archived at Internet Archive Wayback Machine on June 5, 1997 at https://web.archive.org/web/19970605071155/https://www.w3.org/Protocols/HTTP/AsImplemented.html ("T.B. Lee")
EX1026	*Cisco System Inc. v. Centripetal Networks, Inc.*, IPR2018-01760, Paper 41 (PTAB May 18, 2020)
EX1027	*Centripetal Networks, Inc. v. Cisco Systems, Inc.*, Case No. 2:18-cv- 00094 (E.D. Va.), Dkt. No. 621 (Opinion and Order Regarding Findings of Fact and Conclusions of Law)
EX1028	*Centripetal Networks, Inc. v. Cisco Systems, Inc.*, Case 2021-1888 (Fed. Cir.), Dkt. No. 18 (Non-Confidential Brief for Defendant-Appellant Cisco Systems, Inc.)
EX1029	*Centripetal Networks, Inc. v. Cisco Systems, Inc.*, Case No. 2:18-cv-00094 (E.D. Va.), Dkt. No. 548 (June 10, 2020 Trial Transcript)
EX1030	*Centripetal Networks, Inc. v. Keysight Techs., Inc.*, Case No. 2:17-cv-383 (E.D. Va.), Dkt No. 484 ("Keysight Markman")
EX1031	S. Turner & C. Polk, "Prohibiting Secure Sockets Layer (SSL) Version 2.0," IETF RFC 6176, March 2011, archived at Internet Archive Wayback Machine on March 18, 2011 at https://web.archive.org/web/20110318184421/https://tools.ietf.org/html/rfc6176 ("Turner")

EX1032	I. Ristic, "SSL/TLS and PKI History," April 2018, available from https://www.feistyduck.com/ssl-tls-and-pki-history/ ("Ristic")
EX1033	U.S. Pat. Appl. Pub. No. 2007/0271605 ("Le Pennec")
EX1034	U.S. Patent No. 8,504,822 ("Wang")
EX1035	U.S. Patent No. 7,793,342 ("Ebrahimi")
EX1036	J. Wack et al., *Guidelines on Firewalls and Firewall Policy*, National Institute of Standards and Technology ("NIST")
EX1037	U.S. Patent No. 8,813,228 ("Magee")
EX1038	Reserved
EX1039	Reserved
EX1040	U.S. Patent Appl. Pub. No. 2014/0283004 ("Moore")
EX1041	U.S. Patent Appl. Pub. No. 2015/0207809 ("Macaulay")
EX1042	U.S. Patent Appl. Pub. No. 2014/0089661 ("Mahadik")
EX1043	U.S. Patent No. 7,849,502 ("Bloch")
EX1044	U.S. Patent No. 7,584,352 ("Boivie")
EX1045	Young-Ho Kim, et al., *Design of Firewall in Router using Network Processor*, IEEE (Jul. 11, 2005) ("Kim")