

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TREND MICRO, INC.
Petitioner,

v.

WEBROOT, INC.
Patent Owner.

IPR2023-00657

U.S. Patent No. 10,257,224

DECLARATION OF SETH JAMES NIELSON

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS	1
III.	MATERIALS CONSIDERED	12
IV.	LEGAL STANDARDS	13
V.	Technical Background that Would Have Been Known to A POSITA	16
A.	Networking Background	16
B.	Malware and Malicious Network Traffic	20
C.	Malware Profile Services	24
VI.	OVERVIEW OF THE CHALLENGED PATENT	25

A.	THE '224 PATENT SPECIFICATION AND CLAIMS.....	25
B.	PROSECUTION HISTORY.....	27
VII.	LEVEL OF SKILL IN THE ART	28
VIII.	Overview of the Prior art	29
A.	OVERVIEW OF US 7,352,280 TO ROCKWOOD (“ROCKWOOD”)	29
B.	OVERVIEW OF “A SPATIOTEMPORAL EVENT CORRELATION APPROACH TO COMPUTER SECURITY” BY XIE (“XIE”).....	33
IX.	GROUND 1: Claims 1-20 are rendered obvious by ROCKWOOD IN VIEW OF XIE.....	37
A.	Motivation to Combine Rockwood and Xie	37
B.	Claim 1 is rendered obvious by Rockwood in view of Xie	47
1.	Limitation 1[a]	47
2.	Limitation 1[b]	47
3.	Limitation 1[c]	54
4.	Limitation 1[d]	62
5.	Limitation 1[e]	68
6.	Limitation 1[f]	73
C.	Claim 2 is rendered obvious by Rockwood in view of Xie	74
1.	Limitation 2[a]	74
2.	Limitation 2[b]	75
D.	Claim 3 is rendered obvious by Rockwood in view of Xie	76
E.	Claim 4 is rendered obvious by Rockwood in view of Xie	78
F.	Claim 5 is rendered obvious by Rockwood in view of Xie	80
1.	Limitation 5[a]	81
2.	Limitation 5[b]	83
3.	Limitation 5[c]	85

G.	Claim 6 is rendered obvious by Rockwood in view of Xie	86
1.	Limitation 6[a]	86
2.	Limitation 6[b]	87
H.	Claim 7 is rendered obvious by Rockwood in view of Xie	89
I.	Claim 8 is rendered obvious by Rockwood in view of Xie	92
J.	Claim 9 is rendered obvious by Rockwood in view of Xie	93
1.	Limitation 9[a]	93
2.	Limitation 9[b]	94
K.	Claim 10 is rendered obvious by Rockwood in view of Xie	96
1.	Limitations 10[a]	98
1.	Limitations 10[b]-10[f]	98
3.	Limitation 10[k]	102
L.	Claim 11 is rendered obvious by Rockwood in view of Xie	103
M.	Claim 12 is rendered obvious by Rockwood in view of Xie	103
1.	Limitation 12[a]	103
2.	Limitation 12[b]	105
N.	Claims 13-16 are rendered obvious by Rockwood in view of Xie	107
O.	Claim 17 is rendered obvious by Rockwood in view of Xie	110
1.	Limitation 17[h]	113
P.	Claim 18 is rendered obvious by Rockwood in view of Xie	115
1.	Limitation 18[a]	115
2.	Limitation 18[b]	116
3.	Limitation 18[c]	116
4.	Limitation 18[d]	116
Q.	Claim 19 is rendered obvious by Rockwood in view of Xie	123

R.	Claim 20 is rendered obvious by Rockwood in view of Xie	124
1.	Limitation 20[a]	124
2.	Limitation 20[b]	125
3.	Limitation 20[c]	126
4.	Limitation 20[d]	127
5.	Limitation 20[e]	129
6.	Limitation 20[f]	129
7.	Limitation 20[g]	132
8.	Limitation 20[h]	132
9.	Limitation 20[i]	133
X.	There are no objective indicia of nonobviousness.	134
XI.	Conclusion.	134

1. I, Seth James Nielson, submit the following declaration (the “Declaration”) in connection with the proceeding identified above.

I. INTRODUCTION

2. I have been retained by counsel for Trend Micro, Inc. (“Trend Micro” or “Petitioner”) as a technical expert in connection with the proceeding identified above. I submit this Declaration in support of Petitioner Trend Micro’s Petition for Inter Partes Review (the “Petition”) of United States Patent No. 10,257,224 (“the ’224 patent”) against Webroot, Inc. (“Webroot” or “Patent Owner”). All “Ex. 10XX” cites herein are to the Exhibits to the Petitions. All “Ex. _” cites followed by a letter are to exhibits to this Declaration. All citations to “Section XX” are internal citations to the sections of this Declaration. All citations to “Paragraph XX” are internal citations to the paragraphs of this Declaration.

3. I have been asked to consider whether the claims of the ’224 patent are obvious in light of the prior art discussed below. The ’224 patent is directed to gathering and analyzing network events.

4. In summary, my conclusion is that the claims of the ’224 patent are obvious in light of the prior art discussed below. As set forth in detail in this declaration, I have reached the following conclusions:

- a. Claims 1-20 are rendered obvious by Rockwood in view of Xie.

II. QUALIFICATIONS

5. I am a subject matter expert in cybersecurity, computer networks, and software engineering. I am the Founder and Chief Scientist of Crimson Vista, a computer security research and engineering company. Furthermore, I hold appointments at the University of Texas at Austin

as an Adjunct Associate Professor in the department of Computer Science and as a Cybersecurity Fellow in the Robert Strauss Center for International Security and Law.

6. I have been working with computer networks and associated technology, including parallel processing, since January 2000. My experience includes graduate-level teaching, academic research, industry employment, and consulting practice.

7. Exhibit 1003 is a current copy of my C.V. which contains all of my relevant education and work experience, a list of all publications that I have authored in the previous ten years, as well as a list of all cases in which, during the previous four years, I have testified as an expert at trial or by deposition. My experience most relevant to the opinions offered in this report is summarized below.

8. I received my B.S. in Computer Science in April of 2000. During my final undergraduate semester, I worked both as a teaching assistant for a Computer Networking course and as a researcher in the Networked Computing Lab. In these capacities, I assisted students in debugging and designing TCP/IP protocol stacks, Address Resolution Protocol implementations, and Remote Procedure Call projects. In the research lab, I collaborated on work related to resource reservations in high performance networking and MPLS analyses. Our investigations into statistical traffic engineering for bandwidth allocation resulted in a published paper entitled, “Effective Bandwidth for Traffic Engineering.”

9. During the Fall of 2000, I studied for one semester at Brandeis University in the Boston area and, as part of my research, I developed a parallel-processing system for DNA string matching. The system that I created had a central node that managed the search space and solution aggregation. A cluster of available “worker” machines registered with the central node to indicate their availability for parallel processing. The central node would identify subdivisions of the search

space that could be farmed out to the worker nodes for processing. I also studied artificial intelligence and machine learning.

10. From 2001 through 2003, I worked as a software engineer at Metrowerks (formerly Lineo, Inc.), where I had substantial responsibilities relating to software architecture, computer networking, and technical project management. In particular, I developed and maintained the GUI for the Embedix SDK (Software Development Kit), created a network communication system between a Windows front-end and Linux back-end, developed an automated system to forward Linux python scripts to a Windows GUI over the network, and developed a network-deployed packaging and automated updating system for client software. To complete these assignments, I wrote tens of thousands of lines of computer code in C++, C, Python, and Perl.

11. As part of my technical training and development at Lineo/Metrowerks, I also gained and employed an increased exposure to computer networking and network security. I tested and evaluated a prototype firewall product, built a custom VPN solution, and trained in the use of web server and mail server administration (including security). I ran my own personal MTA with an IMAP and POP3 server based on these tools under development. I also deployed a Squid web proxy for both caching and content filtering.

12. Another major networking project involved porting the Embedix SDK from Linux to Windows. For this project I used a virtual-machine/networked solution. The underlying SDK engine remained running in a Linux VM while the GUI operated in native Windows. I developed all of the code for this remote communication system.

13. While working at Lineo/Metrowerks, I also returned to BYU to pursue my Master's degree in Computer Science. In addition to the graduate level course work in wireless computer networks and compilers, I pursued graduate research in software engineering topics, with a special

emphasis on how programmers think while creating and modifying code. During my course work, I took a special topics class called “Programmer Cognition” as well as a graduate-level neuroscience class from the Psychology department.

14. My research included a study of computer architectural patterns and how those patterns might need to change as programming languages change and evolve. Based on my research, I proposed a concept called “Design Dysphasia,” wherein a programmer or software developer becomes trapped in their approach to solving problems based on the paradigms and design approaches of the programming language. My research was published as “Design dysphasia and the pattern maintenance cycle,” in the Journal Information and Software Technology August 2006. This work also was a major component of my Master’s thesis.

15. Another part of my Master’s thesis was the identification of how certain programming language concepts can be “mixed” together. I investigated practical mechanisms whereby the Python programming language could be extended to support features known as “functional programming” and “logic (or declarative) programming.” Languages with this mix of features are known as “multi-paradigm” programming languages.

16. After finishing my Master’s degree, I moved to Houston TX in 2004 to begin a PhD program at Rice University. I chose to focus my research into topics of network security including the security of peer-to-peer cooperative storage and computation technologies.

17. During the 2004 fall semester of my Ph.D. program at Rice University, I identified a security vulnerability in the Google Desktop Search that could have allowed hackers to compromise users’ computers and obtain private information. After contacting Google and assisting them in closing the vulnerability, we published the details of our investigation.

18. In 2005, I completed an internship at Google, where I designed and implemented a solution to privacy loss in Google Web Accelerator. The Google Web Accelerator was designed to increase the speed of browsing the Internet. Once installed on a user's computer, the browser would request all content through a Google Proxy. The proxy performed pre-fetching and extensive caching in order to provide fast and responsive service to the user. At the time of my internship, news reports had identified odd problems in which users of the Accelerator were accessing other individual's private pages. During my internship, I designed and implemented a prototype solution for this issue in C++.

19. In 2005, I published a paper entitled, "A Taxonomy of Rational Attacks." This paper categorized and described the various types of attacks that one might see in a decentralized, peer-to-peer network. When there is no centralized authority, users have to cooperate to obtain service. The term "rational attacks" refers to the economic incentives to not cooperate while still exploiting the system for service.

20. My Ph.D. Thesis was entitled "Designing Incentives for Peer-to-Peer Systems," and it built on this concept. Given a network where participants cannot be forced to cooperate, the operation of said network must induce cooperation by design of the outcomes. In other words, it must be in each participant's best interest to contribute to the cooperative operation. I conducted experiments including simulated extensions to the BitTorrent peer-to-peer protocol for long-term identities and mechanisms for cooperative anonymity. These systems were predecessors to modern cryptocurrency and distributed ledger technologies.

21. From 2005 through 2008, with the approval of my PhD adviser, I worked as a Security Analyst for Independent Security Evaluators (ISE). Much of my early work was spent developing a software encryption library, including the necessary tests and procedures for FIPS-

certification. The encryption library provided advanced operations such as secure data splitting and recovery.

22. In 2009, I went to work full time for ISE as a Security Analyst and later as a Senior Security Analyst. I built a number of advanced projects including a parallel-program for massive code coverage analysis, GPU hardware-accelerated AES encryption, and distributed file-system prototypes.

23. In addition to the software development, I also performed security evaluation services that included port-scanning analyses, security protocol analysis using formal and exploratory methods, and investigated security breaches.

24. I also designed and managed the implementation of a secure communication technology that splits trust between multiple SSL Certificate Authorities (CA), so that if one CA is compromised, the communication stream can still be safely authenticated. My work on the secure communications technology project led to the issuance of multiple patents. In total, I wrote hundreds of thousands of lines of code in C, C++, and Python, including projects where I had to implement the same functionality in two separate languages.

25. In 2011, I began work as a Research Scientist at Harbor Labs and continued with that consulting firm until fall 2015. I worked with a wide range of clients, specializing in network security, network communications, software architecture, and programming languages. I analyzed an extensive collection of commercial software, including software related to secure email, cloud-based multimedia delivery, document signing, anti-virus and anti-intrusion, high-performance routing, networking protocol stacks in mobile devices, PBX telecommunications software, VoIP, and peer-to-peer communications. I also analyzed security considerations for potential technology acquisitions.

26. Also at Harbor Labs, I reviewed technology and source code for multiple clients related to accusations of theft and/or misappropriation of trade secrets. These engagements included an analysis of C, C++, Java, Python, and other source code languages in high-frequency trading, e-commerce, and other similar systems.

27. I also assessed the security and privacy technologies and policies provided by a third-party vendor to the Center for Copyright Infringement (CCI). CCI represents content owners, such as the RIAA and the MPAA, in finding and reducing piracy online. Because this process necessarily involves collecting information about private individuals by scanning a network for illegal activity, I was asked to investigate and determine that the information collected from online computing devices was adequately safeguarded and protected.

28. For other clients, I have “resurrected” or re-created legacy software systems. For example, I helped one client make remote desktop-sharing/viewing systems from the mid 90’s operational. I helped them identify the most compatible components from an old CVS repository, obtain the necessary legacy hardware and software to rebuild the source code, and diagnose why the separate components weren’t completely compatible with each other. Using tools from the era (i.e., the mid-90’s), I identified and fixed these issues in C++ and Java code, and successfully demonstrated the operational system across a small cluster of networked virtual machines.

29. During my final year at Harbor Labs, I was engaged as the principal consultant with a large biomedical device firm in a twelve-month analysis of the security of their products. Notably, medical devices were for some time not considered significant threats in terms of computer security. However, recent demonstrations by security researchers of the various ways in which a malicious individual might harm a person using a medical device has shifted the thinking in the industry. Accordingly, I was engaged to assist this company in the analysis of their products,

their process, and their future roadmap in order to ensure that patients are not harmed. I and my team analyzed design documents, hardware, and a broad range of additional resources in order to expose potential problems. The security of these systems depends, in part, on the architecture and deployment of the networks in which they operate.

30. In December 2015, I left Harbor Labs to assist Ironwood Experts, LLC., as the transitional managing partner. In three months, I helped to establish a new direction, streamline operations, and wrap up difficult negotiations.

31. After handing off management responsibilities at Ironwood, I founded Crimson Vista, Inc. as a boutique computer security engineering company. Similar to the work that I did at Harbor Labs, I continue to provide technical expertise to a wide range of clients in areas of programming languages, computer networks, and network security. My expertise in the area of “security engineering” provides comprehensive analysis, design, and insight into cybersecurity concerns before, during, and after development.

32. For example, I have been retained by a start-up in telecommunications security to provide cryptography expertise and evaluations of their protocols and architectures. My team and I have prototyped new protocols, written up analyses, and presented to potential partners and investors.

33. Another start-up company retained me for guidance in matters relating to Blockchain and Smart Contracts. This technology is very much dealing with a “fad” phase where there is a lot of misinformation and hype. I guided the start-up company through analyzing where these kinds of technologies would help and where they would not. I have also provided training on Blockchain at the Data Architecture Summit and Enterprise Data World conferences.

34. I have also provided technical guidance to an antitrust team in the United States Department of Justice. Although the technologies and parties are confidential, I can disclose that I provided in-person training on technical topics and analyses of competing security products.

35. More recently, I been retained by clients, including a Fortune 100 financial institution, to provide them with post-data-breach analyses of what went wrong, the impact of the lost data, and guidance on resolution. In these engagements, I provided reverse engineering of the data to demonstrate how an attacker can or would use the compromised information, analyzed software development to determine when the system became vulnerable, and helped identify impacted customers that had been missed in the investigations.

36. I have also been retained as a vCISO (virtual Chief Information Security Officer) for a small company that needs security expertise and guidance in protecting their newly developed intellectual property. I advise the company on security policy, operations, implementations, and training.

37. Through Crimson Vista, I also invest in research and development. Recent projects include engaging with a partner to implement prototypes of communications security protocols for next-generation automobiles. I also gave a talk on “Detecting Malicious Sandboxes” at the Workshop on Defensive Deception and Trust in Autonomy, in association with the 2018 Naval Applications of Machine Learning Workshop.

38. I was the primary investigator on a government funded research project. The project dealt with automatic recovery from ransomware attacks and was funded by the United States Army. I led a team of student researchers from Johns Hopkins and Brigham Young University in developing my design and implementing a prototype.

39. I am also a contributor to another small-business research project begun by a former student and funded by the National Science Foundation. We are researching the development of a new system for secure data storage that is protected even against “insider” threats, such as a rogue IT administrator. As part of this work, I am managing several interns for Summer 2022 and directing their work on performance, analysis, and vulnerability detection.

40. I also continue to perform a wide range of code reviews for diverse technologies including CAD software, video game systems, digital mobile radios (DMRs), video streaming, and digital rights management (DRM). I am often retained for my expertise in software forensics and have been instrumental in multiple cases to discovering and identifying stolen source code or misappropriated trade secrets.

41. Moreover, I maintain ties to academia. I held adjunct appointments at Johns Hopkins University from 2014 to 2019. From July 2016 to July 2019, I also held an appointment as the Director of Advanced Research Projects in the Johns Hopkins University Information Security Institute.

42. At Johns Hopkins University I taught Network security and Advanced Network Security. I created a custom curriculum and lab experience wherein students developed their own security protocols as a class and then attempted to break their own creations. Students learned how hard it is to get security right, and how easy it is to find something wrong. I published a paper about the labwork in the Journal of Computer Science Education entitled, “PLAYGROUND: Preparing Students for the Cyber Battleground.”

43. Beyond course instruction, I also mentored Masters students at Johns Hopkins in their capstone projects. These projects included networking security and privacy concerns across a wide range of technologies including cryptography, drone security, iOS security, BitCoin, SSL

vulnerabilities, and Twitter “botnets.” My students and I have published multiple papers from these capstone projects.

44. During my tenure as the Director of Advanced Research Projects, I was tasked with developing collaborative research opportunities. Through my efforts, a wide range of student capstones have been executed with partners from the Johns Hopkins Applied Physics Lab or outside corporate partners.

45. For example, we coordinated with the company OnBoard Security to develop better security for anti-collision protocols for air traffic. Students demonstrated the potential issues related to leaving the protocol unsecured and built a working prototype of a secured variant. OnBoard and Johns Hopkins published press releases which were picked up by aviation-focused news sources.

46. I am now an adjunct professor at the University of Texas at Austin. I have taught the undergraduate Network Security and Privacy class in the Computer Science department. I also teach the Introduction to Cybersecurity Technology class in the Law School

47. I am also the co-founder and current director of the Crypto Done Right project. This project is currently hosted by Johns Hopkins University and funded by a grant from Cisco. I am transitioning the project to be hosted by my company Crimson Vista or spun out into a completely independent non-profit entity. Crypto Done Right is designed to bridge the gap between cryptography SME’s and the IT professionals that use it. It provides authoritative guidance on deployment, lifecycle, and management of cryptography in IT systems, software development, and technical management.

48. Finally, I am the author of “Cryptography in Python: Learning Correct Cryptography by Example.” I am authoring a second book on computer security that will be published later this year.

III. MATERIALS CONSIDERED

49. I have considered information from various sources in forming my opinions. Besides drawing from approximately two decades of experience in the computer industry, I also have reviewed the '224 patent and its file history, the other documents and references as cited herein, and general technical references. I understand that Counsel for Petitioner have submitted a Petition that references by exhibit number. For simplicity and convenience, I have adopted the same exhibit numbering. The exhibits are:

- Ex. 1001: U.S. Patent No. 10,257,224 to Jaroch et al. ('224 patent)
- Ex. 1002: Expert Declaration of Dr. Seth James Nielson
- Ex. 1003: Curriculum Vitae of Dr. Seth James Nielson
- Ex. 1004: U.S. Patent No. 7,352,280 to Rockwood (“Rockwood”)
- Ex. 1005: A Spatiotemporal Event Correlation Approach to Computer Security. Yinglian Xie; August 2005; CMU-CS-05-175. (“Xie”)
- Ex. 1006: Prosecution History of '224 patent
- Ex. 1008: Larry L. Peterson, Computer Networks: A Systems Approach, Morgan Kaufmann Publishers, Inc., (1996).
- Ex. 1009: Ramesh Bangia, Operating Systems and Software Diagnostics, Laxmi Publications, (December 1, 2007).
- Ex. 1010: U.S. Published Patent Application No. 2013/0067576 to Niemela.

- Ex. 1011: Eric Hutchins, Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, The Proceedings of the 6th International Conference on Information Warfare and Security, (March 17-18, 2011).
- Ex. 1012: William R. Cheswick, Firewalls and Internet Security Second Edition, Repelling the Wily Hacker 2, 106-107, (2003).
- Ex. 1013: Ross Anderson, Network Attack and Defense, Security Engineering: A Guide to Building Dependable Distributed Systems 2, 633-678, (2008).
- Ex. 1014: Wilding, Virus Bulletin, The Authoritative International Publication of Computer Virus Prevention Recognition and Removal, (July 1989).
- Ex. 1015: Federic Lardinois, Google Acquires Online Virus, Malware and URL Scanner VirusTotal, TechCrunch, (September 7, 2012).
- Ex. 1016: Jon Lasser, Implementing snort in a production environment, Special Focus Issue: Security 26, ;login: The Magazine of USENIX & Sage, 51-54, (November 2001).

IV. LEGAL STANDARDS

50. I am not an attorney, and I have relied on instructions from counsel as to the applicable legal standards to use in arriving at my opinions in this Declaration.

51. I have been informed that in a parallel district court litigation the owner of the '224 patent is arguing that the '224 patent claims should be afforded their plain and ordinary meaning. I have also been requested to adopt patent owner's position on claim construction for the purposes of rendering my opinions set forth in this declaration.

52. I understand that a patent may include independent claims and dependent claims. An independent claim stands by itself and only includes the limitations it recites. A dependent claim can depend on (a) an independent claim, or (2) another dependent claim. I understand that a dependent claim includes all of the limitations it recites plus the limitations recited in the claim from which it depends.

53. I have been informed and understand that the subject matter of a patent claim is obvious if the differences between the subject matter of the claim and the prior art are such that the subject matter of the claim as a whole would have been obvious at the effective filing date¹ to a person having ordinary skill in the art to which the subject matter pertains. I have also been informed that the framework for determining obviousness involves considering the following factors: (i) the scope and content of the prior art; (ii) the differences between the prior art and the claimed subject matter; (iii) the level of ordinary skill in the art; and (iv) any objective evidence of non-obviousness.

54. I further understand that the claimed subject matter would have been obvious to one of ordinary skill in the art if, for example, it results from the combination of known elements according to known methods to yield predictable results, the simple substitution of one known element for another to obtain predictable results, the use of a known technique to improve similar devices in the same way, the application of a known technique to a known device ready for improvement to yield predictable results, or choosing from a finite number of identified, predictable solutions with a reasonable expectation of success. I have also been informed that the analysis of obviousness may include recourse to logic, judgment, and common sense available to

¹ I have not been asked to assess whether the patent is entitled to a priority date, but it does not affect my analysis

the person of ordinary skill in the art that does not necessarily require explication in any reference. I understand that so-called “secondary considerations of non-obviousness,” must be considered in an obviousness analysis. I understand that an analysis cannot use hindsight in determining whether a patent claim is obvious. I understand that secondary considerations of non-obviousness include: (a) a long-felt but unresolved need for the invention; (b) commercial success of the invention; (c) copying of the invention; (d) praise and recognition of the invention by others; (e) licensing of the rights to the invention; and (f) unexpected results.

55. I understand that obviousness must be determined with respect to the challenged claim as a whole.

56. I understand that a challenged claim can be invalid for obviousness over a combination of prior art references if a reason existed (at the effective filing date of the Patent in question) that would have prompted a person of ordinary skill in the art to combine prior art elements in the way required by the claim in question. I understand that this requirement is sometimes called “motivation to combine.” I further understand that the prior art references themselves might provide a motivation to combine, and that sometimes common-sense as would have been known to a POSITA can provide a reason to combine what is disclosed or suggested by two or more prior art references. I also understand that market demand, rather than scientific literature, often drives innovation, so that market forces can give rise to a motivation to combine. I also understand that known work in a field might prompt variations of it to use in the same field or a different one based on design incentives or market forces if the variations would have been predictable to a POSITA. I also understand that a need or problem known in the field can provide a reason for combining elements in the manner claimed.

57. I understand that in order for a claim to be found invalid based upon a modification or combination of prior art references, a POSITA must have had a reasonable expectation of success in modifying or combining the prior art.

58. I further understand that claims are invalid as obvious if they merely recite combinations of, or obvious variations of, prior art teachings employing the understanding and knowledge of a POSITA and motivated by the general problem that the inventor faced at the time. The prior art references themselves, or any need or problem known in the field at the time of the invention can provide a motivation to combine the elements from prior art references or to attempt obvious variations on them in the claimed manner.

59. I understand that something is “inherent in” the prior art—and thus taught by it—it if necessarily follows from the prior art’s explicit disclosures.

60. I understand that the filing date of a U.S. patent is generally the filing date of the U.S. application that issued as a patent. I also understand, though, that a U.S. patent may be granted an earlier effective filing date if the patent owner properly claimed priority to an earlier patent application.

V. TECHNICAL BACKGROUND THAT WOULD HAVE BEEN KNOWN TO A POSITA

61. In this section of my Declaration, I discuss some of the background and context for the alleged invention in the ’224 Patent that would have been known to a POSITA.

A. Networking Background

62. A computer network is a general purpose communications system between devices. The generality of the system enables the network to transmit data for many types of applications, such as the Internet, video streaming, and file transfer. (Ex. 1008 at pp. 3-10). Connected devices on a network are generically called “nodes” and can be connected directly, or “linked,” or

indirectly through cooperatively “forwarding” data to a more logically distant destination. (*Id.* at 11-15).

63. Data transmitted over computer networks is broken down into relatively small chunks called “packets.” (*Id.* at p. 17). To send data from one node to another, data must be broken up into these packets, transmitted over potentially many links, and arrive at the destination for reassembly. (*Id.*). However, the process of getting the right packets to the right destination, more-or-less at the same time, and across a world-wide Internet is complex. To handle the task, network functionality is broken down into sub-components called “protocols.” A protocol is a software or hardware component that provides some kind of network functionality. A protocol provides networking services to “higher” level components and sends requests to “lower” layer components. (*Id.* at pp. 19-20). For this reason, protocols are often said as working together in layers. (*Id.* at pp. 18-20). “The general idea is that you start with the services offered by the underlying hardware [(e.g., transmitting or receiving data)], and then add a sequence of layers, each providing a higher... level of service. The services provided at the high layers are [built on] the services provided by the low layers.” (*Id.* at p. 19).

64. The two network protocols that have become almost synonymous with the Internet are TCP and IP, collectively called TCP/IP. Various applications that use network communications, such as browsers and e-mail, have their own application-level protocols that take advantage of the lower-level TCP/IP (and related) protocols. (*Id.* at pp. 21-23).

65. Importantly, application-level protocols and the network protocols (e.g., TCP/IP), are generally run separately. Applications that use networking, such as good programs (e.g., “browsers”) and bad programs (e.g., “viruses” or “malware”), are separate programs from the programs that actually send the data out on the network (the “network layer”). Both programs

execute on the same device, with the application transmitting data to the network layer for subsequent transmission on the network.

66. A network layer exposes its functionality (e.g., transmitting data on a network) to application software through an interface: “Transport protocols, however, are often directly available to application programs, and as a result, special attention is paid to how application programs gain access to the communication services they provide. This interface is generically called the application programming interface, or API.” (*Id.* at p. 25). A POSITA would have understood that this API allows application software to be decoupled from network software. That is, application software need not know how network software actually transmits data on a network; it merely knows how to cause network software to initiate such transmissions.

67. Throughout this declaration, I will use the term “network layer” to represent the lower-level components, software, protocols, and/or protocol implementations responsible for receiving and transmitting data on a network. My use of this term accords with the understanding a POSITA would have had at the time of the invention. That is, a POSITA would have understood there to be a distinction between application software and software responsible for handling network transmissions, and that such a “network layer” would encompass all components, software, protocols, and/or protocol implementations necessary to expose network communication operations to application software on a computer.

68. A network layer is typically implemented as part of an Operating System (OS). “Simply stated, network software depends on many of the services provided by the OS, and in fact network software almost always runs as part of the OS. For example, network protocols have to call the network device driver to send and receive messages, the system’s timer facility to schedule events, and the system’s memory subsystem to allocate buffers [i.e., memory space] for messages.”

(*Id.* at p. 24). In other words, a POSITA would have understood that a program that wants to send data on a network (including, e.g., the Internet) uses (and transmits data via) a network layer of the OS by invoking its API. One example of a network layer API, called the “socket” interface (which, for example, I have used extensively for nearly 30 years), gives an application program access to network communications. This API is implemented as part of multiple operating systems. “In the realm of networking, the most popular API is the socket interface originally developed in the Berkely implementation of Unix. Because of its popularity, the socket interface has been ported to nearly all versions of Unix, and a version is now even available on Microsoft Windows.” (*Id.* at pp. 25-26).

69. A POSITA, therefore, would have understood that this separation between application software and network layer (e.g., a network layer implemented as part of an OS) means that application data transmitted over a network is transmitted by a network layer executing on a device, but originally generated by application software executing on the device. A POSITA would have further understood that the network layer of a device is typically part of the operating system.

70. Another important component of network communication is the “IP address.” “Each computer on the Internet is assigned a unique numerical address, called its IP address. Each packet moving around the Internet contains the IP addresses of the sender computer and the destination computer. Using the IP addresses, the routers move the packets across the Internet until they reach their destination machines.” (Ex. 1009 at p. 3). A POSITA would have understood that because IP addresses are unique and used for routing packets between computers on a network, an IP address uniquely identifies a computer on a network. In addition, a small, defined network, sometimes called an “intranet,” uses the same technology as the Internet at large. “An intranet is

a network that exists exclusively within an organization and that is based on Internet technology.” (*Id.*). A POSITA would have understood that within such a network, an IP address would uniquely identify a machine on the network.

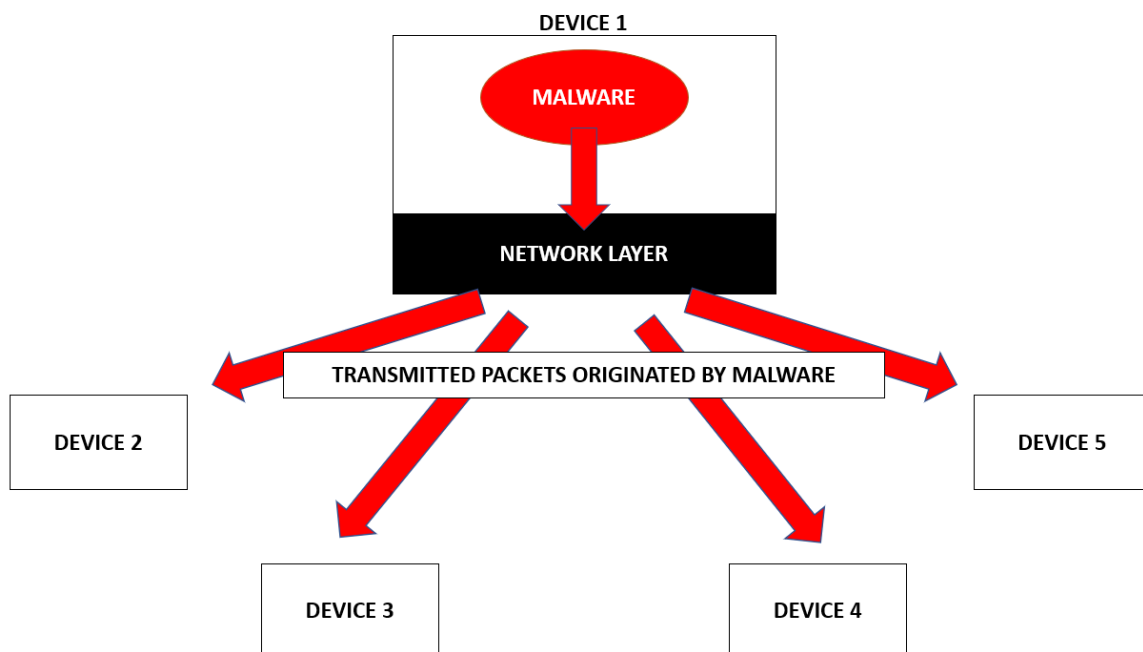
71. A POSITA would have further understood that because network transmissions originating from a device would have been sent by the network layer of the device, the IP address which uniquely identifies the device would therefore also uniquely identify the network layer of the device. For example, a device with a network layer implemented as part of the OS of the device can be uniquely identified by the IP address associated with the device. To the extent that a POSITA would not have understood this to be the case, it would have been obvious to a POSITA to infer the identifying relationship between an IP address and the network layer (e.g., a network layer implemented as part of the OS).

B. Malware and Malicious Network Traffic

72. “Malware” is the generic term given to all software that causes harmful (or unauthorized at the very least) operations to a victim’s computing systems. “Malicious software, also known as malware, refers to any software designed to infiltrate or damage a computer system without the owner’s informed consent. Malware may include multiple types of software, such as computer viruses, worms, trojan horses, rootkits, adware, spyware, ransomware and any other malicious and unwanted software. Once malicious software gains access to a computer system, it takes various actions resulting in a variety of undesirable results depending on the type of malware.” (Ex. 1010 at [0002]-[0003]). Generally, the goal of such malicious access is “violating the confidentiality, integrity, or availability of a system in the environment.” (Ex. 1011 at p. 6).

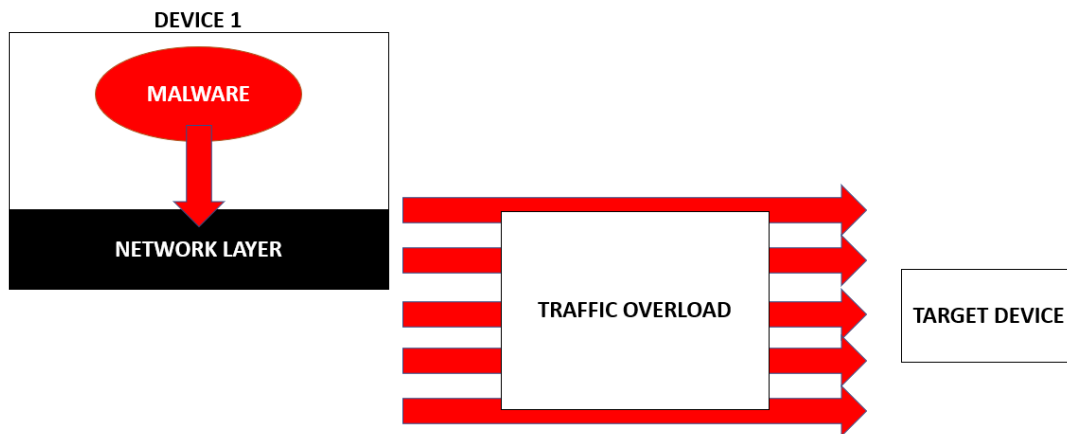
73. Malware, even though it is typically software running on a computer, often either performs harmful network operations or otherwise has an impact on network traffic. I provide a few examples of these malware-generated network operations hereinbelow.

74. First, malware, such as viruses and worms, replicates by sending copies of itself on the network. “Exponential attacks use programs to spread themselves, multiplying their numbers quickly. When the programs travel by themselves, they are worms. When they attach to other programs, they are viruses.” (Ex. 1012 at p. 3). “A virus or worm typically has two components—a replication mechanism and a payload. A worm simply makes a copy of itself somewhere else when it’s run, perhaps by breaking into another system (as the Internet worm did) or mailing itself as an attachment to the addresses on the infected system’s address list (as many recent worms have done). In the days of DOS viruses, the commonest way for a virus to replicate was to append itself to an executable file and patch itself in, so that the execution path jumps to the virus code and then back to the original program.” (Ex. 1013 at p. 14). Either way, the malicious code causes, in one form or another, copies of itself to be transmitted in network traffic. I have visualized this process in the image below. As per my discussion in Section V.A., the malware is shown transmitting data to the network layer (e.g., the OS) that, in turn, transmits the data on the network.



75. The exact kind of damage done by the virus or worm depends on the nature of the payload. For example, a virus or worm could spread **ransomware**, which is “malware that takes a computer as hostage in order to force the user to give money to the author of the malware... [I]f the user carries out the required actions e.g. pays the ransoms, the user gets instructions for restoring the computer” (Ex. 1010 at [0007]). Ransomware operates by making the victim’s files inaccessible until the demands are met. “The ransomware may alter the files in many ways, e.g. by hiding the files (non-visible), modifying the access rights of the files, locking the files, encrypting the files, moving the files to another (hidden) location in the file system, mixing up the names or any possible combination of these.” (*Id.* at [0009]).

76. Some malware uses network packets for purposes other than replication. Some malware, for example, transmits an overwhelming amount of traffic to victims in order to disrupt their normal operations. This is called a “denial of service,” or DOS, attack. Such an attack is “the simple overuse of a service—straining software, hardware, or network links beyond their intended capacity. The intent is to shut down or degrade the quality of a service” (Ex. 1012 at p. 4). The network traffic is often marked by unusual communication patterns. I have depicted this in the following image.



77. This type of attack has another variation. “Distributed Denial-of-Service (DDoS) attacks use many hosts on the Internet. More often than not, the participating hosts are unwilling accomplices to the attack, having been compromised in some way and outfitted with some malicious code.” (*Id.*). In these scenarios, malware has infected many hosts (perhaps through self-replication strategies such as worm techniques), and then launches a coordinated strike wherein all of the compromised machines transmit data to the target device at one time.

78. Finally, malware can also exfiltrate data by stealing sensitive data on a host and transmitting it to a server controlled by the attacker. For example, these attacks may involve “targeted, socially-engineered emails dropping trojans to exfiltrate sensitive information. These intrusions were over a significant period of time, evaded conventional firewall and anti-virus capabilities, and enabled adversaries to harvest sensitive information” (Ex. 1011 at p. 3). I have illustrated data exfiltration below:



79. These examples are by no means limiting. Nevertheless, they serve to indicate some of the ways in which malware acting on a computer causes the OS to send network traffic.

C. Malware Profile Services

80. Since the early 1990's, computer users have been sharing information about malware with one another in order to create community defenses. For example, Virus Bulletin was a printed magazine that began in July 1989. (See Ex. 1014). The very first issue included a list of known viruses.

The following is a list of the known viruses affecting IBM PCs and compatibles, including XTs, ATs and PS/2. If a particular virus does not execute on a 286 or a 386 machine, a statement to that effect is made in the Description section of the table. Each entry includes the virus name and aliases, its type (see the *type code* table below), hexadecimal pattern (virus signature) and its displacement within the virus and a short description of the virus and its symptoms. The hexadecimal pattern can be used to detect the presence of the virus by using any pattern searching software such as the *Norton Utilities*.

81. (Ex. 1014 at p. 4).

82. As shown above, the published magazine provided “patterns” that could be used to detect the presence of viruses. A POSITA would have understood this to be an early example of community profiles of attack software.

83. In the 2000's, other services emerged such as VirusTotal. This service stored vast amounts of known malware tagged with IDs. Users could upload a file to compare against this massive database to determine if it was on the list of bad software. (See Ex. 1015 at p. 1).

84. In addition to community sources related to files, such as viruses and malware specifically, other community sources contain profiles of bad network behavior. For example,

Snort is an intrusion detection system IDS that can be loaded with rules. As early as November 2001, it was possible to download rules from a group called Whitehats for configuring the IDS. (See, e.g., Ex. 1016, pp. 2-3). This same group is mentioned by Xie, as I explain later in this document. “We obtained the binaries and source codes of a few popular worms from public Web sites such as whitehats [111] and packetstorm [72].” (Ex. 1005 at p. 47). A POSITA would have understood that these data constitute attacker profiles.

VI. OVERVIEW OF THE CHALLENGED PATENT

A. THE '224 PATENT SPECIFICATION AND CLAIMS.

85. The '224 patent relates to “[m]ethods and systems for providing forensic visibility into systems and networks...” (’224 Patent at Abstract). As will be described in detail in Section IX. below, various claims of the '224 Patent require a ***first object*** that acts on a ***target***, and an ***originating object***. (See, e.g., Claims 1, 10, and 17).

86. The '224 Patent specification explains that an object “may be a computer file, part of a file or a sub-program, macro, web page or any other piece of code to be operated by or on the computer, or any other event whether executed, emulated, simulated or interpreted.” (’224 Patent at 7:37-41).

87. The term “first object” is mentioned in the claims and in patent summary; it does not appear in the detailed description. Instead, the concept is expressed as an object that acts on a target. An “event” captures the actions of this object on the target: “an event may include the three principal components as a key of the object performing the act (the ‘Actor’), a key of the act being performed (the ‘Event Type’), and the key of the object or identity of another entity upon which the act is being performed (the ‘Victim’ or ‘Target’).” (’224 Patent at 8:14-19).

88. The term “originating object” is also not mentioned in the detailed description of the specification of the '224 Patent. Instead, the concept of “originating processes, devices, and

users” is used. “The analysis starts by correlating events to objects (their originating processes, devices, and users), and creating an audit trail for each individual event. This allows an administrator to forensically track back any event which occurs to what triggered it.” (’224 Patent at 3:48-52). In other places, the term “GenesisActor” is used. “Contextual state information may include, but is not limited to, originating processes, devices, and users. Additionally, contextual state information may include GenesisActor information and ancillary data. GenesisActor information may contain the object that is not the direct Actor of an event but which is the ultimate parent of the event being performed.” (’224 Patent at 8:50-56).

89. The ’224 Patent states that the purported invention correlates events to objects to create contextual information, potentially using caches and global databases. “[T]he local aggregator and interpreter, receives events from the low level system filters... Following the aggregation and interpretation of the events, the context analyzer creates, or obtains a contextual state by correlating events to objects... the computing device 204 may determine if contextual state information is available.” (’224 Patent at 8:29-49). “[T]he global perspective information server 216, in some embodiments, may be able to provide contextual state information to the sensor agent 208 for one or more events.” (’224 Patent at 9:7-11).

90. However, the ’224 Patent explains, the “global perspective” is not limited to contextual state information. “The global perspective information 228 may include information related to the age, popularity, and determination of an object of the event— whether it is known good, bad, or unknown.” (’224 Patent at 9:26-29).

91. The ’224 Patent also states that the collected data, contextual state, and global information may be assembled into an “event line” that can be transmitted to a user (e.g., for display). “An event distributor of the sensor agent 208 may then create an event line comprising

information from the event information from the contextual state, information from the global perspective, and information from regarding the priority of the event.” (’224 Patent at 9:57-61). “Accordingly, the end recipient 220 may utilize the parsed event line to generate an user interface utilizing the user interface generator 444. The generated user interface may then be displayed to a user at a user output device 412.” (’224 Patent at 15:60-64).

92. As demonstrated in detail in the rest of this declaration, the ’224 Patent is not novel, but a straight-forward arrangement of known techniques.

B. PROSECUTION HISTORY

93. The application that issued as the ’224 patent, 15/437,275, was filed on February 20, 2017. (Ex. 1006, p. 379).

94. Prosecution of the 15/437,275 application that issued as the ’224 patent began with a non-final Office Action in which claims 1-20 were rejected under 35 U.S.C. § 103 as unpatentable over U.S. Patent Publication 2013/0298244 to Kumar in view of U.S. Patent Publication 2008/0016570 to Capalik. (Ex. 1006, pp. 364–369).

95. Webroot responded to the non-final Office Action by amending the independent claims and some of the dependent claims on September 5, 2017. Specifically, independent claim 1 was amended to recite “gathering an event defining an action of a first object acting on a target, wherein the first object is executed on a device; generating contextual state information for the event by correlating the event to an originating object of the first object; obtaining a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network; generating an event line comprising information relating to the event, wherein the information relates to at least one of the first object, the action of the first object, the target, and

the originating object; and transmitting the generated event line.” (Ex. 1006, pp. 170-178; underlines showing final amendments). Webroot argued that Kumar and Capalik do not disclose the limitations of “obtaining a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network.” (Ex. 1006, pp. 181–184).

96. A Terminal Disclaimer was filed on February 22, 2018 to overcome a double patenting rejection asserted in a final Office Action. (Ex. 1006, pp. 134–148). An Advisory Action issued on March 26, 2018, to which Webroot responded on April 23, 2018 with claim amendments to correct minor antecedent basis issues (e.g., replacing “a” with “the”). (Ex. 1006, pp. 114–123). A non-final Office Action issued on May 29, 2018 in which claims were rejected under 35 U.S.C. 101. (Ex. 1006, pp. 99–104). Webroot responded on August 29, 2018 with arguments to overcome the rejections. (Ex. 1006, pp. 78–96). A Notice of Allowance then issued on December 26, 2018, without an examiner’s statement of reasons for allowance. (Ex. 1006, pp. 65–69).

VII. LEVEL OF SKILL IN THE ART

97. A person of ordinary skill in the art at the time of the earliest claimed priority date for the ’224 patent would have had a bachelor’s degree in computer science, or a related field, or an equivalent technical degree or equivalent work experience, and an additional one to two years of education or experience in the field of computer security topics such as malware/exploit detection and/or network forensics. More education can supplement practical experience and vice versa..

98. I meet that definition of a person of ordinary skill in the art, and I did so on in and before May 2015. I also have worked with persons of ordinary skill in the art as part of my

professional and academic experience, and I have an understanding of their skill level in and before May 2019.

VIII. OVERVIEW OF THE PRIOR ART

A. OVERVIEW OF US 7,352,280 TO ROCKWOOD (“ROCKWOOD”)

99. U.S. Patent No. 7,352,280 to Troy Dean Rockwood (“Rockwood”) is entitled “System and Method for Intruder Tracking Using Advanced Correlation in a Network Security System.” Rockwood discloses identifying the activities of an attacker through a correlation engine that collects detected events and compares them against “target” events that represent known or suspected attacks. (Ex. 1004 at Abstract, 6:63-7:16).

100. The detected events of Rockwood are collected by sensors that process network communications. Rockwood explains, “Each sensor 20 is located at a network port that receives TCP/IP packets or other similar network communications from network 100. These packets and similar network communications received by sensors 20 are referred to as detected events 110.” (Ex. 1004 at 3:36-40). Accordingly, a sensor may be, for example, a computer that is simply configured to collect the network traffic. As explained in more detail below, Rockwood teaches that by correlating one or more events against “target” events, it can determined what or who is causing the attack.

101. Target events are not necessarily events currently happening on the network. Rockwood explains that “[t]arget event 364 is an event selected **or defined** by operator 60 to serve as the basis for correlating detected events 110 in detected event storage... operator 60 may **arbitrarily define target event 364** by inputting into console 50 attributes 410 of target event 364” (Ex. 1004 at 7:3-5, emphasis added). These operator-defined target events can be based on historical or anticipated attack vectors. Rockwood explains that, “For example, operator 60 may know from an external source that an attacker typically attacks around 3:00 p.m. using Source IP

address 205.252.48.166. Operator 60 may use this information to arbitrarily define attributes 410 of target event 364.” (Ex. 1004 at 7:51-58).

102. Additionally, Rockwood discloses that a target event can be a detected event as well. Once a detected event is designated as a target, the correlation engine can use that to find other related events on the network. “For example, operator 60 may want a particular detected event 110 to serve as target event 364 because the particular detected event 110 is an attack on internal network 70. By correlating other detected events 110 stored in detected event storage 158 with the attack that is, target event 364 system 10 may identify certain detected events 110 in detected event storage 158 that are similar to the attack.” (Ex. 1004 at 7:40-47).

103. The correlation engine disclosed by Rockwood compares detected events against the target event by creating an N-dimensional representation of each event based on its attributes. For example, a network event may identify a network operation of data being sent from one node to another. The source IP address of the sender and the destination IP address of the recipient could each be attributes of the event (among others). The correlation engine can compare the N-dimensional attributes of the detected event against the N-dimensional attributes of the target event. Rockwood explains that “correlation engine 140 uses proximity limits 370 to correlate detected events 110 in detected event storage 158. Proximity limits 370 are numerical values corresponding to particular attributes 410 of target event 364. Proximity limits 370 define a portion of n-dimensional space surrounding target point 362.” (Ex. 1004 at 8:6-12).

104. Moreover, Rockwood discloses that multiple detected events can be correlated to the target event in order to better identify the attacker. “The method concludes by identifying a plurality of detected events wherein the points defined by the attribute values of the identified detected events are within the portion of n-dimensional space defined by the proximity limits.”

(Ex. 1004 at Abstract). The correlation engine can process many events together using, for example, “[r]uleset 152 comprises instructions, algorithms, or any other directives used by correlation engine 140 to ***process, correlate, aggregate, and/or filter event information*** of detected events 110.” (Ex. 1004 at 5:17-20, emphasis added). The detected events are identified by network sensors.

105. Detected events can be gathered from various sources identified by Rockwood. Sensors are one source of detected events. “Each sensor 20 is located at a network port that receives TCP/IP packets or other similar network communications from network 100. These packets and similar network communications received by sensors 20 are referred to as detected events 110... Generally, sensor 20 processes detected events 110... Sensor 20 outputs at least detected events 110...” (Ex. 1004 at 3:36-45).

106. Data from sensors, according to Rockwood, can be processed by manager servers. “Manager server 30 represents any hardware or software module that controls or monitors one or more servant nodes, such as sensor 20. In one example, each manager server 30 includes a correlation engine 140 and a ruleset 152 for receiving and correlating detected events 110 from sensors 20.” (Ex. 1004 at 4:5-10). However, a manager server can also output detected events itself. (Ex. 1004 at 4:25-26).

107. Output data (e.g., detected events) from both sensors and manager servers can be fed into a “global server.” “Global server 40 represents any hardware, software, firmware, or combination thereof operable to process, control, and monitor system 10 at the highest logical level... Global server 40 receives detected events 110, such as from manager servers 30.” (Ex. 1004 at 4:39-41, 5:61-63). Rockwood identifies the use of a global server as a “centralized” embodiment. “The preceding examples illustrate system 10 as a centralized intrusion detection

system 10 that is, sensors 20 and manager servers 30 send detected events 110 to a centralized global server 40 via a common internal network 70. Global server 40 then processes detected events 110 and provides operator 60 with a global view of the state of system 10.” (Ex. 1004 at 14:24-29). Although Rockwood discloses that the manager server can have a correlation engine, global server can also have the correlation engine. (Ex. 1004 at 4:48-49).

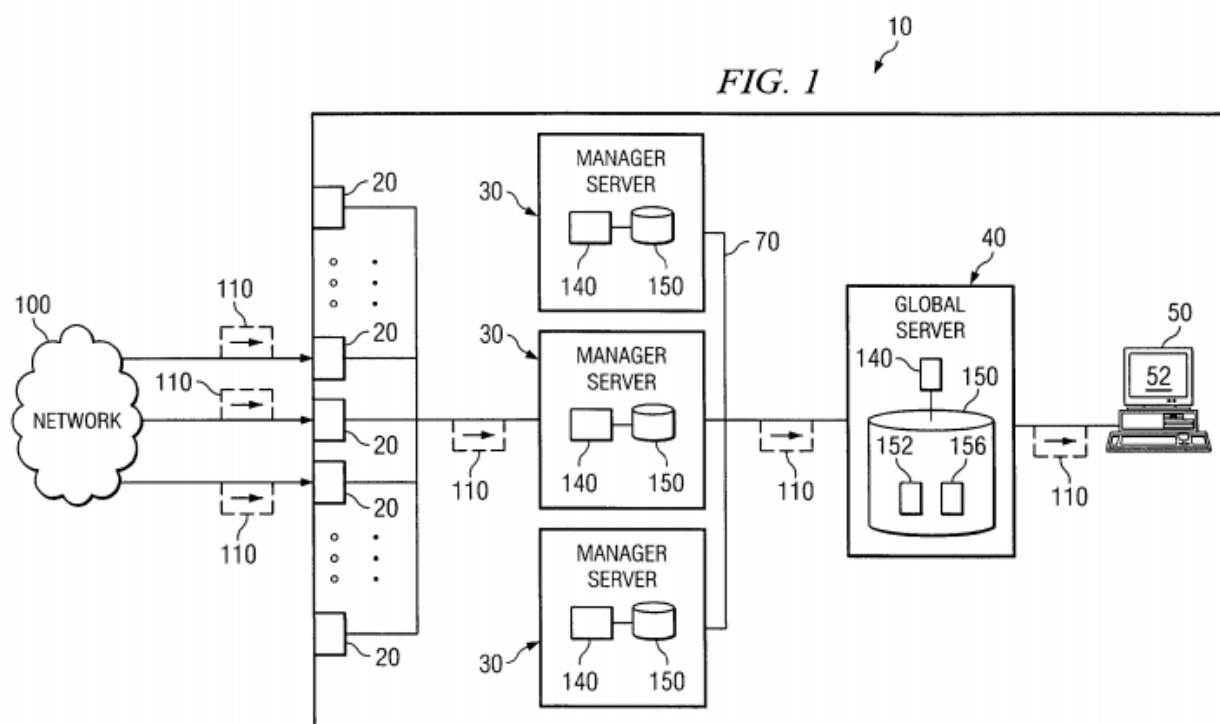
108. Alternatively, “Certain intrusion detection systems 10, however, are decentralized—that is, there is no global server 40. The present invention is operable to protect an enterprise with such a de-centralized architecture.” (Ex. 1004 at 14:30-33).

109. In addition to correlating detected events to target events, “correlation engine 140 identifies one or more existing attacker profiles 224a in profile set 156 that have characteristics that match or are similar to attributes 410 of identified detected events 222.” (Ex. 1004 at 10:27-31). Rockwood discloses having attacker profiles that identify human attackers including information identifying them in the real world or Internet world such as last known address, commonly used IP addresses, etc. that are associated with an attack or potential attack. “Each existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise’s computer network. In certain embodiments, existing attacker profile 224a includes name and background information Such as, for example, last known residence or last known employer... For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks between 5:00 p.m. and 5:30 p.m. using external computers with IP addresses between 205.252.48.160 and 205.252.48.200.” (Ex. 1004 at 10:822).

110. Rockwood also discloses attackers that are also malware. “Attacks may include viruses, Trojan horses, worms, or any other piece of code or data that represents at least a portion of an unwanted attempt to access the protected portion of the enterprise.” (Ex. 1004 at 3:1-4). As

explained in Section V.B., malware such as worms and viruses spread without direct human involvement. Thus, although Rockwood discusses human attackers with reference to attacker profiles, a POSITA would have understood that collecting analogous data for viruses and worms would yield relevant information related to the malware and associated attack behaviors. To the extent that this is not explicitly disclosed, it would have been obvious because, as I have already explained, malware and viruses are not directly controlled by a human.

111. Figure 1 of Rockwood shows the architecture of these elements, as reproduced hereinbelow:



B. OVERVIEW OF “A SPATIOTEMPORAL EVENT CORRELATION APPROACH TO COMPUTER SECURITY” BY XIE (“XIE”)

112. The PhD thesis, “A Spatiotemporal Event Correlation Approach to Computer Security,” by Yinglian Xie (“Xie”) proposes identifying malware through correlation of data. However, it identifies that many correlation approaches are temporally based or correlated across

space; Xie proposes the improvement of tracking both space and time in a single analysis. As Xie explains, “This thesis explores a new spatiotemporal event correlation approach to capture the abnormal patterns of a wide class of attacks, whose activities, when observed individually, may not seem suspicious or distinguishable from normal activity changes.” (Ex. 1005 at p. 3).

113. An example of “space” in Xie is a set of files duplicated across many computers (i.e., many Windows operating system files will be identical across computers on a network). By tracking events across the space of files as well as across time Xie increases detection effectiveness. “For anomaly detection, we present a ‘pointillist’ method to detect similar, coincident changes to the patterns of file updates that are shared across multiple hosts... in a multi-dimensional feature space.” (*Id.*).

114. Xie applies this combined space/time correlation approach in multiple ways. One application was in detecting malware on filesystems that caused malicious network activity. It did this by collecting information about file attribute modifications in order to identify files that could be the sources of a malware infection.

We have installed the Seurat data collection tool on a number of campus office machines and a teaching cluster that are used by students daily. By default, the tool scans the attributes of all system files on a host. The attributes of a file include the file name, type, device number, permissions, size, inode number, important timestamps, and a 16-byte MD5 checksum of file content. Each day, each host compares the newly scanned disk snapshot against that from the previous day and generates a file update summary report. In the current prototype, all the reports are uploaded daily to a centralized server where system administrators can monitor and correlate the file updates using the correlation module.
(Ex. 1005 at p. 40).

Since Seurat outputs a list of suspicious files as the cause of each alarm, system administrators can tell if the new clusters [of related events] are caused by malicious intrusions.
(Ex. 1005 at p. 42, emphasis added).

115. Xie demonstrated these techniques in the tracking the Linux worm known as “Lion.” As Xie explains, “Now we proceed to examine the efficacy of Seurat during a real worm outbreak... and present the result with the Lion worm [86] in this experiment.” (Ex. 1005 at p. 47). Lion is a worm that generates a lot of network traffic. “Once Lion infects a system, it sets up backdoors, leaks out confidential information (/etc/passwd, /etc/shadow) via email, and scans the Internet to recruit vulnerable systems.” (*Id.*). A POSITA would have recognized that Xie is describing network events (emails, scans, etc.) that are caused by malware.

116. In Xie’s disclosures, the tracking system identifies the malicious code files that are part of the worm (and, therefore, are related to generating the malicious traffic). “Figure 2.13 lists the 22 files selected by Seurat as the causes of the alarm. These files provide enough hints to the administrators to confirm the existence of the Lion worm. Once detected, these compromised hosts as well as the list of suspicious files can be marked for future detection.” (Ex. 1005 at p. 48).

117. One of the other aspects of Xie’s thesis is that the same file is being tracked on multiple computers, with file events on one machine acting as global context information for the others. The file modification data from this collective community of machines was cached² at a central server and analyzed for malware indications. Xie explains the analysis “regards each individual file update status as a discrete time series signal $S_{i,n}$... on day n ... is defined as the total number of hosts that update file i on day n in the network system.” (Ex. 1005 at p. 35).

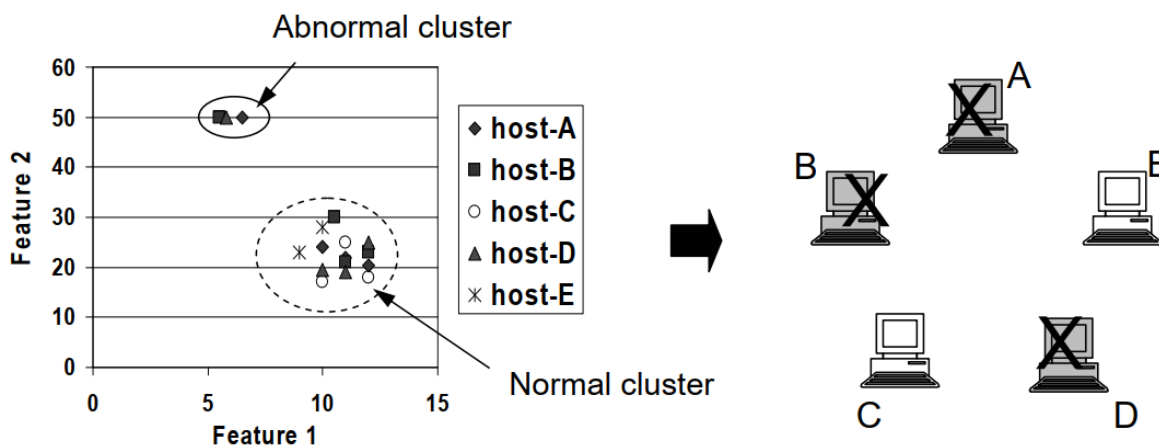
118. In Xie’s disclosure, the collective set of file updates represents a view of the changes to files across the network (a “global” view). As Xie explains, “As such, the dimension [of the feature vector] of the space is the number of file names present on any machine in the

² Caching refers to any storage of data in additional locations for performance, convenience, or operational purposes.

network system.” (Ex. 1005 at p. 33). This global view of changes to the file is an aggregate of a series of component events from different machines. However, any subset of events is also global view for that file.

119. Each update, comprising the global view, is stored across the computers that comprise the network, with each computer storing its own event information. The aggregate data is cached at a centralized server. “Each day, *each host* compares the newly scanned disk snapshot against that from the previous day and *generates a file update summary report*. In the current prototype, all *the reports are uploaded daily to a centralized server* where system administrators can monitor and correlate the file updates using the correlation module.” (Ex. 1005 at p. 40, emphasis added). The data is cached at the central server so that correlation module can work quickly on this cached data, rather than having to collect the data directly from each machine.

Figure 2.1 of Xie illustrates a part of this architecture, as reproduced hereinbelow:



120. This combined view is resilient to small amounts of loss. As Xie explains, “For those small number of missing reports, we simply ignore them because they do not affect the aggregated file update patterns.” (Ex. 1005 at p. 40).

IX. GROUND 1: CLAIMS 1-20 ARE RENDERED OBVIOUS BY ROCKWOOD IN VIEW OF XIE

121. Claims 1-20 of the '224 Patent are rendered obvious by Rockwood in view of Xie, as explained in the sections below. I also explain that it would have been obvious for a POSITA to combine Rockwood and Xie.

A. Motivation to Combine Rockwood and Xie

122. A POSITA would have been motivated to combine Rockwood and Xie. Both references are directed toward the same types of problems (i.e., identifying malware based on event analysis), and both references discuss similar or overlapping solutions. For example, Rockwood discloses “assigning a plurality of attribute values to each detected event, the attribute values of each detected event defining a point in [a conceptual] n-dimensional space.” (Ex. 1004 at Abstract). The events identified by Rockwood are primarily network data, “Each sensor 20 is located at a network port that receives TCP/IP packets or other similar network communications from network 100. These packets and similar network communications received by sensors 20 are referred to as detected events 110.” (Ex. 1004 at 3:36-40). Rockwood correlates and compares stored (previously analyzed) events to new “target events” like, for example, seeing which stored event is closest to the target event in the conceptual n-dimensional space. (See, e.g., Ex. 1004 at Abstract, 1:36-56, 6:63-7:16).

123. Rockwood is directed towards identifying “attacks” on a network. As Rockwood explains, “This invention relates to intrusion detection systems... [which] are used by an enterprise to detect and identify unauthorized or unwanted use (commonly called an attack) of the enterprise’s computer network...” (Ex. 1004 at 1:7-18). Rockwood identifies attacks caused by human intruders as well as attacks caused by malware. With respect to human attackers, Rockwood teaches identification through “attacker profiles” with information about the attacker. “Each

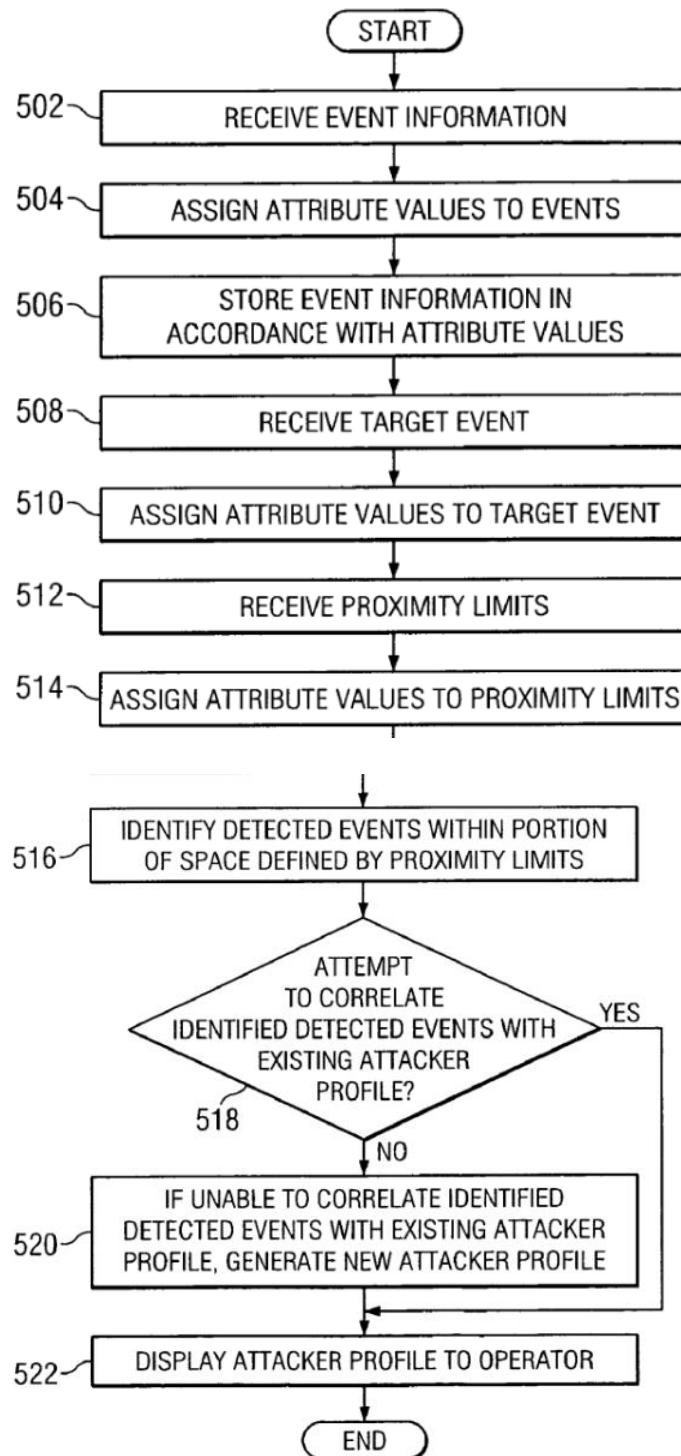
existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise's computer network. In certain embodiments, existing attacker profile 224a includes name and background information Such as, for example, last known residence or last known employer.” (Ex. 1004 at 10:8-15). Further, Rockwood also is interested in attacks caused by malicious code or data. Rockwood states that “[a]ttacks may include viruses, Trojan horses, worms, or any other piece of code or data that represents at least a portion of an unwanted attempt to access the protected portion of the enterprise.” (Ex. 1004 at 3:1-4).

124. As previously stated, Rockwood identifies attacks through event correlation. Rockwood's events are network events, which Rockwood refers to as “detected events.” “Each sensor 20 is located at a network port that receives TCP/IP packets or other similar network communications from network 100. These packets and similar network communications received by sensors 20 are referred to as detected events 110.” (Ex. 1004 at 3:36-40). Detected events are compared to “target events” which are “events selected or defined by operator 60 to serve as the basis for correlating detected events 110 in detected event storage.” (Ex. 1004 at 7:2-5). A target event may be a previously detected event that, for example, is determined to be an attack or it can be an event defined by an operator. For example, Rockwood teaches “operator 60 may want a particular detected event 110 to serve as target event 364 because the particular detected event 110 is an attack on internal network 70. By correlating other detected events 110 stored in detected event storage 158 with the attack that is, target event 364 system 10 may identify certain detected events 110 in detected event storage 158 that are similar to the attack.” (Ex. 1004 at 7:40-47). Alternatively, “Operator 60 may arbitrarily define target event 364. For example, operator 60 may know from an external source that an attacker typically attacks around 3:00 p.m. using source IP address 205.252.48.166. Operator 60 may use this information to arbitrarily define attributes 410

of target event 364.” (Ex. 1004 at 7:53-58). Target event may be “a computer virus; a trojan horse; a computer worm; [or] at least a portion of malicious computer code.” (Ex. 1004 at Claim 9). Thus, Rockwood discloses identifying malware corresponding to detected events.

125. Rockwood explains further that “proximity limits” are used to determine if detected events match target events (e.g., such as malware-identifying target events). The proximity limits work by comparing the attributes of the detected event against those of the target event in N-dimensional space. Rockwood describes this process in one example like this: “At step 516, correlation engine 140 identifies those particular detected events 110 in detected event storage 158 corresponding to points 360 that are within proximity limits 370 of target point 362... Proximity limits 370 define a portion of n-dimensional space surrounding target point 362...” (Ex. 1004 at 14:1-5; 13:65-67). Once the detected event is identified as being within the proximity limits Rockwood explains that the detected event can also be correlated to an attacker profile in an attempt to identify a human actor associated with the suspicious activity: “correlation engine 140 attempts to correlate identified detected events 222 with at least one existing attacker profile 224a in profile set 156.” (Ex. 1004 at 14:7-10). See also, e.g., Rockwood Fig. 5 reproduced herein below:

FIG. 5



126. A POSITA would have been motivated to look for additional references in order to better identify the source of attacks caused by autonomous malware. A POSITA would have found

that Xie teaches how to solve the problem. Xie discloses an expanded event model capable of correlating events not just from network sources, as Rockwood does, but also from the filesystem. As Xie explains, “[w]e use the file name, including its complete path, to identify a file in the network system. We regard different instances of a file that correspond to a common path name as a same file across different hosts, since we are mostly interested in system files which tend to have canonical path names exploited by malicious attacks.” (Ex. 1005 at pp. 32-33). Xie adds, “[f]or anomaly detection, we present a ‘pointillist’ method to detect similar, coincident changes to the patterns of file updates that are shared across multiple hosts... in a multi-dimensional feature space.” (*Id.*).

127. Using anomalies in files across multiple machines, Xie tracks down the outbreak of malware such as worms. Xie named the prototype system Seurat. As Xie explains, “[w]e have implemented a prototype system, called Seurat, that uses file system updates to represent host state changes for anomaly detection.” (Ex. 1005 at p. 31). “The goal of Seurat is to automatically identify anomalous events by correlating the state change events of all hosts in a network system. Hence Seurat defines an anomalous event as an unexpected state change close in time across multiple hosts in a network system.” (Ex. 1005 at p. 32).

128. Xie further discloses identifying and storing data about the correlated file attribute modifications as part of the process of identifying files that could be the sources of a malware infection and the resulting network traffic. For example, Xie explains that because its Seurat correlation tool “outputs a list of suspicious files as the cause of each alarm, system administrators can tell if the new clusters are caused by malicious intrusions.” (Ex. 1005 at p. 42). In a test environment, Xie explains that Seurat was able to detect specific files related to and indicative of an infestation of the Lion worm:

Figure 2.13 lists the **22 files selected by Seurat as the causes of the alarm**. These files provide enough hints to the administrators **to confirm the existence of the Lion worm**. Once detected, these compromised hosts as well as **the list of suspicious files can be marked for future detection**. If, in the following days, there are more hosts that are clustered together with the already infected machines, **or experience the same file updates, then we may conclude they are infected by the same attack**. (Ex. 1005 at p. 348, emphasis added).

129. As explained above, Rockwood discloses providing details on human attackers, such as physical addresses, in attacker profiles. Similarly, a POSITA would have been motivated to provide comparable details for software-based attackers, such as those disclosed by Xie: the computers on which the malware is present and the files that are indications of the attack.

130. A POSITA would have also recognized that Rockwood's network events and Xie's file events fit the well-known models of Network Intrusion Detection Systems (NIDS) and Host Intrusion Detection Systems (HIDS). HIDS systems, like Xie's teachings, focus on detections of anomalies from within the computer system itself. "A large number of IDSs employ their host OS's audit trail as the main source of input for detecting intrusions." (Mukherjee at p. 28). NIDS systems are network-oriented, either aggregating data from multiple hosts or from network traffic, like Rockwood discloses. A POSITA would have understood the evolution from HIDS to NIDS to include "more recent models [that] accommodate the monitoring of a number of hosts interconnected by a network [but]... do not monitor network traffic" as well as systems like the Network Security Model (NSM) that "is different from the IDSs discussed earlier in that it does not analyze audit trails [17, 27-29]. The NSM analyzes traffic on a broadcast LAN to detect intrusive behavior." (*Id.* at p. 33). Thus, at the time of the alleged invention, it was extremely well known in the art to use both NIDS analysis of network events and HIDS analysis of file/host events to detect intrusions. Moreover, it was also common to combine results from both NIDS and HIDS in order to overcome weaknesses found in each individual approach. For example,

“Distributed Intrusion Detection System... is an outgrowth of the NSM project [and] is designed to guard against some of NSM’s deficiencies ... [in DIDS] each host in the monitored domain is also equipped with a host monitor.” (*Id.* at 36.). Thus, a POSITA would have recognized the compatibility of Rockwood and Xie as complementary solutions.

131. Moreover, even though Xie uses file-based identification of malware, the identified malware includes network impacts. For example, Xie demonstrated these techniques in the tracking the Linux worm known as “Lion.” As Xie explains, “Now we proceed to examine the efficacy of Seurat during a real worm outbreak... and present the result with the Lion worm [86] in this experiment.” (Ex. 1005 at p. 47). Lion is a worm that generates a lot of network traffic. “Once Lion infects a system, it sets up backdoors, leaks out confidential information (/etc/passwd, /etc/shadow) via email, and scans the Internet to recruit vulnerable systems.” (*Id.*). A POSITA would have recognized that Xie is describing network events (emails, scans, etc.) that are caused by malware.

132. Xie explains that the goal of the research is to identify “the true source of attacks” observed on a network. That is, Xie is interested not just in knowing that an attack is occurring (i.e., that malicious network traffic is being generated) but the actual source of the malware generating the traffic:

We define the problem of tracing and reconstruction of arbitrary network attacks in terms of two fundamental components, Attacker Identification and Attack Reconstruction. These act as building blocks on which attack investigation can be based and attackers held accountable. Attacker Identification is the ability to accurately pinpoint the source(s) of the attack or infection. Attack Reconstruction is the process of inferring which communications carry the attack forward. This not only identifies the compromised hosts for subsequent correction, but also provides crucial information about the attack propagation that can help in precluding future attacks of a similar kind. ***The focus of our work on identifying the true source of attacks through Attacker Identification and Attack Reconstruction*** differentiates it from other efforts that seek to identify when

an attack is occurring or to reactively blunt the effect of an attack already in progress.
(Ex. 1005 at p. 61).

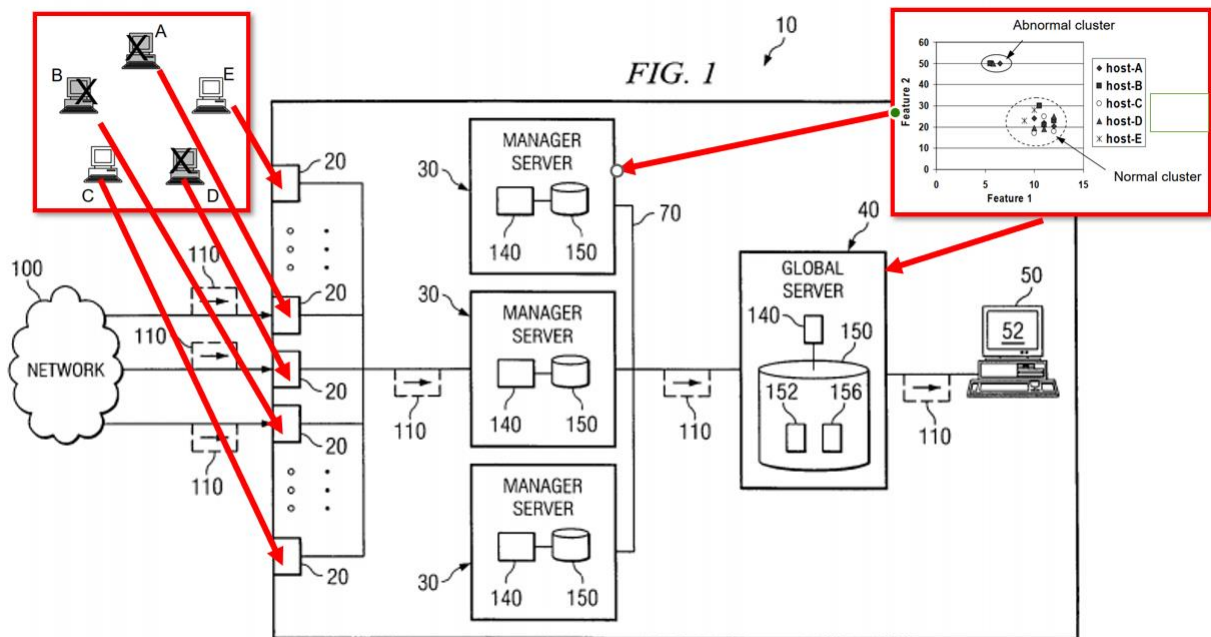
133. Thus, a POSITA would have recognized that Xie's identifying the "true source" of an attack carried out by autonomous software is analogous to Rockwood's identifying of the human attacker. As stated above, a POSITA would have been motivated to have identification for both types of attack.

134. A POSITA also would have been motivated to combine Xie with Rockwood in order to have better accuracy identifying malware. Xie explains that its disclosures are capable of detecting malicious attacks that are otherwise difficult to identify. "By exploring both the temporal and spatial locality of host state changes, our approach identifies malicious events that are hard to detect in isolation, without foreknowledge of normal changes or system-specific knowledge." (Ex. 1005 at p. 3). Xie is explicit that a global view, such as Xie's approach, is needed to identify stealthy attacks. "Meanwhile, by globally correlating events across multiple locations, it is possible to identify stealthy attacks that may not be detectable by looking only at events at individual locations." (Ex. 1005 at p. 22). And, just as importantly as finding stealthy attacks, Xie's approach also reduces false alarms. "Such approach can eliminate false alarms caused by normal activity pattern shifts at single locations, without foreknowledge of normal changes and without system-specific knowledge for rules." (Ex. 1005 at pp. 21-22).

135. To illustrate the combination of Rockwood and Xie, I have created an illustration of an example of a combined architecture using Figure 1 of Rockwood with Figure 2.1 of Xie. I have annotated Xie's additions to Rockwood in red.

136. As explained above in my overview of the reference, Rockwood discloses a series of Sensor 20 for detecting network events. But as Rockwood explains, "...sensor 20 may comprise

a **computer, server, lower-level intrusion detection system**, firewall, or any module...” (Ex. 1004 at 3:59-61, emphasis added). Thus, a POSITA would find it obvious that sensors that are computers, servers, or even other intrusion detection systems could deploy Xie’s file analysis components as illustrated below.



137. Moreover, Xie discloses a centralized correlation of data. As described in Section VIII.B., data about file events was cached at a centralized system where the data was processed to identify anomalies. As shown above, a POSITA would have combined this correlation from Xie with correlation from Rockwood, which took place at either or both of the Manager Server 30 and Global Server 40. In particular, both of these servers contained correlation engine 140 and memory modules 150 (which stored attacker profiles). Thus, a POSITA would have found it obvious to combine Xie’s correlation algorithms with Rockwood’s correlation engine 140, and Xie’s centralized processing of file event data with Rockwood’s memory modules 150.

138. A POSITA also would have expected a high chance of success in combining Rockwood and Xie. As explained above, both references work by collecting events and correlating

information across related events. As also explained above, Rockwood analyzes network events while Xie analyzes file system events. A POSITA would have understood that analyzing these two types of events can be combined together successfully. This is especially the case because both Rockwood and Xie use a central location for receiving and analyzing event data and because Xie's file system events are used, at least in part, for tracing network activities such as those caused by malicious software (e.g., worms).

139. As Rockwood explains:

Correlation engine 140 is operable to correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack. Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.

(Ex. 1004 at 4:49-58).

140. As I noted above, Rockwood discloses "correlation engine 140 [also] identifies one or more existing attacker profiles 224a in profile set 156 that have characteristics that match or are similar to attributes 410 of identified detected events 222." (Ex. 1004 at 10:27-31). In other words, Rockwood correlates the network events to identify a corresponding attacker profile. Moreover, as described in the citation above, Rockwood envisions a general correlation engine that can pull data from other inputs (e.g., "servant nodes" and "any other data"). Thus, a POSITA would have recognized that Rockwood's correlation engine is meant to be extensible and interoperable.

141. Like Rockwood, Xie's global server is also used for correlation. "Each day, each host compares the newly scanned disk snapshot against that from the previous day and generates a file update summary report. In the current prototype, all the reports are uploaded daily to a

centralized server where system administrators can monitor and correlate the file updates using the correlation module.” (Ex. 1005 at p. 40).

142. Thus, a POSITA would have been motivated to combine Rockwood and Xie. Moreover, a POSITA would have expected the combination to be successful.

B. Claim 1 is rendered obvious by Rockwood in view of Xie

143. Each of the limitations of Claim 1 of the ’224 patent is disclosed by Rockwood in view of knowledge of a POSITA and Xie as described below.

1. Limitation 1[a]

144. Limitation 1[a] of the ’224 Patent reads:

A method comprising:

145. I understand that the preamble of a patent claim is not always considered limiting. However, to the extent that the preamble of Claim 1 of the ’224 Patent is considered limiting, Rockwood and Xie disclose the preamble of claim 1. For example, Rockwood and Xie disclose a method. (See, e.g., Ex. 1004 at Title (“SYSTEM AND METHOD FOR INTRUDER TRACKING USING ADVANCED CORRELATION IN A NETWORK SECURITY SYSTEM”), Abstract (“A method for correlating event information comprises receiving event information for a plurality of detected events wherein the event information for a particular detected event comprises a plurality of attributes associated with that detected event.”); Ex. 1005 at p. 3 (“For anomaly detection, we present a ‘pointillist’ method to detect similar, coincident changes to the patterns of file updates that are shared across multiple hosts.”).) Rockwood in view of knowledge of a POSITA and Xie discloses a method for Claim 1 as described below for the remaining limitations.

2. Limitation 1[b]

146. Limitation 1[b] of the ’224 Patent reads:

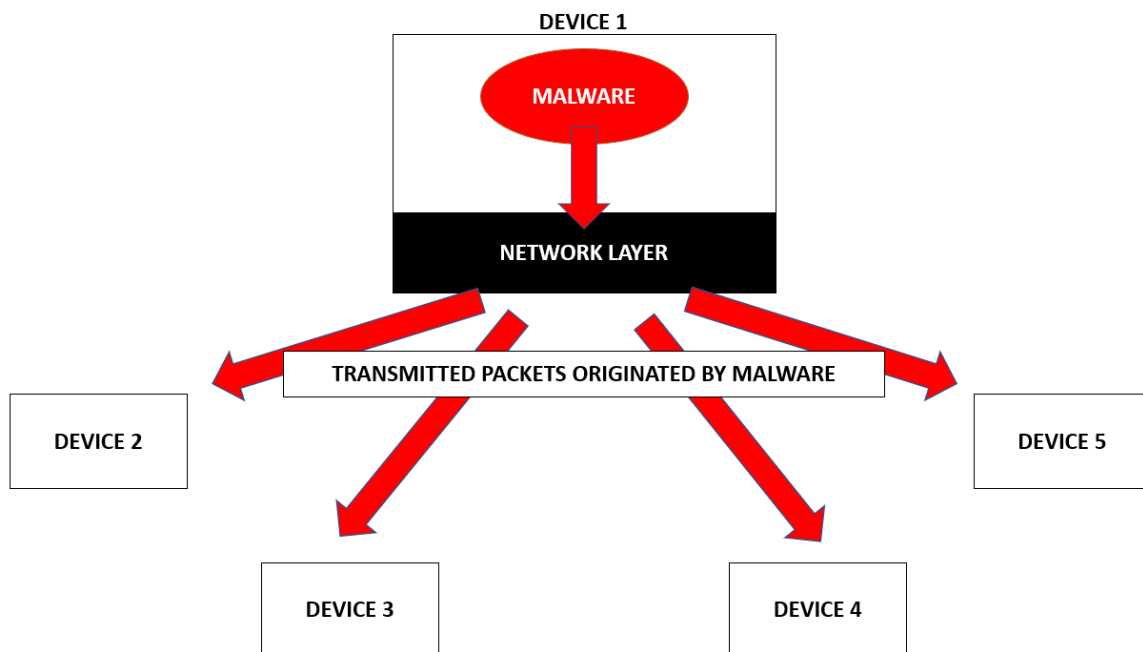
*gathering an event defining an action of a first object acting on a target,
wherein the first object is executed on a device;*

147. Rockwood discloses limitation 1[b]. Rockwood discloses “gathering an event defining an action of a first object acting on a target, wherein the first object is executed on a device”.

148. As I will explain in more detail below, Rockwood discloses network transmission sources (i.e., devices on a network) identified with a source IP address sending data to network transmission destinations (i.e., other devices on a network) with a destination IP address in an enterprise network. As I explained in my Technical Background Section V.A., network traffic from the source device is sent by the “network layer”. I will use the term “network layer” throughout this declaration to represent the lower-level components, software, protocols, and/or protocol implementations responsible for receiving and transmitting data on a network. The network layer is generally part of the “Operating System” (OS) of the device. A POSITA would have found it obvious that the network layer of the device (e.g., the device OS) manages network transmissions from the device, and network traffic sent by the device was thus sent by the network layer of the device (e.g., the device OS). In addition, the device has an IP address, which uniquely identifies that device on the network. Further, network traffic sent from the device describes its source as the IP address of the device. Therefore, because IP addresses uniquely identify computers on a network, and because network traffic sent by the device is generated by its network layer (e.g., its OS), a POSITA would have found it obvious that the IP address identifies not only the device, but also the network layer (e.g., the OS) of that device. (See Section V.A.).

149. As I described in Section V.A., a network layer is software that provides services to other software. All other software running on a device that wants to send network data does so

through the network layer. As shown in the figure below (also found in Section V.B.), malware could be one program executing on a device. It sends data (e.g., malicious data) to the network layer (e.g., the OS), which is another program executing on the device. The network layer (e.g., the OS) transmits the data (e.g., malicious data) to other devices on the malware's behalf.



150. Rockwood gathers event information about these transmissions and events from various sensors on the network. Rockwood explains:

Each sensor 20 is located at a network port that receives TCP/IP packets or other similar network communications from network 100. ***These packets and similar network communications received by sensors 20 are referred to as detected events*** 110. The data that makes up detected events 110 is referred to as event information. A detected event 110 may be an attack—that is, an unauthorized or unwanted use of the enterprise's computer network.
(Ex. 1004 at 3:36-43, emphasis added).

151. These detected events include information about the identity of the sending device (i.e., the source IP address) and the identity of the receiving device (i.e., the destination IP address).
“Each detected event 110 comprises a plurality of characteristics such as, for example, time, ***source***

IP address, and destination IP address.” (Ex. 1004 at 5:63-65, emphasis added). Rockwood illustrates in an example:

An example illustrates certain embodiments of the present invention. Sensors 20 receive three detected events 110a, 110b, and 110c. Referring to FIG. 3, event 110a occurred at 5:25 p.m. and has a source IP address of 205.252.48.163 and a destination IP address of 192.187.23.220. (Ex. 1004 at 16:62-66).

152. Rockwood further teaches that detected events may contain information about the operating system of the sending device:

System 10 detects this attack as detected event 110f, which occurred at 5:20 p.m. and **originated from a UNIX-based computer**. Detected event 110f contains a segment of code signifying that detected event 110f is an attack. (Ex. 1004 at 12:39-43, emphasis added).

153. As explained in Section V.A., a POSITA would have understood that even if other programs generate the data, network traffic is sent by the network layer executing on the device. A POSITA would have found it obvious that because the network layer generally runs as part of the OS of the device, the transmissions are transmitted by the OS. A POSITA would further understand that the OS is associated with the device’s IP address. A POSITA would recognize that “a first object” (a network layer or OS) is taught by Rockwood’s disclosure that events include at least a source IP address and information about the OS of the sending device (such as the chosen source port number) and would further recognize that the OS transmits the network data that originates from that source IP address. That is, Rockwood discloses “a first object” (a network layer or OS) “acting on a target” (i.e., sending a network transmission to a destination).

154. Moreover, Rockwood discloses recording all of the packet information about the sent data. The data includes “packet headers, packet payloads, and/or any other data” about the transmission (Ex. 1004 at 3:54-55).

Correlation engine 140 is operable to correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack. Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.
(Ex. 1004 at 4:49-58).

155. A POSITA would understand that the data in a packet also identifies the “action” (transmission of data from source to destination). That is, a POSITA would have understood that transmitting network data to a device acts on that device by causing it to receive and process the data.

156. Thus, a POSITA would understand that because the network layer (or OS) executes on the source device in order to send network transmissions, “the first object [a network layer or OS] is executed on a device”. Furthermore, Rockwood’s detected event discloses the claimed “event defining an action of a first object acting on a target”, because it contains information about the “first object” (network layer or OS) (e.g., IP address, information about the operating system of the sending device, such as the chosen source port number), the device associated with the “first object” (e.g., IP address), the “target” (destination device / IP address), and the “action of a first object acting on a target” (network transmission from the network layer or OS to a target device).

157. Alternatively (or to the extent that Patent Owner argues that a network layer or OS of a device is not “a first object”), “a first object” can be data that malware has caused to be transmitted on the network (e.g., a full or partial copy of the malware).

158. As discussed above, Rockwood explains that such transmitted data can be an attack such as malware. (See, *e.g.*, Ex. 1004 at 3:40-46 (“**The data that makes up detected events 110 is referred to as event information. A detected event 110 may be an attack**—that is, an

unauthorized or unwanted use of the enterprise's computer network. Generally, sensor 20 processes detected events 110 to detect the presence of an attack. Sensor 20 outputs at least detected events 110 . . ."); see also *id.* at 3:1-4 ("**Attacks may include viruses, Trojan horses, worms, or any other piece of code or data** that represents at least a portion of an unwanted attempt to access the protected portion of the enterprise.") (emphasis added).)

159. Rockwood discloses processing data transmitted on the network (e.g., data caused to be transmitted by malware) when detecting "detected events":

Sensor 20 may use any suitable detection technique to process and output detected events 110 and appropriate alerts. For example, sensor 20 may use algorithms, signatures, scripts, or any suitable detection or comparison technique to ***process packet headers, packet payloads, and/or any other data.***
(Ex. 1004 at 3:50-55, emphasis added).

160. The goal of Rockwood's "Intrusion detection system 10" (Ex. 1004 at 2:40-41) is detecting and identifying attacks on a network. It discloses the following:

An "attack" may be any malicious, destructive, or suspicious activity communicated from a source external and/or internal to the portion of the enterprise protected by system 10. Attacks may include viruses, Trojan horses, worms, or any other piece of code or data that represents at least a portion of an unwanted attempt to access the protected portion of the enterprise.
(Ex. 1004 at 2:65-3:4, emphasis added).

161. As discussed above, a POSITA would have understood that Rockwood's disclosures about "attacks" teach the concepts otherwise known in the art as malware. For example, as cited in Section V.B., "Malicious software, also known as malware, refers to any software designed to infiltrate or damage a computer system without the owner's informed consent. Malware may include multiple types of software, such as computer viruses, worms, trojan horses, rootkits, adware, spyware, ransomware and any other malicious and unwanted software. Once malicious software gains access to a computer system, it takes various actions resulting in a

variety of undesirable results depending on the type of malware.” (Ex. 1010 at [0002]-[0003]; see also Ex. 1004 at 3:40-46 (“***The data that makes up detected events 110 is referred to as event information. A detected event 110 may be an attack ...***”; 3:1-4 (“***Attacks may include viruses, Trojan horses, worms, or any other piece of code or data*** that represents at least a portion of an unwanted attempt to access the protected portion of the enterprise.”), emphasis added). A POSITA would further understand that malware must act on a victim “computer system” (i.e., “it takes various actions”) in order to inflict such damage (e.g., “resulting in a variety of undesirable results”) (*Id.*).

162. As I described in Section V.B., malware propagates by transmitting copies of itself on the network from one device to another device. It may do this by attaching itself to other (usually benign) programs which are then copied to other devices on the network (e.g., via email), as in the case of viruses. Alternatively, it may propagate by replicating on the network without being attached or inserted into other programs, as in the case of worms. In both cases, the malicious code causes, in one form or another, copies of itself to be transmitted in network traffic in order to cause damage to other devices on the network.

163. Thus, a POSITA would understand that Rockwood teaches monitoring the contents of network traffic in order to detect and identify attacks on the system. A POSITA would further understand that because malware propagates by replicating itself via network traffic, data that malware has caused to be transmitted on the network could represent copies of said malware. As I discussed in Technical Background Section V.B., a POSITA would also recognize that such transmissions are intended to execute transmitted malware on another device in the network in order to cause harm. Therefore, a POSITA would recognize that Rockwood teaches “a first object”

of transmissions caused by malware, where such transmissions “act[] on a target” and are “executed on a device”.

164. In either case, the events detected by Rockwood represent “a first object”. If “a first object” is the network layer of a device (e.g., its OS), Rockwood’s event includes the IP address that uniquely identifies the device (and, thus, identifies the network layer or OS). If “a first object” is network transmissions caused by malware (e.g., copies of the malware), Rockwood’s “detected event is associated with at least one data packet in an enterprise network” (e.g., said network transmission) (Ex. 1004 at 23:47-48). Thus, my discussion below with respect to events in this section is applicable to either example of “a first object”.

165. Thus, I conclude that Rockwood discloses limitation 1[b].

3. Limitation 1[c]

166. Limitation 1[c] of the ’224 Patent reads:

generating contextual state information for the event by correlating the event to an originating object of the first object;

167. Rockwood in view of Xie discloses limitation 1[c].

168. As explained in Technical Background Sections V.A. and V.B., the network layer (e.g., the OS of a device) performs the actual transmission and reception of network packets. The network layer executes on a device, and the device is identified by an IP address. Malware is harmful application software (e.g., a process) that generates unwanted network traffic and causes the network layer to transmit it. As noted in the ’224 specification, “as illustrated in at least FIG. 5B, a contextual state may include originating processes, devices, and users.” (’224 patent at 21:1-2; see also Fig. 5B). Similarly, the ’224 specification also explains that, “[i]n some embodiments, the context state 1216 may include, without limitation, Originating Process 1240 information,

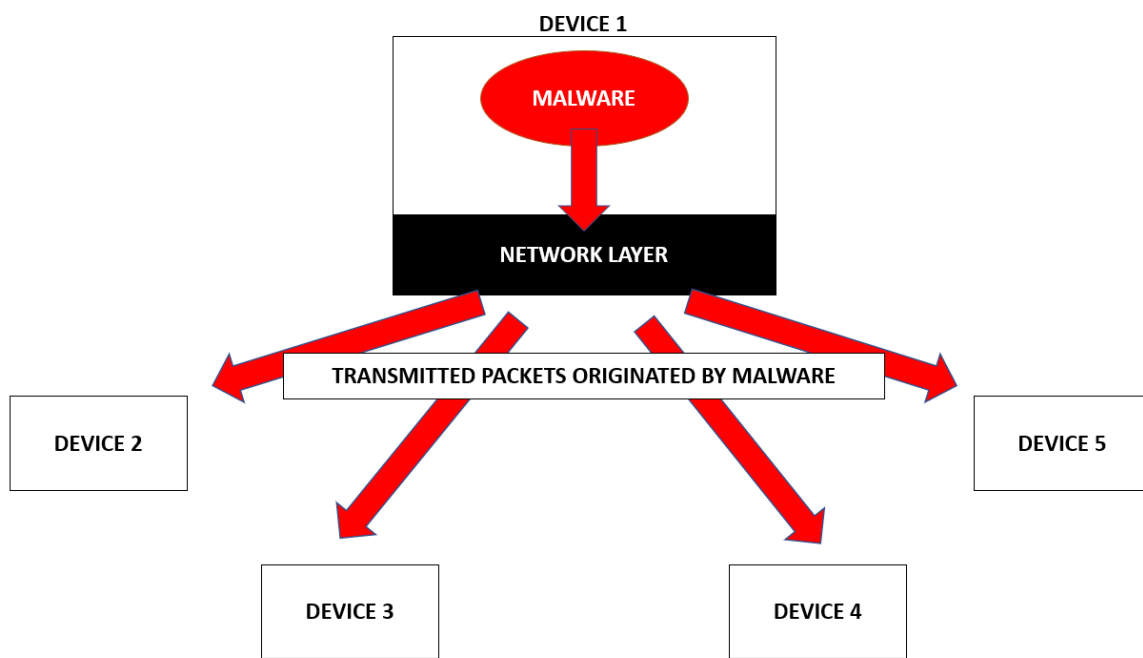
Device information 1244, User information 1248, dependent data such as, but not limited to additional meta data and/or ancillary data describing or containing detailed information about one or more objects 1252, and object info 1256.” (’224 patent at 23:2-8; see also Fig. 12 (object 1216)).

169. Rockwood acknowledges that attacks can be generated by bad software. Examples of the malicious software include viruses, trojan horses, worms, and other types of malicious code that access the system in unwanted ways. As stated by Rockwood:

In this example, intrusion detection system 10 monitors network communications on the three external ports. Based upon data received in these input streams, system 10 attempts to detect, locate, or block an attack on the enterprise. An attack may be any malicious, destructive, or suspicious activity communicated from a source external and/or internal to the portion of the enterprise protected by system 10. ***Attacks may include viruses, Trojan horses, worms, or any other piece of code or data that represents at least a portion of an unwanted attempt to access the protected portion of the enterprise.***

(Ex. 1004 at 2:61-3:4, emphasis added)

170. To emphasize the difference between the malware, which originates the data, and the network layer, which transmits the data, I reproduce one of my figures from Section V.B.



171. As shown in this illustration, the network traffic is being sent by the network layer of the device. As explained for limitation 1[b], the network layer (e.g., the OS) is the “first object” that transmits data to the “target” and Rockwood discloses identifying this event using attributes, including source and destination IP addresses. These attributes, such as IP addresses, are controlled by the network layer (e.g., the OS). Nevertheless, the network layer does not generate the data contents of the network transmission, it simply transmits whatever data is sent to it by application software (e.g., malware). In effect, the OS is somewhat like a mailroom, providing packaging, addressing, and transmission services. But the contents within the packages are given to it.

172. As I explained in Section V.B. hereinabove, malware, such as viruses and worms, replicates by sending copies of itself over the network. A POSITA would understand that in order for such replication to occur, malware causes the network layer of a device to transmit data to other machines on the network, such as a copy of the malware.

173. Thus, a POSITA would recognize that “an originating object of the first object” required by limitation 1[c] is taught by malware that causes “a first object”, the network layer of a device (e.g., the OS), to transmit data to targets on the network (other devices in the above illustration). As explained in more detail below, Rockwood in view of Xie detects malware causing network traffic and not just the network layer, OS, or device that transmits the network traffic.

174. Alternatively (or, to the extent that Patent Owner argues the network layer of a device (e.g., the OS) cannot be “a first object”), as I explained with respect to limitation 1[b], “a first object” may be network transmissions (e.g., copies of malware) contained in network traffic. As I explained in Section V.A. hereinabove, network traffic is transmitted by the network layer of

a device (e.g., the OS), and the contents of the transmission are supplied not by the network layer but by application software such as malware. As I explained in Section V.B. hereinabove, malware, such as viruses and worms, replicates by sending copies of itself on the network. For example, a virus on a device may cause the network layer to transmit copies of the virus in network traffic in order to propagate the virus to other machines on the network. As a further example, a worm on a device may cause the network layer to transmit copies of the worm in network traffic in order to propagate the worm to other machines on the network. Thus, “an originating object” required by limitation 1[c] is taught by malware that causes data (“a first object”) to be transmitted in network traffic.

175. In either case, a POSITA would have recognized “an originating object” in both interpretations to be taught by malware causing network transmissions. The remainder of my analysis of limitation 1[c] applies equally to both interpretations.

176. Rockwood also collects and correlates data about the transmission events. The correlation of network traffic data to an attack source correlates the bad traffic to the originating device and user that generated it. To do this, Rockwood correlates attack events with attacker profiles containing information about the source of the attack:

Correlation engine 140 is operable to correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack. Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.
(Ex. 1004 at 4:49-58).

177. As explained in Section VIII.A. and IX.A, Rockwood discusses attacker profiles in terms of human actors. However, Rockwood discloses attacks that are autonomous, such as

viruses and worms. Thus, and as explained in Section IX.A., a POSITA would have been motivated to combine Rockwood and Xie in order to correlate attacks to attacker profiles in terms of autonomous software attackers. A POSITA would have known that the human-centric attacker profiles, as disclosed by Rockwood, would be less useful for virus and/or worm attacks and would have been motivated to correlate data to identify the sources of these automated attacks as well. Xie discloses identifying the actual files that are related to and indicate the network infection, providing a data similar to information contained in Rockwood's attacker profile. A POSITA, as explained in Section IX.A., would have been motivated to combine this functionality.

178. A POSITA would have understood that the attacker profiles disclosed by Rockwood in view of Xie are part of "contextual state information", as they identify contextual information about the attack. As explained above, this information identifies the associated devices by IP addresses and users.

Correlation engine 140 is operable to correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack. (Ex. 1004 at 4:49-54).

Each existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise's computer network. In certain embodiments, existing attacker profile 224a includes name and background information such as, for example, last known residence or last known employer. For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks between 5:00 p.m. and 5:30 p.m. using external computers with IP addresses between 205.252.48.160 and 205.252.48.200. These characteristics are included in existing attacker profile 224a for this individual. By determining the identity of an attacker, operator 60 may be able to preempt future attacks. (Ex. 1004 at 10:8-22).

179. Rockwood in view of Xie discloses the claimed "contextual state information", because it discloses correlation of "detected events" and "attacker profiles" (which includes

information on a device (identified by an IP address) on which a network layer / OS (the “first object”) is executed, and a user of such a device containing the network layer/OS (user associated with the first object)), which is consistent with the ’224 Patent specification’s disclosures concerning a “device” and “a user”: “the contextual state including an indication of the originating object of the first object and an indication of at least one of a device on which the first object is executed and a user associated with the first object” (’224 patent, 4:50-54). Rockwood makes this clear as it describes “aggregating” the collected data in order to detect the attack: “Generally, through correlating and aggregating detected events 110, manager server 30 is capable of detecting an attack or potential attack spread among sensors 20.” (Ex. 1004 at 4:10-11).

180. For attacks caused by humans, Rockwood describes how identity details provided about an attacker, including last known address, commonly used IP addresses, etc. are associated with an attack or potential attack:

Referring back to FIG. 2, correlation engine 140 is operable to correlate detected events 110 in detected event storage 158 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to determine the likely identity of an attacker. (Ex. 1004 at 6:63-67).

Each existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise’s computer network. In certain embodiments, existing attacker profile 224a includes name and background information such as, for example, last known residence or last known employer. For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks between 5:00 p.m. and 5:30 p.m. using external computers with IP addresses between 205.252.48.160 and 205.252.48.200. These characteristics are included in existing attacker profile 224a for this individual. By determining the identity of an attacker, operator 60 may be able to preempt future attacks. (Ex. 1004 at 10:9-22).

181. As explained above, a POSITA would be motivated to combine Rockwood with Xie. Xie, for example, also discloses collecting information about file attribute modifications in

order to identify files that could be the sources of a malware infection (which discloses “generating contextual state information for the event by correlating the event to an originating object of the first object”).

We have installed the Seurat data collection tool on a number of campus office machines and a teaching cluster that are used by students daily. By default, the tool scans the attributes of all system files on a host. The attributes of a file include the file name, type, device number, permissions, size, inode number, important timestamps, and a 16-byte MD5 checksum of file content. Each day, each host compares the newly scanned disk snapshot against that from the previous day and generates a file update summary report. In the current prototype, all the reports are uploaded daily to a centralized server where system administrators can monitor and correlate the file updates using the correlation module.
(Ex. 1005 at p. 40).

Since Seurat outputs a list of suspicious files as the cause of each alarm, system administrators can tell if the new clusters [of related events] are caused by malicious intrusions.
(Ex. 1005 at p. 42, emphasis added).

182. A POSITA would have understood that Xie’s “suspicious files” are examples of the malware that cause the network transmissions in Rockwood. This is made explicit in Xie’s demonstration of tracking the Linux worm known as “Lion.” As Xie explains, “Now we proceed to examine the efficacy of Seurat during a real worm outbreak... and present the result with the Lion worm [86] in this experiment.” (Ex. 1005 at p. 47). Lion is a worm that generates a lot of network traffic. “Once Lion infects a system, it sets up backdoors, leaks out confidential information (/etc/passwd, /etc/shadow) via email, and scans the Internet to recruit vulnerable systems.” (*Id.*). A POSITA would have recognized that Xie is describing network events (e.g., emails, scans) that are caused by malware. A POSITA would have recognized that Xie’s network events correspond to Rockwood’s network events.

183. In Xie's disclosures, the tracking system identified the malicious code files that are part of the worm (and, therefore, are related to generating the malicious traffic). Xie explains that Seurat was able to detect specific files related to and indicative of an infestation of the Lion worm:

Figure 2.13 lists the **22 files selected by Seurat as the causes of the alarm**. These files provide enough hints to the administrators **to confirm the existence of the Lion worm**. Once detected, these compromised hosts as well as **the list of suspicious files can be marked for future detection**. If, in the following days, there are more hosts that are clustered together with the already infected machines, **or experience the same file updates, then we may conclude they are infected by the same attack**.
(Ex. 1005 at p. 48, emphasis added).

184. By combining Xie with Rockwood, it is possible to identify the specific files that are part of the malware infection, the computers infected, and when the malware infection started. Thus, a POSITA would have found it obvious to identify information about software attackers (e.g., malware) with equivalent levels of detail (e.g., infected files) as human attackers with physical addresses.

185. For the same reason, it would have been obvious to a POSITA to include the additional details of Xie along with the attacker profiles of Rockwood in order to communicate the source of the attack (which discloses the "originating object of the first object"). Thus, the correlation of Rockwood's detected events and attacker profile (as modified by Xie's disclosure to include information on an originating object/malware) discloses "generating contextual state information for the event by correlating the event to an originating object of the first object." As explained above, this is consistent with the disclosure of the '224 Patent ("a contextual state may include originating processes, devices, and users" ('224 Patent at 21:1-2)), as a POSITA would have understood that Xie's identification of an originating malware (e.g., Lion worm) and its processes that infect computers on a network correspond to and disclose the "originating processes" of "contextual state information."

186. A POSITA would have been motivated to combine Rockwood and Xie as I explained in section IX.A, and I incorporate this analysis into this section by reference. A POSITA would have recognized that Xie's events and event analysis techniques are compatible with Rockwood's detected events and the correlation engine as described above. A POSITA would have understood that using Xie's file system events to identify the malware causing an attack enhances Rockwood's stated goals and objectives.

187. As explained above, Rockwood explicitly uses the correlation engine to analyze network events in combination with attacker profiles in order to identify the source of the attack. Rockwood also identifies that malware can be the source of an attack. A POSITA would have been motivated to use Xie's techniques in order to identify the malware in greater detail and specificity as well as Xie's capacity for detecting stealthy malware with low false alarms.

188. Thus, I conclude that Rockwood in view of Xie discloses limitation 1[c].

4. Limitation 1[d]

189. Limitation 1[d] of the '224 Patent reads:

obtaining a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network;

190. Rockwood in view of Xie discloses limitation 1[d].

191. For clarity, I have broken this limitation down into two components. Limitation 1[d] begins as follows:

obtaining a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object,

192. Rockwood in view of Xie discloses “obtaining a global perspective for the event based on the contextual state information.” For example, Rockwood discloses correlating multiple events (detected events) over a network. And (as discussed in limitation 1[c] above), such correlated detected events are further correlated with attacker profiles (as modified by Xie to include information regarding originating objects/processes/malwares), wherein the correlation of a detected event and an attacker profile (as modified by Xie) discloses the “contextual state information.”

Correlation engine 140 is operable to ***correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack.*** Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.
(Ex. 1004 at 4:49-58, emphasis added).

193. Such correlated detected events (over a network/enterprise) that are also correlated with attacker profiles disclose “a global perspective for the event.” For example, Rockwood explains that its system “provides operator 60 with **a global view** of the state of system 10.” (See Ex. 1004 at 14:24-29 (“The preceding examples illustrate system 10 as a centralized intrusion detection system 10—that is, sensors 20 and manager servers 30 send detected events 110 to a centralized global server 40 via a common internal network 70. Global server 40 then processes detected events 110 and provides operator 60 with a global view of the state of system 10.”)). Moreover, as I explain in detail below, Rockwood-Xie system/method’s correlated detected events

that are also correlated with attacker profiles (which include information on the originating objects/malware) include information on a determination of whether or not an originating object is malware, IP address, age, and popularity, and thus disclose “a global perspective for the event.”

194. This is consistent with disclosures of the specification of the '224 Patent, which explains that “[t]he global perspective may comprise information for each object in the event and/or the contextual state 524. For example, a global perspective may be obtained for Object 2, Object 3, and Object 1 as illustrated. Accordingly, a global perspective 528 indicating the age of Object 2, the popularity of Object 2, the determination of Object 2, and/or additional IP/URL information of Object 2 may be provided.” ('224 Patent at 23:8-16). Thus, as per the specification, the information within the global perspective may be IP addresses, URLs, popularity, age, or a determination of whether or not the object is malware. (See *id.*).

195. As I explained above for limitation 1[c], Rockwood-Xie system/method’s “contextual state information” (which is part of the correlated detected events (over a network/enterprise) that are also correlated with attacker profiles (which correspond to a “global perspective,” as I discussed above)) includes an identification of an original object/malware such as a Lion worm. Thus, Rockwood-Xie system/method’s correlated detected events (over a network/enterprise) that are also correlated with attacker profiles include a determination as to whether an originating object is malware, and thus discloses a “global perspective for the event.”

196. In addition to identifying whether or not the originating object is malware, Rockwood in view of Xie also gathers data on the event including IP addresses, popularity, and age. For example, Rockwood discloses tracking IP addresses and timestamps (which represent age) in the attacker profiles associated with events (which are part of Rockwood-Xie’s “global perspective”):

Each existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise's computer network. In certain embodiments, existing attacker profile 224a includes name and background information such as, for example, last known residence or last known employer. For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks between 5:00 p.m. and 5:30 p.m. using external computers with IP addresses between 205.252.48.160 and 205.252.48.200. These characteristics are included in existing attacker profile 224a for this individual. By determining the identity of an attacker, operator 60 may be able to preempt future attacks. (Ex. 1004 at 10:8-22).

An example illustrates certain embodiments of the present invention. Sensors 20 receive three detected events 110a, 110b, and 110c. Referring to FIG. 3, event 110a occurred at 5:25 p.m. and has a source IP address of 205.252.48.163 and a destination IP address of 192.187.23.220. (Ex. 1004 at 16:62-66).

197. Similarly, Xie discloses using the popularity of files to determine whether or not a given file is abnormal on a system.

We have installed the Seurat data collection tool on a number of campus office machines and a teaching cluster that are used by students daily. By default, the tool scans the attributes of all system files on a host. The attributes of a file include the file name, type, device number, permissions, size, inode number, important timestamps, and a 16-byte MD5 checksum of file content. Each day, each host compares the newly scanned disk snapshot against that from the previous day and generates a file update summary report. ***In the current prototype, all the reports are uploaded daily to a centralized server where system administrators can monitor and correlate the file updates using the correlation module.*** (Ex. 1005 at p. 40).

Since Seurat outputs a list of suspicious files as the cause of each alarm, system administrators can tell if the new clusters [of related events] are caused by malicious intrusions. (Ex. 1005 at p. 42, emphasis added).

198. Thus, Rockwood in view of Xie discloses “obtaining a global perspective for the event.”

199. Moreover, because correlation of a detected event and an attacker profile (as modified by Xie to include information regarding an originating object/malware) corresponds to

and discloses the “contextual state information” (as I explained above for limitation 1[c]), and Rockwood-Xie system/method’s correlated multiple detected events (over a network/enterprise) that are also correlated with attacker profiles (which as I explained above disclose the “global perspective for the event”) are based on such “contextual state information” (correlation of a detected event to an attacker profile), “a global perspective for the event [is obtained] ***based on the contextual state information.***”

200. Furthermore, as I explained with reference to limitation 1[c], Rockwood in view of Xie obtains (e.g., for the correlation of a detected event and an attacker profile (with information on an originating object/malware), which is part of the global perspective) information related to suspected malware, including files, processes, devices, and/or users. A POSITA would have understood the claimed “wherein the global perspective comprises information associated with one or more of the first object and the originating object” to be taught by this collecting, by Rockwood in view of Xie, of information (e.g., files, processes, devices, users) about an “originating object” (suspected malware) and/or the “first object” (e.g., device on which the first object executes; user associated with the first object).

201. Moreover, a POSITA would have understood that Rockwood-Xie’s “global perspective comprises information associated with one or more of the first object and the originating object,” because, as discussed above, Rockwood-Xie system/method’s “global perspective” includes “contextual state information,” which (as I explained above for claim limitation 1[c]) is generated based on, and thus includes, information relating to correlation of an event to an originating object of the first object.

202. As explained with reference to limitation 1[c] and Section IX.A., a POSITA would have understood that these correlations from Rockwood and Xie were obvious to combine.

203. Limitation 1[d] continues as follows:

and wherein the global perspective relates to one or more other events related to the event across a network;

204. As I explained above, Rockwood discloses correlating multiple events over a network together in order to locate, identify, and mitigate attacks. As I explained above, a POSITA would recognize that “the global perspective” for an event is taught by Rockwood’s examining related events across the network individually or together (each of which is related to other events), together with any information derived from related events such as the determination of malware discussed above. The events are related to each other precisely because they collectively identify an attack on the network (e.g., “an attack occurring upon or within the enterprise”). Rockwood relies on this multi-event analysis for the identification.

Each sensor 20 is located at a network port that receives TCP/IP packets or other similar network communications from network 100. ***These packets and similar network communications received by sensors 20 are referred to as detected events 110.*** The data that makes up detected events 110 is referred to as event information. A detected event 110 may be an attack—that is, an unauthorized or unwanted use of the enterprise's computer network.

(Ex. 1004 at 3:36-43, emphasis added).

Correlation engine 140 is operable to ***correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack.*** Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.

(Ex. 1004 at 4:49-58, emphasis added).

205. Moreover, Rockwood explains that detected events are correlated based on similarities of events or related attributes of events. For example, Rockwood explains: “Referring back to FIG. 2, ***correlation engine 140 is operable to correlate detected events 110 in detected***

event storage 158 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to determine the likely identity of an attacker. According to certain embodiments, *correlation engine 140 correlates detected events 110 using a target event 364 and proximity limits 370. Target event 364 is an event selected or defined by operator 60 to serve as the basis for correlating detected events 110 in detected event storage 158. Target event 364 comprises one or more attributes 410 such as, for example, time, source IP address, and destination IP address.*” (Ex. 1004 at 6:63-7:7, emphasis added; see also *id.* at 8:8-10 (“Proximity limits 370 are numerical values corresponding to particular attributes 410 of target event 364.”)). Thus, a POSITA would have understood that Rockwood-Xie system/method’s correlated multiple detected events (over a network/enterprise) that are correlated with attacker profiles, which as I explained above is the “global perspective for the event,” “relates to one or more other events related to the event across a network.” (See *id.*)

206. As explained with reference to limitation 1[c], the attacker profiles of Rockwood and the file abnormalities of Xie comprise information that indicates whether the originating object is malware. As noted in the Specification of the ’224 Patent, the “global perspective” can include information from the “contextual state information”. I incorporate my analysis of 1[c] into this section by reference.

207. As explained with reference to limitation 1[c] and Section IX.A., a POSITA would have understood that relevant teachings from Rockwood and Xie were obvious to combine.

208. Thus, I conclude that Rockwood in view of Xie discloses limitation 1[d].

5. Limitation 1[e]

209. Limitation 1[e] of the ’224 Patent reads:

generating an event line comprising information relating to the event, wherein the information relates to at least one of the first object, the action of the first object, the target, and the originating object; and

210. Rockwood in view of Xie discloses limitation 1[e].

211. For clarity, I have broken this limitation down into two components.

generating an event line comprising information relating to the event

212. Rockwood discloses “generating an event line comprising information relating to the event”. For example, Rockwood discloses collecting and combining information about events, including contextual information and attacker profile information, that is subsequently displayed in a Graphical User Interface (GUI). The information related to events that is displayed in a GUI discloses “an event line comprising information relating to the event.” Moreover, the displayed information is also related to collected events and the associated attacker profile that, as explained in my analysis of 1[c] and 1[d], is or is part of both contextual state and the global perspective. This is consistent with the specification of the ’224 Patent which explains, “an event line may be assembled utilizing the event information, contextual state, and global perspective information.” (’224 Patent at 17:66-18:1). For example, Rockwood discloses displaying the attacker profile on the GUI, in connection with detected events:

Correlation engine 140 then ***correlates identified detected events 222 with one or more existing attacker profiles 224a*** in profile set 156. In particular, correlation engine 140 may identify the particular existing attacker profile 224a in profile set 156 that most closely matches identified detected events 222. ***Global server 40 sends the identified existing attacker profile 224a to console 50 as identified attacker profile 230. GUI 52 displays identified attacker profile 230 to operator 60. Operator 60 thus learns the likely identity of the individual or organization that caused identified detected events 222.***

(Ex. 1004 at 10:34-44, emphasis added).

213. The GUI disclosed by Rockwood displays relevant details for each associated event. “In one example, GUI 52 presents the relevant event information of each detected event 110 to the operator and conceals the remaining information in order to reduce visual clutter.” (Ex. 1004 at 5:48-51). The relevant information can include event information for events associated with a suspected attack:

Target event 364 is an event selected or defined by operator 60 to serve as the basis for correlating detected events 110 in detected event storage 158. Target event 364 comprises one or more attributes 410 such as, for example, time, source IP address, and destination IP address. According to certain embodiments, operator 60 selects a particular detected event 110 displayed on GUI 52 to serve as target event 364. (Ex. 1004 at 7:3-10).

Detected event 110d comprises a section of code indicating that detected event 110d is an attack. GUI 52 displays detected event 110d to operator 60. (Ex. 1004 at 7:20-22).

An example illustrates certain embodiments of the present invention. Sensors 20 receive three detected events 110a, 110b, and 110c. Referring to FIG. 3, event 110a occurred at 5:25 p.m. and has a source IP address of 205.252.48.163 and a destination IP address of 192.187.23.220. (Ex. 1004 at 16:62-66).

214. Moreover, as shown above, Rockwood’s event information such as a detected event that is displayed in a GUI includes a time stamp, which the ’224 Patent explains can be part of an event line. (See *id.*; see also ’224 Patent at 22:61-66).

215. Thus, Rockwood displays event lines comprising information relating to the event. A POSITA would have known that for data to be displayed in a GUI it must first be assembled (which is “generating an event line”). Moreover, a POSITA would have been motivated to provide these details to the operator of the GUI in order to provide information on the source of the attack,

machines that need to be cleaned, entry points of the malware, and so forth in order to better clean up the network.

wherein the information relates to at least one of the first object, the action of the first object, the target, and the originating object

216. Rockwood and Xie disclose “wherein the information [“relating to the event”] relates to at least one of the first object, the action of the first object, the target, and the originating object” for the reasons discussed above for limitations 1[b] and 1[c]. I incorporate such analyses by reference. Moreover, Rockwood explains further that the displayed information may include event information/details such as packet header information (which a POSITA would have understood to include both the source IP address and the destination IP address) and packet payloads (which can include data relating to attacks/malware): “Then, upon receiving a request from the operator, GUI 52 expands the visual representation of event information to display packet headers and payloads to the operator.” (Ex. 1004 at 5:51-53). Moreover, as I explained above, Rockwood’s event information such as detected events (which is displayed in a GUI) includes source IP and destination IP addresses: “An example illustrates certain embodiments of the present invention. Sensors 20 receive three detected events 110a, 110b, and 110c. Referring to FIG. 3, event 110a occurred at 5:25 p.m. and has a source IP address of 205.252.48.163 and a destination IP address of 192.187.23.220.” (Ex. 1004 at 16:62-66).

217. The destination IP address relates to the target. The source IP address relates to the source device, including the programs running on the device, such as the network layer (e.g., the OS). As I explained with respect to limitation 1[b], the network layer of the device represents “the first object”. Alternatively, “the first object” is taught by network traffic that malware (“the

originating object”) has caused to be transmitted (e.g., a copy of the malware). For example, such network traffic (that is an attack) is part of a detected event, which as I explained above is displayed in a GUI. (See Ex. 1004 at 3:40-46 (“The data that makes up detected events 110 is referred to as event information. A detected event 110 may be an attack ...”). In this interpretation, a POSITA would have understood that the packet payload includes this network traffic and would thus represent “the first object”. In either case, the claimed “information [“relating to the event”] relat[ing] to . . . the first object” is disclosed by Rockwood, either via the source IP address or the packet payload (e.g., data of a detected event). Moreover, as I explained above for limitation 1[b], Rockwood’s event information (which as I explained above for this limitation is part of the “event line”) includes information regarding an “action of a first object acting on a target,” so Rockwood-Xie’s “event line” includes information relating to the event such as “the action of the first object [and] the target.”

218. To the extent that Rockwood does not disclose information about the originating object, Rockwood in view of Xie does. I have already explained with respect to limitations 1[c] and 1[d] that Rockwood in view of Xie discloses identifying the originating object/malware and including that information in Rockwood’s attacker profile (which I explained above is displayed in a GUI). Thus, information relating to an event that is displayed in a GUI relates to an originating object as well.

219. A POSITA would have recognized that Xie’s information identifying the originating object has a similar purpose to Rockwood’s disclosures related to the attacker profile (e.g., to provide these details to the operator of the GUI in order to provide information on the source of the attack and so forth in order to better clean up the network). Thus, a POSITA would be motivated to include the information on software equivalent of attacker profile data (e.g.,

originating object/malware) in the event line. And, as explained in section IX.A, a POSITA would have been motivated to combine Xie with Rockwood.

220. I also noted with respect to limitation 1[b] and 1[c] that an alternative first object is the malware copied in the network traffic. All other analysis in this limitation is the same regardless of either interpretation.

221. Thus, I conclude that Rockwood in view of Xie discloses limitation 1[e].

6. Limitation 1[f]

222. Limitation 1[f] of the '224 Patent reads:

transmitting the generated event line.

223. Rockwood discloses limitation 1[f]. Specifically, Rockwood discloses transmitting event data for end users, which is a “generated event line,” to the user interface.

224. As I explained with respect to limitation 1[e], the line related to the event is assembled in order to display the data in the GUI. A POSITA would have known that displaying the data in the GUI involves a transmission of the assembled event line to the graphical display.

225. To the extent that limitation 1[f] requires transmitting the event line over a network communications interface, Rockwood discloses that the GUI could be operable on a remote system and receiving data over a network. For example, Rockwood explains that, “GUI 52 may open a secure shell (SSH) tunnel to provide additional secure communications between console 50 and the other portions of system 10.” (Ex. 1004 at 5:43-45). A POSITA would have known that SSH is a network communications software. Also, Rockwood’s Figure 1 shows event information such as a detected event that is displayed in a GUI being transmitted to console 50 containing the GUI. (See Ex. 1004 at Fig. 1.)

226. Thus, I conclude that Rockwood in view of Xie discloses limitation 1[f].

C. Claim 2 is rendered obvious by Rockwood in view of Xie

227. Each of the limitations of Claim 2 of the '224 patent, which depends from Claim 1, is disclosed by Rockwood in view of Xie as described below.

1. Limitation 2[a]

228. Limitation 2[a] reads:

The method of claim 1, wherein the information associated with the one or more of the first object and the originating object is obtained from a community database; and

229. Rockwood discloses a database of attacker profiles.

According to certain embodiments, memory module 150 comprises profile set 156. Profile set 156 refers to one or more portions of memory module 150 used for storing attacker profiles 224. According to certain embodiments, correlation engine 140 identifies one or more existing attacker profiles 224a in profile set 156 that have characteristics that match or are similar to attributes 410 of identified detected events 222. (Ex. 1004 at 10:23-31).

230. Rockwood does not disclose the source of the attacker profiles, although a POSITA would have known that attacker profiles based on community sources was common. Rockwood also discloses obtaining target events from external sources. (See, e.g., Ex. 1004 at 7:53-56: “For example, operator 60 may know from an external source that an attacker typically attacks around 3:00 p.m. using source IP address 205.252.48.166.”).

231. However, Xie explicitly teaches that malware profile information (e.g., information relating to the originating object/malware such as worms) could be obtained from Internet-wide community sources, such as public websites from defensive organizations. “We obtained the binaries and source codes of a few popular worms from public Web sites such as whitehats [111] and packetstorm [72].” (Ex. 1005 at p. 47). Xie’s disclosures include information about how files are modified by particular malware, explaining that “[w]e compile a subset of files modified by

each worm based on the descriptions from public Web sites such as Symantec [99] and F-Secure information center.” (Ex. 1005 at p. 46). As discussed in Section V.C., Internet-wide community databases were well known. Xie discloses obtaining “the information associated with the one or more of the first object and the originating object” from such community databases for the reason stated above.

232. A POSITA would have been motivated to combine Rockwood and Xie, as I described in Section X.A. With respect to this particular limitation, a POSITA would have recognized the value in having attacker profiles in combination with other malware profile information, including information used for the identification of space and time events, generated by public sources, because this would reduce the amount of time required to build out the Rockwood system and incorporate knowledge from a wider range of experts. And, as I explained in Section IX.A. above, a POSITA would have been motivated to combine Xie with Rockwood in order to obtain Xie’s capacity for detecting stealthy malware and automated malware with lower rates of false positive alarms.

233. Thus, I conclude that Rockwood in view of Xie discloses limitation 2[a].

2. Limitation 2[b]

234. Limitation 2[b] reads:

wherein the event line is transmitted to an end recipient

235. I have already explained with reference to 1[f] that Rockwood transmits an event line to an end recipient through a GUI or, alternatively, through SSH to a GUI. I incorporate my analysis of 1[f] into this section by reference.

236. Thus, I conclude that Rockwood in view of Xie discloses limitation 2[b].

237. Thus, Rockwood in view of Xie renders obvious Claim 2.

D. Claim 3 is rendered obvious by Rockwood in view of Xie

238. Claim 3 of the '224 Patent is rendered obvious by Rockwood in view of Xie. This claim, which depends from Claim 1, reads:

The method of claim 1, further comprising: obtaining an event priority based on one or more of the generated contextual state information and the global perspective, wherein the event line further includes the event priority.

239. Rockwood discloses obtaining an event priority such as high, medium, and low severity. The '224 Patent explains that severity relates to priority: “The priority level may represent a perceived severity or risk in which such an event poses.” ('224 Patent at 22:40-42). The severity, as disclosed by Rockwood, represents how closely related a detected series of events relates to a target event profile (e.g., representing attacks/threats). As Rockwood explains:

The severity of the alert may depend upon the visibility of points 360 in cluster 310 or pattern of points 360 in global n-dimensional graph 300. Referring to the preceding example, ruleset 152p in network node 910p may include a rule to generate an alert of high severity if all of points 360 in cluster 310 in global n-dimensional graph 300 have a visibility between 75 percent and 100 percent. Ruleset 152 may also include a rule to generate an alert of medium severity if all of points 360 in cluster 310 in global n-dimensional graph 300 have a visibility between 100 percent and 50 percent. Referring to FIG. 9, three points 360s in cluster 310 in global n-dimensional graph 300 correspond to detected events 110 s received from network 100 by network node 610s. Network node 610s is two levels removed from network node 610p. Based on ruleset 152p, correlation engine 140p dims points 360s in global n-dimensional graph 300 by 50 percent... According to certain embodiments, operator 60p is able to analyze the severity of a threat based on the severity of an alert. In particular, operator 60p may be most concerned with network nodes 610 that are directly connected to network node 610p. (Ex. 1004 at 21:22-61).

240. A “cluster” is defined thus by Rockwood: “A cluster 310 of points 360 refers to a concentrated grouping of points 360 in global n-dimensional graph 300.” (Ex. 1004 at 19:63-65).

“Points” in those clusters are defined thus: “Points 360 of n-dimensional graphs 340 ***are defined by attribute values 420 of detected events 110*** assigned by correlation engines 140 based on attributes 410 of detected events 110.” (Ex. 1004 at 19:37-40, emphasis added). As described with reference to limitation 1[c], “detected events” are one component Rockwood uses in generating “contextual state information”. Therefore, because “points” represent “detected events”, the analysis of a cluster pattern containing such points is or includes generated contextual state information. Because the severity (priority) information is based on analysis of a cluster pattern, the severity is based on the generated contextual state information.

241. As I also explained with respect to Claim 1, Rockwood discloses correlating multiple events over a network together in order to locate, identify, and mitigate attacks. The “global perspective” for one event relates to any of the other related events across a network individually or together (each of which is related to the other event), or any information derived from related events such as the determination of malware discussed with respect to limitation 1[d]. The severity discussed in Rockwood above uses the relationship of the events to each other to determine “visibility” of an event in n-dimensional space. Thus, the severity (event priority) is also based on the “global perspective.”

242. Rockwood also discloses that the event line includes the event priority (severity information). As explained above, “According to certain embodiments, operator 60p is able to analyze the severity of a threat based on the severity of an alert.” (Ex. 1004 at 21:57-59). As I explained with reference to limitation 1[e], Rockwood discloses displaying the event line to an operator through a Graphical User Interface (GUI). “In one example, GUI 52 presents the relevant event information of each detected event 110 to the operator and conceals the remaining information in order to reduce visual clutter.” (Ex. 1004 at 5:48-51). “Detected event 110d

comprises a section of code indicating that detected event 110d is an attack. GUI 52 displays detected event 110d to operator 60.” (Ex. 1004 at 7:20-22).

243. Further, a POSITA would have found it obvious to do so. A POSITA would have known that an operator would have needed to see the event priority, which is information relevant to the event, displayed or otherwise indicated in order to “analyze the severity of a threat based on the severity of an alert.”

244. Thus, Rockwood in view of Xie renders obvious Claim 3.

E. Claim 4 is rendered obvious by Rockwood in view of Xie

245. Claim 4 of the ’224 Patent is rendered obvious by Rockwood in view of Xie. This claim, which depends from Claim 1, reads:

The method of claim 1, wherein the event includes information identifying the first object performing the action, information identifying the action being performed, and information identifying the target upon which the act is being performed.

246. As I explained with reference to limitation 1[b], Rockwood discloses the network layer of a device (e.g., the OS), the device being identified with a source IP address, as a “first objects.” The network layer transmits data to devices with a destination IP address, which are “targets.” By sending data to them, the senders “act” on the receivers in the enterprise network. And, as I explained with reference to limitation 1[b], Rockwood’s event includes all such information.

247. Moreover, Rockwood discloses recording all of the packet information about the sent data. The data includes “packet headers, packet payloads, and/or any other data” about the transmission (Ex. 1004 at 3:54-55).

Correlation engine 140 is operable to correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224

(illustrated in FIG. 2) to identify the source of an attack. Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.
(Ex. 1004 at 4:49-58).

248. A POSITA would have understood that the data in a packet identifies the “action” (the data transmitted from source to destination). That is, a POSITA would have understood that transmitting network data to a device acts on that device by causing it to receive and process the data.

249. I also explained with respect to limitation 1[d] that a POSITA would have been motivated to combine Rockwood and Xie in order to better identify malware in attacks. As I explained, Xie, discloses creating contextual data about file attribute modifications in order to identify files that could be the sources of a malware infection.

We have installed the Seurat data collection tool on a number of campus office machines and a teaching cluster that are used by students daily. By default, the tool scans the attributes of all system files on a host. The attributes of a file include the file name, type, device number, permissions, size, inode number, important timestamps, and a 16-byte MD5 checksum of file content. Each day, each host compares the newly scanned disk snapshot against that from the previous day and generates a file update summary report. In the current prototype, all the reports are uploaded daily to a centralized server where system administrators can monitor and correlate the file updates using the correlation module.
(Ex. 1005 at p. 40).

Since Seurat outputs a list of suspicious files as the cause of each alarm, system administrators can tell if the new clusters [of related events] are caused by malicious intrusions.
(Ex. 1005 at p. 42, emphasis added).

250. A POSITA would have understood that Xie’s “suspicious files” are examples of the malware that cause the network transmissions in Rockwood. This is made explicit in Xie’s demonstration of tracking the Linux worm known as “Lion.” As Xie explains, “Now we proceed

to examine the efficacy of Seurat during a real worm outbreak... and present the result with the Lion worm [86] in this experiment.” (Ex. 1005 at p. 47). Lion is a worm that generates a lot of network traffic. “Once Lion infects a system, it sets up backdoors, leaks out confidential information (/etc/passwd, /etc/shadow) via email, and scans the Internet to recruit vulnerable systems.” (*Id.*). A POSITA would have recognized that Rockwood’s network events correspond to these exact operations.

251. In the combined Rockwood-Xie system, the originating object of an attack can also be the malware that causes the attack data on the network. The originating object is identifiable as malware, such as “Lion.” A POSITA would have been highly motivated by the knowledge that the combined Rockwood-Xie system, with its greater detail, specificity, capability to detect stealthy attacks with low false alarms, coupled with the high expectation of success makes this combination the outstanding choice.

252. As explained in previous limitations, to the extent that the Patent Owner argues that “the first object” cannot be the network layer of a device (e.g., the OS), “the first object” is taught by network traffic caused, by malware, to be transmitted (e.g., a copy of malware). In this interpretation, a POSITA would have understood Rockwood’s disclosure of “packet payloads” to include this network traffic. As explained hereinabove, a POSITA would have understood that the data in a packet also identifies the “action”, as the data is being transmitted from a source to a destination, where the data will be received by the destination device. As with previous limitations, the analysis in this section applies equally to this alternative interpretation.

253. Thus, Rockwood in view of Xie renders obvious Claim 4.

F. Claim 5 is rendered obvious by Rockwood in view of Xie

254. Each of the limitations of Claim 5 of the ’224 patent, which depends from Claim 4, is disclosed by Rockwood in view of Xie as described below.

1. Limitation 5[a]

255. Limitation 5[a] reads:

The method of claim 4, further comprising: determining whether the global perspective for the event has been cached;

256. As I explained with respect to limitation 1[d], Rockwood discloses correlating multiple events over a network together in order to locate, identify, and mitigate attacks. Rockwood teaches the claimed “global perspective” for one event by disclosing any of the other related events across a network individually or together (each of which is related to the other event), or any information derived from related events such as the determination of malware discussed with respect to limitation 1[d].

257. Rockwood also discloses a memory for storing events for processing. “In certain embodiments, correlation engine 140 is communicatively connected to a memory module 150. Memory module 150 stores detected events 110 received by sensors 20 for later processing, retrieval, or searches.” (Ex. 1004 at 5:1-4). Xie, however, discloses caching time-based correlation of related events, which also provides the global perspective, in a central server that is, for example, updated once per calendar day.

258. In more detail, Xie analyzes malware by looking for changes in files over time and “space” (i.e., a filename space across computers on a network). The file changes, however, are grouped into time periods such as a day. Xie explains that for part of the analysis “regards each individual file update status as a discrete time series signal $S_{i,n}$ on day n ... is defined as the total number of hosts that update file i on day n in the network system.” (Ex. 1005 at p. 35).

259. In Xie’s disclosure, the collective set of file updates represents the “global perspective.” As Xie explains, “As such, the dimension [of the feature vector] of the space is the

number of file names present on any machine in the network system.” (Ex. 1005 at p. 33). Just as Rockwood’s related events provide “global perspective” to each other, each event in Xie is part of the “global perspective” for any other event. That is, Xie’s “global perspective” represents an aggregate of a series of component events from different machines. However, any subset of events in Xie represents a “global perspective” for a given event. For any given event in Xie, a “global perspective” is obtained from any additional events, including a single additional event. Thus, each event in Xie represents “global perspective” information for every other event.

260. Data about each event or “global perspective” is stored across the computers that comprise the network, with each computer storing its own event information. The aggregate data is cached at a centralized server. “Each day, *each host* compares the newly scanned disk snapshot against that from the previous day and *generates a file update summary report*. In the current prototype, all *the reports are uploaded daily to a centralized server* where system administrators can monitor and correlate the file updates using the correlation module.” (Ex. 1005 at p. 40, emphasis added). As explained in section VIII.B., by caching the data in the central server, the correlation module could operate much faster than if it had to collect the data directly from each individual host. Furthermore, Xie discloses that sometimes data will not be cached, such as when updates are not made from a particular host. Xie detects this behavior: “During this period, there are a few days when a couple of hosts failed to generate or upload reports due to system failure or reconfigurations. For those small number of missing reports, we simply ignore them because they do not affect the aggregated file update patterns.” (Ex. 1005 at p. 40). Thus, the system of Xie is able to detect and determine whether reports (concerning collective set of file updates representing the “global perspective”) have been cached. Thus, Xie discloses “determining whether the global perspective for the event has been cached.”

261. I also explained with respect to limitation 1[d] that a POSITA would have been motivated to combine Rockwood and Xie in order to better identify malware in attacks. In the combined Rockwood-Xie system, the event identified in Claim 4, from which Claim 5 depends, is a network event. However, Xie's file events are "global perspective" for the network event. All other analysis remains the same.

262. I have already explained why a POSITA would have been motivated to combine Rockwood and Xie generally, in Section IX.A., and with respect to limitation 1[d] in Section IX.B. I also explained in Section IX.A. that it would have been obvious to a POSITA to combine the correlation engines of Rockwood and Ex. 1005 at the central server. I incorporate this analysis by reference. Because Xie's correlation engine (being thus combined with Rockwood's correlation engine) includes this caching element, the combination of correlation engines at the central server would also have included Xie's caching technique. Thus, it would have been obvious to a POSITA to combine the caching techniques described in Xie with the disclosures of Rockwood so that Rockwood-Xie's method "determin[es] whether the global perspective for the at least one event has been cached."

263. Thus, I conclude that Rockwood in view of Xie discloses limitation 5[a].

2. Limitation 5[b]

264. Limitation 5[b] reads:

when the global perspective for the event has been cached, obtaining the global perspective for the event based on the cached global perspective; and

265. I have already explained with respect to limitation 5[a] that Rockwood discloses a memory for storing events and information about events. I also explained that Xie discloses collecting file update events within a calendar day period. As explained, the system caches a new

set of file update events, which represent a “global perspective,” every day. Each node individually collects information about modified files that relate to other modified files on the network. Thus, each node gathers a “global perspective” on events happening on other nodes, all of which are pooled together to collectively represent the community’s “global perspective” of events happening on the network. As Xie explains, the cache (central database) uses the reports that are uploaded. In addition, Xie detects when updates are not made from hosts. In other words, when updated reports are not received from hosts, the reports have not yet been cached. “For those small number of missing reports, we simply ignore them because they do not affect the aggregated file update patterns.” (Ex. 1005 at p. 40). Otherwise, if reports (concerning collective set of file updates representing the “global perspective”) have been uploaded (i.e., cached), the central database uses (e.g., retrieves) them. Thus, Xie discloses “when the global perspective for the event has been cached, obtaining the global perspective for the event based on the cached global perspective.”

266. I have already explained why a POSITA would have been motivated to combine Rockwood and Xie generally, in Section IX.A., and with respect to limitation 1[d] in Section IX.B. I also explained in Section IX.A. that it would have been obvious to a POSITA to combine the correlation engines of Rockwood and Ex. 1005 at the central server. I incorporate this analysis by reference. Because Xie’s correlation engine (being thus combined with Rockwood’s correlation engine) includes this caching element, the combination of correlation engines at the central server would also have included Xie’s caching technique. Thus, it would have been obvious to a POSITA to combine the caching techniques described in Xie with the disclosures of Rockwood so that “when [Rockwood-Xie’s] global perspective for the event has been cached, the global perspective for the event [is obtained] based on the cached global perspective.”

267. Thus, I conclude that Rockwood in view of Xie discloses limitation 5[b].

3. Limitation 5[c]

268. Limitation 5[c] reads:

when the global perspective for the event has not been cached, obtaining the global perspective for the event from a community database.

269. As I explained with respect to limitation 5[a], Rockwood discloses a memory for storing information about events. I also explained that Xie updates the cache of global perspective every calendar day and it detects when updates are not made from a particular host. In other words, when those updates are not made from a particular host, it has not yet been cached. Xie explains that in their testing, they did not fix the missing upload. “During this period, there are a few days when a couple of hosts failed to generate or upload reports due to system failure or reconfigurations. For those small number of missing reports, we simply ignore them because they do not affect the aggregated file update patterns.” (Ex. 1005 at p. 40).

270. Thus, Xie discloses that when a report was missing, which again represents “global perspective” information that has not been cached, that other reports of sufficient number are enough to make up for it. These other reports are obtained from the nodes that collectively make up the community database. In this way, Xie discloses obtaining a global perspective for the at least one event from a community database. As explained in Section V.C., a POSITA would have understood that a community database is any database wherein the contributors and the consumers are more-or-less the same group.

271. I have already explained why a POSITA would have been motivated to combine Rockwood and Xie generally, in Section IX.A., and with respect to limitation 1[d] in Section IX.B. I also explained in Section IX.A. that it would have been obvious to a POSITA to combine the

correlation engines of Rockwood and Ex. 1005 at the central server. This also would have included this caching element of Xie in the combination. I incorporate this analysis by reference. Moreover, it would have been obvious to a POSITA to combine the caching techniques described in Xie with the disclosures of Rockwood in order to obtain the performance benefits for correlation of global data.

272. Specifically, as I explained with respect to Limitation 2[a], a POSITA would have been motivated to combine Rockwood and Xie in order to augment Rockwood's attacker profiles with additional information about malware behavior through the use of a community database. This is because such a combination would reduce the amount of time required to build out the Rockwood system and incorporate a broader range of information, and because combining this information would allow for better malware detection and lower rates of false positive alarms. Thus, it would have been obvious to a POSITA to combine the caching techniques described in Xie with the disclosures of Rockwood so that "when the global perspective for the event has not been cached, the global perspective for the event [is obtained] from a community database."

273. Thus, I conclude that Rockwood in view of Xie discloses limitation 5[c]

274. Thus, I conclude that Rockwood in view of Xie renders obvious Claim 5.

G. Claim 6 is rendered obvious by Rockwood in view of Xie

275. Each of the limitations of Claim 6 of the '224 patent, which depends from Claim 5, is disclosed by Rockwood in view of Xie as described below.

1. Limitation 6[a]

276. Limitation 6[a] reads:

The method of claim 5, further comprising: determining whether the cached global perspective for the event has expired, based on a time-to-live value associated with the cached global perspective; and

277. I have already explained with respect to limitation 5[b] that Rockwood discloses a memory for storing information about events, which discloses "global perspective" for other events. I also explained that Xie discloses collecting file update events within a calendar day period. As explained, the system caches a new set of file update events, which is a "global perspective," every day. A POSITA would have understood that the collected update events have a 1-day TTL (time-to-live value) and the system determines if the TTL has been exceeded each new calendar day. Xie is explicit that the system determines if an update is not received for a given day: "[d]uring this [testing] period, there are a few days when a couple of hosts failed to generate or upload reports due to system failure or reconfigurations." (Ex. 1005 at p. 40). As I explained above for Claim 5, it would have been obvious to a POSITA to combine the caching techniques described in Xie with the disclosures of Rockwood. Thus, it would have been obvious to combine the caching techniques described in Xie with the disclosures of Rockwood so that the Rockwood-Xie method "determin[es] whether the cached global perspective for the event has expired, based on a time-to-live value associated with the cached global perspective."

278. Thus, I conclude that Rockwood in view of Xie discloses limitation 6[a].

2. Limitation 6[b]

279. Limitation 6[b] reads:

when the global perspective for the event has expired, obtaining the global perspective for the event from the community database.

280. I have already explained with respect to limitation 5[a] that Rockwood discloses a memory (e.g., attached to a correlation engine; see, e.g., Ex. 1004 at 5:1-15: "correlation engine 140 is communicatively connected to a memory module 150.") for storing information about events, which discloses "global perspective" for other events. I also explained that Xie discloses

collecting file update events within a calendar day period. As explained, the system caches a new set of file update events, which represents a “global perspective,” every day. I incorporate my analysis of this limitation into this section by reference. Thus, when the events expire (i.e., after one day) new events, also part of the global perspective, are uploaded from the individual hosts (collectively the “community database”). As explained in Section V.C., a POSITA would have understood that a community database is any database wherein the contributors and the consumers are more-or-less the same group.

281. As I explained above, each node individually collects information about modified files that relate to other modified files on the network. Thus, each node gathers a global perspective on events happening on other nodes, all of which is pooled together to collectively represent the community’s global perspective of events happening on the network. The aggregated information is sufficient that a loss of data from a small number of nodes has minimal impact on the detection of malware. As Xie explains, “For those small number of missing reports, we simply ignore them because they do not affect the aggregated file update patterns.” (Ex. 1005 at p. 40).

282. As I have explained previously with respect to limitations 2[b] and 5[c], a POSITA would have been motivated to combine Rockwood and Xie in order to augment Rockwood’s information with Xie’s information through the use of a community database, because such a combination would reduce the amount of time required to build out the Rockwood system, would incorporate a broader range of information, and would allow for better malware detection and lower rates of false positive alarms. Thus, and as I explained above for Claim 5, it would have been obvious to a POSITA to combine the caching techniques described in Xie with the disclosures of Rockwood. Thus, it would have been obvious to combine the caching techniques described in Xie with the disclosures of Rockwood so that the Rockwood-Xie method, “when the global

perspective for the event has expired, obtain[s] the global perspective for the event from the community database.”

283. Thus, I conclude that Rockwood in view of Xie discloses limitation 6[b].

284. Thus, I conclude that Rockwood in view of Xie renders obvious Claim 6.

H. Claim 7 is rendered obvious by Rockwood in view of Xie

285. Claim 7 of the '224 Patent is rendered obvious by Rockwood in view of Xie. This claim, which depends on Claim 4, reads:

The method of claim 4, further comprising: heuristically displaying the information identifying the first object performing the action, the information identifying the action being performed, the information identifying the target upon which the act is being performed, and information associated with the global perspective according to one or more of: a location associated with the first object; at least one characteristic of an outbound connection associated with the first object; a dropped executable file; an age of the first object; and a number of concurrent connections associated with the first object.

286. As I have already explained with respect to limitation 1[e] and Claim 4, Rockwood discloses displaying information related to collected events and the associated attacker profile in a Graphical User Interface (GUI). The GUI disclosed by Rockwood displays relevant details for each associated network event including the sender (either the first object performing the action or the sender of first object performing the action), the data transmitted itself (information identifying the action being performed), and the receiver (information identifying the target upon which the act is being performed):

Target event 364 is an event selected or defined by operator 60 to serve as the basis for correlating detected events 110 in detected event storage 158. Target event 364 comprises one or more attributes 410 such as, for example,

time, source IP address, and destination IP address. According to certain embodiments, operator 60 selects a particular detected event 110 displayed on GUI 52 to serve as target event 364. (Ex. 1004 at 7:3-10).

Then, upon receiving a request from the operator, GUI 52 expands the visual representation of event information to display packet headers and payloads to the operator. (Ex. 1004 at 5:51-53).

Detected event 110d comprises a section of code indicating that detected event 110d is an attack. GUI 52 displays detected event 110d to operator 60. (Ex. 1004 at 7:20-22).

An example illustrates certain embodiments of the present invention. Sensors 20 receive three detected events 110a, 110b, and 110c. Referring to FIG. 3, event 110a occurred at 5:25 p.m. and has a source IP address of 205.252.48.163 and a destination IP address of 192.187.23.220. (Ex. 1004 at 16:62-66).

287. Moreover, as I explained above for Claim 4, Rockwood-Xie method's event information includes "the information identifying the first object performing the action, the information identifying the action being performed, the information identifying the target upon which the act is being performed," and as I explained above for Claim 1, such event information is displayed in a GUI. Thus, Rockwood-Xie's method discloses displaying the above-listed information.

288. The IP address of the sender is also a network location. An IP address is a location associated with the network stack (e.g., the OS) running on a device, which is a "first object". Rockwood discloses additional location information associated with the first object that is derived from attacker profiles (as I explained above for limitation 1[e] is displayed in a GUI), which (as I explained above for limitation 1[d]) is part of or "information associated with the global perspective."

Each existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise's computer network. In certain embodiments, existing attacker profile 224a includes name and background information such as, for example, last known residence or last known employer. For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks between 5:00 p.m. and 5:30 p.m. using external computers with IP addresses between 205.252.48.160 and 205.252.48.200. These characteristics are included in existing attacker profile 224a for this individual. By determining the identity of an attacker, operator 60 may be able to preempt future attacks. (Ex. 1004 at 10:8-22).

289. Thus, Rockwood-Xie's method discloses "displaying the information identifying the first object performing the action, the information identifying the action being performed, the information identifying the target upon which the act is being performed, and information associated with the global perspective according to . . . a location associated with the first object."

290. Moreover, Rockwood describes displaying data in various forms and abstractions. For example, "GUI 52 may include multiple levels of abstraction including groupings and boundaries." (Ex. 1004 at 5:54-55). The goal, explains Rockwood, is "a graphical user interface (GUI) 52 that tailors and filters the data presented to the user." (Ex. 1004 at 5:39-40). In another example of heuristic displaying of information, "GUI 52 presents the relevant event information of each detected event 110 to the operator and conceals the remaining information in order to reduce visual clutter." (Ex. 1004 at 5:48-51). A POSITA would understand that Rockwood's disclosures ("multiple levels of abstraction", "tailors and filters the data presented", "conceals the remaining information in order to reduce visual clutter") teach ways of heuristically displaying information. Thus, a POSITA would have understood that Rockwood-Xie's method discloses "heuristically displaying" the claimed information discussed above.

291. As explained in previous limitations, to the extent that Patent Owner argues the network layer of a device (e.g., the OS) cannot be a "first object", network traffic that malware has

caused to be transmitted (e.g., copies of the malware transmitted in packets on the network) can be a “first object”, and the IP address is a characteristic of an outbound connection associated with such transmissions.

292. Thus, I conclude that Rockwood in view of Xie renders obvious Claim 7.

I. Claim 8 is rendered obvious by Rockwood in view of Xie

293. Claim 8 of the '224 Patent is rendered obvious by Rockwood in view of Xie. This claim, which depends on Claim 1, reads:

The method of claim 1, wherein the global perspective comprises information comprises [sic] at least one of: an age; a popularity; a determination as to whether the first object is malware; a determination as to whether the originating object is malware; an Internet Protocol (IP) Address; and a Uniform Resource Locator (URL).

294. Rockwood in view of Xie renders obvious Claim 8.

295. As I explained with respect to limitation 1[d], Rockwood in view of Xie discloses “obtaining a global perspective for the event” and that “the global perspective comprises information comprises [sic] at least one of: an age; a popularity; . . . ; a determination as to whether the originating object is malware; an Internet Protocol (IP) Address”. I incorporate my analysis of that limitation here by reference in its entirety.

296. For example, as I explained for limitation 1[d], Rockwood teaches the specific limitations of Claim 8 by disclosing tracking these attributes: “age” and “an Internet Protocol (IP) Address”. Rockwood uses “attacker profiles” associated with events, and it teaches that they contain IP addresses and timestamps. A POSITA would have understood a timestamp to represent the claimed “age”.

Each existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise's computer network. In certain embodiments, existing attacker profile 224a includes name and background information such as, for example, last known residence or last known employer. For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks ***between 5:00 p.m. and 5:30 p.m.*** using external computers with ***IP addresses between 205.252.48.160 and 205.252.48.200.*** These characteristics are included in existing attacker profile 224a for this individual. By determining the identity of an attacker, operator 60 may be able to preempt future attacks. (Ex. 1004 at 10:8-22, emphasis added).

297. Moreover, as I explained for limitation 1[d], Rockwood-Xie system/method's "global perspective" comprises information on "popularity" and "a determination as to whether the originating object is malware."

298. Thus, I conclude that Rockwood in view of Xie renders obvious Claim 8.

J.Claim 9 is rendered obvious by Rockwood in view of Xie

299. Each of the limitations of Claim 9 of the '224 patent, which depends from Claim 8, is disclosed by Rockwood in view of Xie as described below.

1. Limitation 9[a]

300. Limitation 9[a] reads:

The system of claim 8, wherein the sensor agent is further operable to: cache the event and one or more other events; and

301. Rockwood discloses this limitation.

302. As I will discuss in more detail with respect to Claim 10, Rockwood teaches a "correlation engine" as part of its "manager server" or "global server", which correlation engine discloses "a sensor agent". I incorporate that analysis here by reference. As explained there and below, the correlation engine is operable to perform various tasks relating to events, including processing, handling, and caching of cached events. (See Ex. 1004 at 5:10-15 ("Correlation engine 140 is operable to process event information of detected events 110 stored in memory module 150

to detect the presence of a substantially long-term or multi-staged attack that had previously gone undetected by sensors 20 and manager servers 30.”). Rockwood also explains, “correlation engine 140 is communicatively connected to a memory module 150 [which contains detected event storage 158]. Memory module 150 stores detected events 110 received by sensors 20 for later processing, retrieval, or searches.” (See Ex. 1004 at 5-1:4, Figs. 1, 2.) Moreover, Figure 2 of Rockwood shows correlation engine 140 providing detected events 110 to memory module 150 and/or detected event storage 158. (See *id.*) Thus, a POSITA would have understood (or at least would have found obvious) that the correlation engine (“sensor agent”) is operable to “cache the event and one or more other events.”

303. Rockwood discloses caching the event and one or more other events in an event storage of the correlation engine:

Referring back to FIG. 2, correlation engine 140 is operable to correlate detected events 110 ***in detected event storage 158*** to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to determine the likely identity of an attacker. (Ex. 1004 at 6:63-67, emphasis added).

304. The event storage caches events which may be later compared to a target event:

Target event 364 is an event selected or defined by operator 60 to serve as the basis for correlating detected events 110 ***in detected event storage 158***. Target event 364 comprises one or more attributes 410 such as, for example, time, source IP address, and destination IP address. According to certain embodiments, operator 60 selects a particular detected event 110 displayed on GUI 52 to serve as target event 364. (Ex. 1004 at 7:3-10, emphasis added).

305. Thus, Rockwood discloses limitation 9[a].

2. Limitation 9[b]

306. Limitation 9[b] reads:

asynchronously consolidate and normalize the event and the one or more other events.

307. Rockwood discloses limitation 9[b].

308. Rockwood discloses receiving events from many sensors into storage. A POSITA would have understood that network events arrive at different times and that any gathering (e.g., storing) of these records occurs asynchronously. Rockwood also discloses calculating, at its “correlation engine”, the N-dimensional space for each event. The N-dimensional space allows the data to be “normalized” so that it is consistent and uniform.

In one example, each manager server 30 includes a correlation engine 140 and a ruleset 152 for receiving and correlating detected events 110 from sensors 20.

(Ex. 1004 at 4:7-10).

According to certain embodiments, correlation engine 140 stores each detected event 110 in detected event storage 158 **according to attribute values 420 of that detected event 110**. Detected event storage 158 refers to a portion or portions of memory module 150 used to store detected events 110 according to attribute values 420 assigned by correlation engine 140... Referring briefly to FIG. 4, **attribute values 420 of each detected event 110 define a point 360 in n-dimensional space in detected event storage 158**. (Ex. 1004 at 6:35-48, emphasis added).

309. As discussed above for limitation 9[a], the correlation engine is operable to perform various tasks relating to events, including processing and handling of detected events. Thus, a POSITA would have understood (or at least would have found obvious) that the correlation engine (“sensor agent”) is operative to perform tasks related to events such as those discussed above that disclose “asynchronously consolidate[ing] and normalize[ing] the event and the one or more other events.”

310. Thus, I conclude that Rockwood discloses limitation 9[b].

311. Thus, I conclude that Rockwood in view of Xie renders obvious Claim 9.

K. Claim 10 is rendered obvious by Rockwood in view of Xie

312. Rockwood in view of Xie renders obvious Claim 10. Claim 10 of the '224 Patent is nearly identical to Claim 1 for many of the limitations. The changes and additions are identified below in bold and strikethrough.

Claim 1	Claim 10
[a] A method comprising:	[a] A system for providing forensic visibility into a system, comprising:
	[b] a device comprising:
	[c] a communication interface;
	[d] a processor;
	[e] data storage; and
	[f] a sensor agent stored on the data storage that is executable by the processor, wherein the sensor agent is operable to:
[b] gathering an event defining an action of a first object acting on a target, wherein the first object is executed on a device;	[g] gather an event defining an action of a first object acting on a target ~~wherein the first object is executed on a device;~~
[c] generating contextual state information for the event by correlating the event to an originating object of the first object;	[h] generate contextual state information for the event by correlating the event to an originating object of the first object;

[d] obtaining a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network;	[i] obtain a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network;
[e] generating an event line comprising information relating to the event, wherein the information relates to at least one of the first object, the action of the first object, the target, and the originating object; and	[j] generate an event line comprising information relating to the event, wherein the information relates to at least one of the first object, the action of the first object, the target, and the originating object; and
[f] transmitting the generated event line.	[k] transmit the generated event line utilizing the communication interface.

313. As I have shown, much of the language is identical or equivalent. I address the limitations that require additional explanation below. For each limitation in Claim 10 that has a corresponding limitation in Claim 1, I incorporate the analysis for said corresponding limitation from Claim 1 by reference.

314. Although the language for limitations 10[g]-10[k] includes language that is the same or similar to Claim 1, I note that limitation 10[f] requires that these limitations be performed by the sensor agent software executed on the processor. As explained in greater detail below, Rockwood discloses a manager server which can be a combination of hardware (device) and software (the sensor agent stored on the data storage and executed on the processor). Moreover, as also explained below, a manager server includes a correlation engine and other components that provide the functionality described in Claim 1. Thus, my analysis of Claim 1 is applicable to Claim 10 as the analyzed functionality is included in the manager server.

1. Limitations 10[a]

315. I understand that the preamble of a Patent claim is not always considered limiting. However, to the extent that the preamble of Claim 10 of the '224 patent is considered limiting, Rockwood discloses it. Limitation 10[a] requires “A system for providing forensic visibility into a system, comprising:”

316. Rockwood discloses a system for providing forensic visibility. As explained in Section VIII.A., Rockwood discloses identifying the activities of an attacker through a correlation engine that collects detected events and compares them against “target” events that represent known or suspected attacks. (Ex. 1004 at Abstract, 6:63-7:16). As explained below, the correlation engine is part of the software that runs on the manager server device.

1. Limitations 10[b]-10[f]

317. Limitations 10[b]-10[f] require the following:

a device including;

a communication interface;

a processor;

data storage; and

a sensor agent stored on the data storage that is executable by the processor, wherein the sensor agent is operable to:

318. Rockwood discloses a manager server that uses hardware, such as a general-purpose computer, which is a device. “Manager server 30 represents any hardware or software module that controls or monitors one or more servant nodes, such as sensor 20... According to certain embodiments, manager server 30 comprises a general-purpose personal computer... Manager server 30 may include any hardware, software, firmware, or combination thereof...” (Ex. 1004 at 4:5-7, 4:17-23).

319. Moreover, a POSITA would have known that a general-purpose computer that runs software includes a processor and a data storage. A processor is required for computation, and storage is required to hold the instructions for the processor (as well as intermediate data and output). A processor and data storage are explicitly disclosed by Rockwood with respect to the manager server. For example, Rockwood also explains that, “[a]s used in this document, the term ‘computer’ is intended to encompass a personal computer, server... **one or more processors within these or other devices...**” (Ex. 1004 at 3:30-35, emphasis added). Moreover, Rockwood also discloses that “In one example, each manager server 30 includes a correlation engine...” (Ex. 1004 at 4:7-8). “[C]orrelation engine 140 is **communicatively connected to a memory module 150**. Memory module 150 stores detected events 110 received by sensors 20 for later processing, retrieval, or searches. Memory module 150 may include any memory or database module and may take the form of volatile or non-volatile memory comprising, without limitation, magnetic media, optical media, random access memory (RAM), read-only memory (ROM), removable media, or any other suitable local or remote memory component.” (Ex. 1004 at 5:1-10).

320. Rockwood also discloses a console that contains GUI: “Console 50 may include a graphical user interface (GUI)...” (Ex. 1004 at 5:39). A POSITA would have known that a GUI includes a communications interface that transmits data from the processor to the display device. Rockwood explains that the manager server and console may be linked. “Console 50 may be, alternatively or additionally, linked to one or more manager servers 30 and/or global servers 40 without departing from the scope of this disclosure.” (Ex. 1004 at 5:34-37). A POSITA would have understood “linked” to generically include some kind of communications interface.

321. To the extent that the console GUI does not include a communications interface, Rockwood explicitly discloses that the GUI may be “remote” (e.g., on “console 50”) from other computer systems (e.g., “manager server 30”) and communicate over SSH. For example, Rockwood explains that, “GUI 52 may open a secure shell (SSH) tunnel to provide additional secure communications between console 50 and the other portions of system 10.” (Ex. 1004 at 5:43-45). A POSITA would have known that SSH is network communications software.

322. Finally, Rockwood discloses that “Manager server 30 represents any hardware or software module... In one example, each manager server 30 includes a correlation engine 140 and a ruleset 152 for receiving and correlating detected events 110 from sensors 20.” (Ex. 1004 at 4:5-9). The ’224 Patent explains that “six components, or modules, may be utilized by the sensor agent 208 to gather and process events. The sensor agent 208 may include, but is not limited to, low level system filters, a local aggregator and interpreter, a context analyzer, a global perspective module, an event priority module, and an event distributor.” (’224 Patent at 7:17-22). The manager server software module, including the correlation engine software discloses a “sensor agent stored on the

data storage that is executable by the processor.”³ For example, the correlation engine software, which is part of the manager server software, is used to perform various tasks relating to events such as processing, handling, and analyzing events, including (for example) “receiving and correlating detected events 110” (Ex. 1004 at 4:8-9) and “process[ing] event information of detected events 110 stored in memory module 150 to detect the presence of a substantially long-term or multi-staged attack” (Ex. 1004 at 5:10-15; *id.* at 4:49-67). That is (and as discussed below and for Claim 1), it collects, processes, and analyzes events in a manner discussed for Claims 1 and 10, and provides functions relating or similar to at least a local aggregator and interpreter of events, a context analyzer for the contextual information and attacker profiles, and a global perspective module. Rockwood discloses that the functions relating to those of Claim 1 (“gathering an event”, “generating contextual state information for the event by correlating...”, “obtaining a global perspective”, “generating an event line”, and “transmitting the generated event line”) are performed by a module of the manager server (e.g., the correlation engine). Specifically, Rockwood discloses the following:

Correlation engine 140 is operable to correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack. Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data. Correlation engine 140 may be written in any appropriate computer language such as, for example, C, C++, Java, Perl, and others. It will be understood by those skilled in the art that

³ Rockwood explains that correlation engine can be part of a manager server and/or global server. (See Ex. 1004 at 4:60-54 (“It will be understood by those skilled in the art that correlation engine 140 may reside locally in manager server 30, global server 40, remotely on another computer server, or distributed across servers.”). Thus, although I discuss a correlation engine in the context of a manager server here and below as an example, it should be noted that Rockwood’s disclosures regarding correlation engine (in or in the context of a global server) are also applicable to my discussions of a correlation engine and sensor agent.

correlation engine 140 may reside locally in manager server 30, global server 40, remotely on another computer server, or distributed across servers. It will be further understood that while correlation engine 140 is illustrated as a single module, the features and functionalities performed by this module may be performed by multiple modules.
(Ex. 1004 at 4:49-67, emphasis added).

323. Given that Rockwood discloses the requirements of Claim 1, and that the functions described in Claim 1 are performed by, e.g., the correlation engine (which is a component of the manager server), a POSITA would have understood that the Rockwood-Xie system discloses “a sensor agent stored on the data storage that is executable by the processor, wherein the sensor agent is operable to” perform the steps recited in Claim 1 and Claim 10, or would have been motivated to configure the correlation engine in such a manner.

324. Thus, I conclude that Rockwood in view of Xie discloses limitations 10[b]-10[f].

2. Limitation 10[g]-10[j]

325. Limitations 10[g] through 10[j] are similar to limitations 1[b] through 1[e]. I incorporate my analysis of those limitations here by reference and conclude that Rockwood in view of Xie discloses limitations 10[g] through 10[j].

3. Limitation 10[k]

326. Limitation 10[k] is similar to limitation 1[f] but adds a requirement for “utilizing the communication interface.” I have already explained with respect to limitation 1[f] that a POSITA would understand transmitting data to a GUI includes a communication interface between a processor and the display. I also explained that Rockwood also explicitly discloses that the GUI can be remote requiring transmissions over SSH, which utilizes a network communication interface.

327. Thus, I conclude that Rockwood in view of Xie discloses limitation 10[k].

328. Thus, I conclude that Rockwood in view of Xie renders obvious Claim 10.

L. Claim 11 is rendered obvious by Rockwood in view of Xie

329. Claim 11 of the '224 Patent is rendered obvious by Rockwood in view of Xie. This claim, which depends on Claim 10, reads

The system of claim 10, wherein the event is executed by the processor.

330. As explained with respect to limitations of Claim 1, events in Rockwood include the transmission of data on a network. As explained in the Technical Background Section V.A., a POSITA would have understood that low-level network software (which I refer to as a “network layer”) performs the actual transmission and reception of network packets. This network layer is software that executes on a device. Thus, the network layer on the sending device executes the sending event on a processor, and the network layer on the receiving device executes the receiving event on a processor. Furthermore, because Rockwood’s manager server (the “system” of Claim 10) contains a correlation engine, which “may use any suitable detection or comparison technique to process packet headers, packet payloads and/or any other data” (Ex. 1004 at 4:56-58), including the “detected events” received (via network packets) for processing, a POSITA would have understood that the manager server, by receiving the event, executes the receiving event on its processor (i.e., the “processor” of Claim 10).

331. Thus, I conclude that Rockwood in view of Xie renders obvious Claim 11.

M. Claim 12 is rendered obvious by Rockwood in view of Xie

332. Each of the limitations of Claim 12 of the '224 patent, which depends from Claim 10, is disclosed by Rockwood in view of Xie as described below.

1. Limitation 12[a]

333. Limitation 12[a] reads:

The system of claim 10, further comprising: a global perspective information server; and

334. Rockwood discloses a database of attacker profiles.

According to certain embodiments, memory module 150 comprises profile set 156. Profile set 156 refers to one or more portions of memory module 150 used for storing attacker profiles 224. According to certain embodiments, correlation engine 140 identifies one or more existing attacker profiles 224a in profile set 156 that have characteristics that match or are similar to attributes 410 of identified detected events 222. (Ex. 1004 at 10:23-31).

335. Rockwood further discloses a global server: “Global server 40 represents any hardware, software, firmware, or combination thereof operable to process, control, and monitor system 10 **at the highest logical level...**” (Ex. 1004 at 4:39-41, emphasis added). Moreover, as I have explained with respect to Claim 1, Rockwood discloses that Global Server 40 includes event storage 158 that stores events from around the network. (See Ex. 1004 at 10:1-45). A POSITA would have understood that the claimed “global perspective” is taught by Rockwood’s disclosures. The server holding the data discloses, therefore, a “global perspective information server”. Rockwood also discloses that the memory that holds attacker profile data could be stored remotely (e.g., on a global server): “Although FIG. 1 illustrates ruleset 152 and memory module 150 as residing internally to global server 40, ruleset 152 and memory module 150 may reside externally or at any other location or locations accessible by global server 40 or other components of system 10.” (Ex. 1004 at 5:22-26).

336. Xie also teaches that malware profile information could be obtained from internet-wide community sources, which is another example of a global perspective information server, such as public websites from defensive organizations. “We obtained the binaries and source codes of a few popular worms from public Web sites such as whitehats [111] and packetstorm [72].” (Ex. 1005 at p. 47). Xie’s disclosures include information about how files are modified by particular malware, explaining that “[w]e compile a subset of files modified by each worm based

on the descriptions from public Web sites such as Symantec [99] and F-Secure information center.” (Ex. 1005 at p. 46).

337. Moreover, as I explained for Claims 5 and 6, Rockwood-Xie’s system discloses caching/storing global perspective information. Thus, a network computer (e.g., in Rockwood-Xie’s system) that stores global perspective information discloses “a global perspective information server.”

338. A POSITA would have been motivated to combine Rockwood and Xie, as I described in Section IX.A. With respect to this particular limitation, a POSITA would have recognized the value in having Rockwood’s “attacker profiles” augmented with additional malware profile information, including information used for the identification of space and time events, generated by public sources. This would reduce the amount of time required to build out the Rockwood system and incorporate knowledge from a wider range of experts.

339. Thus, I conclude that Rockwood in view of Xie discloses limitations 12[a]

2. Limitation 12[b]

340. Limitation 12[b] reads:

an end recipient device, wherein the information associated with the one or more of the first object and the originating object is obtained from the global perspective information server and the sensor agent transmits the event line to the end recipient device

341. Rockwood discloses this limitation.

342. Rockwood discloses a console that receives information from the Global Server. For example: “Global server 40 sends to console 50 detected events 110a and 110c as identified detected events 222.” (Ex. 1004 at 9:59-61). “Global server 40 sends the identified existing attacker profile 224a to console 50 as identified attacker profile 230.” (Ex. 1004 at 10:39-41). As

I described earlier with respect to limitations of Claim 1, “detected events” and “attacker profile” contain “information associated with the one or more of the first object and the originating object”, such as timestamps, information on an attack, and IP addresses. Rockwood is explicit that the Global Server sends this information to the console. As I explained with respect to limitation 12[a], the claimed “global perspective information server” is taught by Rockwood’s Global Server.

343. In addition, as I explained with respect to Claim 10, Rockwood teaches the claimed “sensor agent” as part of its “correlation engine”. “It will be understood by those skilled in the art that ***correlation engine 140 may reside locally in manager server 30***, global server 40, remotely on another computer server, or distributed across servers.” (Ex. 1004 at 4:60-64, emphasis added). Thus, a POSITA would understand that Rockwood’s “manager server” can perform the functions of the claimed “sensor agent”.

344. Furthermore, as I have already explained with reference to 1[e], Rockwood transmits an event line to an end recipient through a GUI or network (e.g., SSH). I incorporate my analysis of 1[e] into this section by reference. The data for the “event line” is transmitted by the manager server, which Rockwood teaches may be connected to the console: “***Console 50 may be, alternatively or additionally, linked to one or more manager servers 30*** and/or global servers 40 without departing from the scope of this disclosure.” (Ex. 1004 at 5:34-37, emphasis added). Moreover, I explained for Claims 1 and 10 that “the sensor agent transmits the event line to the end recipient device.”

345. Thus, I conclude that Rockwood discloses limitations 12[b]

346. Thus, Rockwood in view of Xie renders obvious Claim 12.

N. Claims 13-16 are rendered obvious by Rockwood in view of Xie

347. Claims 13 through 16 are identical or effectively identical to Claims 4-7. In the table below, I visualize the minor differences between them. Changes in Claims 13 through 16 are bolded.

Claim 4. The method of claim 1, wherein the event includes information identifying the first object performing the action, information identifying the action being performed, and information identifying the target upon which the act is being performed.	Claim 13. The system of claim 10, wherein the event includes information identifying the first object performing the action, information identifying the action being performed, and information identifying the target upon which the act is being performed.
Claim 5. The method of claim 4, further comprising: determining whether the global perspective for the event has been cached; when the global perspective for the event has been cached, obtaining the global perspective for the event based on the cached global perspective; and when the global perspective for the event has not been cached, obtaining the global	Claim 14. The system of claim 13, wherein the sensor agent is further operable to: determine whether the global perspective for the event has been cached; when the global perspective for the event has been cached, obtain the global perspective for the event based on the cached global perspective; and when the global perspective for the event has not been

perspective for the event from a community database.	cached, obtain the global perspective for the event from a community database.
Claim 6. The method of claim 5, further comprising: determining whether the cached global perspective for the event has expired, based on a time-to-live value associated with the cached global perspective; and when the global perspective for the event has expired, obtaining the global perspective for the event from the community database.	Claim 15. The system of claim 14, wherein the sensor agent is further operable to: determine whether the cached global perspective for the event has expired, based on a time-to-live value associated with the cached global perspective; and when the global perspective for the event has expired, obtain the global perspective for the event from the community database.
Claim 7. The method of claim 4, further comprising: heuristically displaying the information identifying the first object performing the action, the information identifying the action being performed, the information identifying the target upon which the act is being performed, and information associated with the global perspective	Claim 16. The system of claim 13, wherein the end recipient device heuristically displays the information identifying the first object performing the action, the information identifying the action being performed, the information identifying the target upon which the act is being performed, and information associated with the global perspective

according to one or more of: a location associated with the first object; at least one characteristic of an outbound connection associated with the first object; a dropped executable file; an age of the first object; and a number of concurrent connections associated with the first object.	according to one or more of: a location associated with the first object; at least one characteristic of an outbound connection associated with the first object; a dropped executable file; an age of the first object; and a number of concurrent connections associated with the first object.
--	--

348. The major difference in Claims 14 and 15 are the recitations of “wherein the sensor agent is further operable to”. Rockwood in view of Xie discloses these limitations.

349. As I explained with reference to Claim 10, Rockwood in view of Xie teaches the claimed “sensor agent” with its disclosure of a “correlation engine” as part of its “manager server”. As I further explained with reference to Claim 5, it would have been obvious to a POSITA to combine the caching techniques described in Xie with the disclosures of Rockwood’s correlation engine, such that it is capable of caching a global perspective, determining whether a global perspective has been cached, and obtaining a global perspective either from a cache or from a community database. Therefore, a POSITA would have understood that Rockwood in view of Xie discloses the additional requirements of Claim 14 that “the sensor agent [i.e., the correlation engine of Rockwood-Xie] is further operable to” perform these caching functions on the global perspective, or would have been motivated to configure the correlation engine in such a manner. I incorporate my analysis of Claim 5 here by reference.

350. Furthermore, Rockwood in view of Xie discloses the additional requirements of Claim 15 that “the sensor agent is further operable to” perform operations on the global perspective “based on a time-to-live value”, as I explained with reference to Claim 6. A POSITA would have understood that Rockwood’s memory module (which is connected to a correlation engine that discloses the claimed “sensor agent”), combined with Xie’s caching techniques, supports the operations based on time-to-live values required by Claim 6, or would have been motivated to configure the correlation engine in such a manner. I incorporate my analysis of Claim 6 here by reference.

351. None of the remaining differences above have any material impact on my analysis. Accordingly, I further incorporate by reference my analysis of Claims 4 and 7. I conclude that Claims 13-16 are rendered obvious by Rockwood in view of Xie.

O. Claim 17 is rendered obvious by Rockwood in view of Xie

352. Rockwood in view of Xie renders obvious Claim 17. Claim 17 of the ’224 Patent is nearly identical to Claim 10 for many of the limitations. The changes and additions are identified below in bold and strikethrough.

Claim 10	Claim 17
[a] A system for providing forensic visibility into a system, comprising:	[a] A system for providing forensic visibility into a system, comprising:
[b] a device comprising:	[b] a first device comprising:
[c] a communication interface;	[c] a communication interface;
[d] a processor;	[d] a processor;
[e] data storage; and	[e] data storage; and

[f] a sensor agent stored on the data storage that is executable by the processor, wherein the sensor agent is operable to:	[f] a sensor agent stored on the data storage that is executable by the processor, wherein the sensor agent is operable to:
[g] gather an event defining an action of a first object acting on a target;	[g] gather an event defining an action of a first object acting on a target;
[h] generate contextual state information for the event by correlating the event to an originating object of the first object;	
[i] obtain a global perspective for the event based on the contextual state information, wherein the global perspective comprises information associated with one or more of the first object and the originating object, and wherein the global perspective relates to one or more other events related to the event across a network;	[h] determine a global perspective for the event based on ~~the~~ contextual state information for the event , wherein the global perspective relates to one or more other events related to the event across a network, and wherein the contextual state comprises an indication of an originating object of the first object and an indication of at least one of a second device on which the first object is executed and a user associated with the first object;

[j] generate an event line comprising information relating to the event, wherein the information relates to at least one of the first object, the action of the first object, the target, and the originating object; and	[i] generate an event line comprising information relating to the event, wherein the information relates to at least one of the first object, the action of the first object, the target, and the originating object; and
[k] transmit the generated event line utilizing the communication interface.	[j] transmit the generated event line utilizing the communication interface.

353. As I have shown, much of the language is identical or equivalent. For each limitation in Claim 17 that has a corresponding limitation in Claim 10, I incorporate the analysis for said corresponding limitation from Claim 10 by reference. Therefore, each limitation in Claim 17 that has a corresponding limitation in Claim 10 is disclosed by Rockwood in view of Xie based on the same analysis.

354. Furthermore, with respect to Claim 17, Rockwood teaches that functions of its system can occur on either its Manager Server or its Global Server. The following examples illustrate this interchangeability: “In one example, each manager server 30 includes a correlation engine 140...” (Ex. 1004 at 4:7-8). “According to certain embodiments, global server 40 comprises a correlation engine 140.” (Rockwood 4:48-49). “The data is correlated by manager servers 30 and/or global server 40 in accordance with rules designed to detect attacks on enterprise systems.” (Ex. 1004 at 2:45-48). “Console 50 may be, alternatively or additionally, linked to one or more manager servers 30 and/or global servers 40 without departing from the scope of this disclosure.” (Ex. 1004 at 5:34-37). “It will be understood by those skilled in the art that correlation

engine 140 may reside locally in manager server 30, global server 40, remotely on another computer server, or distributed across servers.” (Ex. 1004 at 4:60-64).

355. Only limitation 17[h] has any significant differences as described below.

1. Limitation 17[h]

356. Limitation 17[h] reads:

determine a global perspective for the event based on contextual state information for the event, wherein the global perspective relates to one or more other events related to the event across a network, and wherein the contextual state comprises an indication of an originating object of the first object and an indication of at least one of a second device on which the first object is executed and a user associated with the first object;

357. Rockwood in view of Xie discloses limitation 17[h].

358. The language of this limitation is similar to limitations 1[c] and 1[d], and I incorporate those analyses into this section by reference.

359. The second “wherein” clause of this limitation refers to “the contextual state”. However, Claim 17 is an independent claim contains no antecedent for this phrase. A POSITA would likely have interpreted “the contextual state” to refer to “contextual state **information** for the event” earlier in the limitation (emphasis added). As such, for my analysis, I will assume that “contextual state information” is the antecedent of “the contextual state”.

360. Limitation 17[h] uses the phrase “determine a global perspective” as opposed to “obtaining a global perspective” as used in Limitation 1[d]. However, the “obtaining” identified in my analysis of Limitation 1[d] demonstrates “determining”. The global perspective includes contextual information, information about whether the originating object is malware, and an

indication of the device on which the first object executes. Accordingly, my analysis for limitation 1[d] demonstrates “determin[ing] a global perspective.”

361. Limitation 17[h] also recites an indication of at least one of a second device on which the first object is executed. As explained in Limitation 1[b], the first object is the Operating System (OS) executing on the device (e.g., second device such as a device of an attacker). I also explained with reference to limitations 1[c] and 1[d] that the contextual state and associated global perspective identified the device executing the first object by IP address. “For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks between 5:00 p.m. and 5:30 p.m. using external computers with IP addresses between 205.252.48.160 and 205.252.48.200.” (Ex. 1004 at 10:9-22).

362. To the extent that Patent Owner argues that the OS cannot be the first object, the copies of the malware in the transmitted packets are alternatively the first objects. These objects execute on the targets. As I explained with reference to limitation 1[b], the events themselves, which are part of the contextual state for the event, identify the sender and receiver by IP address. The receiver IP address is an indication of the device on which the malware executes.

363. Finally, limitation 17[h] also recites a user associated with the first object. As explained previously, attacker profiles attempt to identify the precise individual associated with the attacking device.

Each existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise's computer network. In certain embodiments, existing attacker profile 224a includes name and background information such as, for example, last known residence or last known employer. For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks between 5:00 p.m. and 5:30 p.m. using external computers with IP addresses between 205.252.48.160 and 205.252.48.200. These characteristics are included in existing attacker profile 224a for this individual. By determining the identity of an attacker, operator 60 may be able to preempt future attacks.

(Ex. 1004 at 10:8-22).

364. Moreover, I explained for limitation 1[c] that Rockwood-Xie system's contextual state information includes information relating to "device on which the first object is executed" and "a user associated with the first object," and I incorporate my analysis by reference.

365. Thus, I conclude that Rockwood in view of Xie discloses limitation 17[h].

366. Thus, Rockwood in view of Xie renders obvious claim 17.

P. Claim 18 is rendered obvious by Rockwood in view of Xie

367. Each of the limitations of Claim 18 of the '224 patent, which depends from Claim 17, is disclosed by Rockwood in view of Xie as described below.

1. Limitation 18[a]

368. Limitation 18[a] reads:

The system of claim 17, further comprising a global perspective information server including: a processor;

369. Rockwood in view of Xie discloses Limitation 18[a]. Rockwood discloses global servers, which are examples of global perspective information servers, that could receive information from manager servers. Global servers are composed of hardware and software including, for example, general purpose computers. "Global server 40 represents any hardware, software, firmware, or combination thereof operable to process, control, and monitor system 10 at the highest logical level. Global server 40 may comprise a general-purpose personal computer (PC), a workstation, a Unix-based computer, a server computer, or any other suitable processing device." (Ex. 1004 at 4:39-44). A POSITA would have known that a general-purpose personal computer has a processor.

370. Moreover, Rockwood in view of Xie discloses a system comprising a global perspective information server for the reason discussed in Claim 12.

371. Thus, Rockwood discloses limitation 18[a].

2. Limitation 18[b]

372. Limitation 18[b] reads:

a communication interface;

373. Rockwood discloses limitation 18[b]. Rockwood discloses that global server communicates over a network with manager server and end users. The systems are connected over the network: “sensors 20 and manager servers 30 send detected events 110 to a centralized global server 40 via a common internal network 70...” (Ex. 1004 at 14:25-27).

374. A POSITA would have known that a server cannot receive data over a “common internal network” without a communication interface.

375. Thus, Rockwood discloses limitation 18[b].

3. Limitation 18[c]

376. Limitation 18[c] reads:

data storage;

377. Rockwood discloses limitation 18[c]. Rockwood discloses that global server could have an internal, external, or network-attached memory, which is used for data storage. “Although FIG. 1 illustrates ruleset 152 and memory module 150 as residing internally to global server 40, ruleset 152 and memory module 150 may reside externally or at any other location or locations accessible by global server 40 or other components of system 10.” (Ex. 1004 at 4:44-47).

378. Thus, Rockwood discloses limitation 18[c].

4. Limitation 18[d]

379. Limitation 18[d] reads:

a global perspective module, wherein the global perspective module is operable to:

380. Rockwood explains that “Global server 40 represents any hardware, software, firmware, or combination thereof operable to process, control, and monitor system 10 at the highest logical level. Global server 40 may comprise a general-purpose personal computer (PC), a workstation, a Unix-based computer, a server computer, or any other suitable processing device.” (Ex. 1004 at 4:38-43). A POSITA would have understood that such a global server would have contained a program or module that handles tasks of the server and that such a program/module discloses “a global perspective module.” For one example, Rockwood explains that a global server can contain a correlation engine and that “Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.” (See Ex. 1004 at 4:54-58, 4:48-67). Thus, a POSITA would have understood that a global server would have contained correlation engine or a similar program/module that discloses “a global perspective module,” and that such a program/module would have performed event-related tasks of a global server including the processes discussed in the limitations below, or would have been motivated to configure a program/module in such a manner. Moreover, Rockwood in view in Xie discloses limitation 18[d] as shown in the following requirements.

5. Limitation 18[d](i)

381. Limitation 18[d](i) reads:

receive the event line from the device;

382. Rockwood discloses limitation 18[d](i). Rockwood discloses that the global server received event information, which represents the claimed “event line”, from manager servers. The

event line incorporating the information about a given event, including a plurality of characteristics, derived from potentially multiple sources. “According to certain embodiments, alerts received from sensors 20 or generated by manager server 30 may be combined with the event information of detected events 110... Global server 40 receives detected events 110, such as from manager servers 30. Each detected event 110 comprises a plurality of characteristics such as, for example, time, source IP address, and destination IP address.” (Ex. 1004 at 4:7-11; 5:60-64).

383. I also incorporate by reference my analysis for limitations 1[e] and 1[f].

384. Thus, Rockwood discloses limitation 18[d][i].

6. Limitation 18[d](ii)

385. Limitation 18[d](ii) reads:

assemble a second event line including details associated with the event, wherein the second event line comprises information identifying the first object, information identifying the action of the first object, information identifying the target, and information identifying the originating object; and

386. Rockwood in view of Xie discloses limitation 18[d](ii).

387. For example, Rockwood discloses network software on devices identified with a source IP address, which is information identifying the first object. Rockwood also discloses sending data to devices with a destination IP address, which is information identifying the target in an enterprise network. Rockwood gathers event information and events from various sensors on the network. Rockwood explains:

Each sensor 20 is located at a network port that receives TCP/IP packets or other similar network communications from network 100. ***These packets and similar network communications received by sensors 20 are referred to as detected events*** 110. The data that makes up detected events 110 is referred to as event information. A detected event 110 may be an attack—

that is, an unauthorized or unwanted use of the enterprise's computer network.
(Ex. 1004 at 3:36-43, emphasis added).

388. These detected events include information about the identity of the sending device (i.e., the source IP address) and the identity of the receiving device (i.e., the destination IP address). “Each detected event 110 comprises a plurality of characteristics such as, for example, time, **source IP address, and destination IP address.**” (Ex. 1004 at 5:63-65, emphasis added). Rockwood illustrates in an example:

An example illustrates certain embodiments of the present invention. Global server 40 receives detected event 110a, which occurred at 5:25 p.m. and originated from source IP address 205.252.48.163. One attribute 410a of detected event 110a is time—5:25 p.m. Another attribute 410b of detected event 110a is source IP address 205.252.48.163.
(Ex. 1004 at 6:10-15).

389. A POSITA would have understood that network traffic is sent by a computer program executing on the device.

390. Rockwood also discloses that the global server, upon receiving event information from the manager server, could identify whether the network transmission event, which includes information identifying the action of the first object, is caused by malware. This identification is, therefore, information identifying the originating object.

According to certain embodiments, global server 40 comprises a correlation engine 140. Correlation engine 140 is operable to correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack. Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.
(Ex. 1004 at 4:48-58).

391. Rockwood discloses that the attack may be caused by malware.

An attack may be any malicious, destructive, or suspicious activity communicated from a source external and/or internal to the portion of the enterprise protected by system 10. Attacks may include viruses, Trojan horses, worms, or any other piece of code or data that represents at least a portion of an unwanted attempt to access the protected portion of the enterprise.

(Ex. 1004 at 2:61-3:4).

392. Although Rockwood discloses identifying the originating object (malware) using the correlating events with an attacker profile, for more guidance, a POSITA would have turned to known techniques, like Xie, that discuss identifying malware on hosts in greater specificity. Xie, for example, also discloses creating contextual data about file attribute modifications in order to identify files that could be the sources of a malware infection.

We have installed the Seurat data collection tool on a number of campus office machines and a teaching cluster that are used by students daily. By default, the tool scans the attributes of all system files on a host. The attributes of a file include the file name, type, device number, permissions, size, inode number, important timestamps, and a 16-byte MD5 checksum of file content. Each day, each host compares the newly scanned disk snapshot against that from the previous day and generates a file update summary report. In the current prototype, all the reports are uploaded daily to a centralized server where system administrators can monitor and correlate the file updates using the correlation module.

(Ex. 1005 at p. 40).

Since Seurat outputs a list of suspicious files as the cause of each alarm, system administrators can tell if the new clusters [of related events] are caused by malicious intrusions.

(Ex. 1005 at p. 42, emphasis added).

393. A POSITA would have understood that Xie's "suspicious files" are examples of the malware that cause the network transmissions in Rockwood. This is made explicit in Xie's demonstration of tracking the Linux worm known as "Lion." As Xie explains, "Now we proceed to examine the efficacy of Seurat during a real worm outbreak... and present the result with the Lion worm [86] in this experiment." (Ex. 1005 at p. 47). Lion is a worm that generates a lot of network traffic. "Once Lion infects a system, it sets up backdoors, leaks out confidential

information (/etc/passwd, /etc/shadow) via email, and scans the Internet to recruit vulnerable systems.” (*Id.*). A POSITA would have recognized that Rockwood’s network events correspond to these exact operations.

394. In Xie’s disclosures, the tracking system identified the malicious code files that are part of the worm (and, therefore, are related to generating the malicious traffic). “Figure 2.13 lists the 22 files selected by Seurat as the causes of the alarm. These files provide enough hints to the administrators to confirm the existence of the Lion worm. Once detected, these compromised hosts as well as the list of suspicious files can be marked for future detection.” (Ex. 1005 at p. 48).

395. Rockwood discloses preparing information for display. Because this is combined from a first event line, it is a second event line related to collected events. Rockwood also discloses displaying the associated attacker profile, which comprises information relating to the event, in a Graphical User Interface (GUI). The GUI disclosed by Rockwood displays relevant details for each associated event. “In one example, GUI 52 presents the relevant event information of each detected event 110 to the operator and conceals the remaining information in order to reduce visual clutter.” (Ex. 1004 at 5:48-51). The relevant information can include event information for events associated with a suspected attack:

Target event 364 is an event selected or defined by operator 60 to serve as the basis for correlating detected events 110 in detected event storage 158. Target event 364 comprises one or more attributes 410 such as, for example, time, source IP address, and destination IP address. According to certain embodiments, operator 60 selects a particular detected event 110 displayed on GUI 52 to serve as target event 364.
(Ex. 1004 at 7:3-9).

Detected event 110d comprises a section of code indicating that detected event 110d is an attack. GUI 52 displays detected event 110d to operator 60.
(Ex. 1004 at 7:20-22).

396. Rockwood explains further that the displayed information may include packet header information, which includes the source IP address, the destination IP address, and packet payloads. This information relates to the first object, which is either the OS of the sender or the copies of the malware in network packets, and the target, which is the destination. The payloads also include data from the malware and are, therefore, related to the originating object. As Rockwood explains: “Then, upon receiving a request from the operator, GUI 52 expands the visual representation of event information to display packet headers and payloads to the operator.” (Ex. 1004 at 5:51-59).

397. Rockwood also states that the GUI displays attacker profile data such as the identity of the attacker, which is the originating object: “GUI 52 displays identified attacker profile 230 to operator 60. Operator 60 thus learns the likely identity of the individual or organization that caused identified detected events 222.” (Ex. 1004 at 10:40-45). Rockwood also includes the identity of the malicious code (e.g., virus or worm) as part of the attacker profile. (See, e.g., Ex. 1004 at 3:1-4, and Claims 9, 19, and 29).

398. A POSITA would have known that for data to be displayed in a GUI it must first be assembled.

399. I also incorporate by reference my analysis for limitations 1[e] and 1[f].

400. Thus, Rockwood in view of Xie discloses limitation 18[d](ii).

7. Limitation 18[d](iii)

401. Limitation 18[d](iii) reads:

transmit the second event line utilizing the communication interface to an end recipient.

402. Rockwood in view of Xie discloses limitation 18[d](iii). I have already explained this functionality in the Rockwood-Xie combination with respect to limitations 1[f], 10[k], and 18[b]. I incorporate this analysis into this section by reference.

403. Thus, Rockwood in view of Xie discloses limitation 18[d](iii).

404. Thus, I conclude that Rockwood in view of Xie renders obvious claim 18.

Q. Claim 19 is rendered obvious by Rockwood in view of Xie

405. Claim 19 is very similar to Claim 16. In the table below, I visualize the minor differences between them. Changes or additions in Claim 19 are bolded and deletions are struck through.

Claim 16. The system of claim 13, wherein the end recipient device heuristically displays the information identifying the first object performing the action, the information identifying the action being performed, the information identifying the target upon which the act is being performed, and information associated with the global perspective according to one or more of: a location associated with the first object; at least one characteristic of an outbound connection associated with the first	Claim 19. The system of claim 17, wherein the end recipient ~~device~~ heuristically displays the information identifying the first object ~~performing the~~ ~~action~~ , the information identifying the action of the first object , the information identifying the target ~~upon which the act~~ ~~is being performed~~ , and information associated with the global perspective according to one or more of: a location associated with the first object; at least one characteristic of an outbound connection associated with the first
---	--

object; a dropped executable file; an age of the first object; and a number of concurrent connections associated with the first object.	object; a dropped executable file; an age of the first object; and a number of concurrent connections associated with the first object.
---	---

406. As shown here, the language of Claim 19 is almost identical to Claim 16. The very few changes are either deletions or functionality inconsequential. I incorporate my analysis of Claim 16 into this section by reference.

407. Thus, Rockwood in view of Xie renders obvious Claim 19.

R. Claim 20 is rendered obvious by Rockwood in view of Xie

408. Each of the limitations of Claim 20 of the '224 patent, which depends from Claim 17, is disclosed by Rockwood in view of Xie as described below.

1. Limitation 20[a]

409. Limitation 20[a] reads:

The system of claim 17, further comprising a bridge comprising: a processor;

410. Rockwood discloses Limitation 20[a]. Rockwood discloses global servers that are composed of hardware and software including, for example, general purpose computers. “Global server 40 represents any hardware, software, firmware, or combination thereof operable to process, control, and monitor system 10 at the highest logical level. Global server 40 may comprise a general-purpose personal computer (PC), a workstation, a Unix-based computer, a server computer, or any other suitable processing device.” (Ex. 1004 at 4:39-44). A POSITA would have known that a general-purpose personal computer has a processor.

411. Moreover, Rockwood discloses that the Global Server could provide attacker profile data to Manager Servers. Rockwood discloses that Manager Server may contain the correlation engine: “It will be understood by those skilled in the art that correlation engine 140 may reside locally in manager server 30,” (Ex. 1004 at 4:60-62). Rockwood also discloses that the Global Server has attacker profile data, which can be externally accessed. A POSITA would recognize this system as a “bridge”. (See, e.g., Ex. 1004 at Figure 1 (showing Global Server 40 being located between devices such as Console 50 and the rest of the network and serving as a bridge)).

According to certain embodiments, memory module 150 comprises profile set 156. Profile set 156 refers to one or more portions of memory module 150 used for storing attacker profiles 224. According to certain embodiments, correlation engine 140 identifies one or more existing attacker profiles 224a in profile set 156 that have characteristics that match or are similar to attributes 410 of identified detected events 222. (Ex. 1004 at 10:23-31).

Memory module 150 may include any memory or database module and may take the form of volatile or non-volatile memory comprising, without limitation, magnetic media, optical media, random access memory (RAM), read-only memory (ROM), removable media, or any other suitable local or **remote memory component**. (Ex. 1004 at 5:4-10, emphasis added).

412. Thus, Rockwood discloses limitation 20[a].

2. Limitation 20[b]

413. Limitation 20[b] reads:

a communication interface;

414. Rockwood discloses limitation 20[b]. Rockwood discloses that global server communicated over a network with manager server and end users. The systems are connected over the network: “sensors 20 and manager servers 30 send detected events 110 to a centralized global server 40 via a common internal network 70...” (Ex. 1004 at 14:25-27).

415. A POSITA would have known that a server cannot receive data over a “common internal network” without a communication interface.

416. Thus, Rockwood discloses limitation 20[b].

3. Limitation 20[c]

417. Limitation 20[c] reads:

a cached event store, wherein the cached event store is operable to;

418. Rockwood discloses limitation 20[c]. Rockwood discloses that global server could have an internal, external, or network-attached memory. This memory is used for storing events and, therefore, the global server with its associated memory and a relevant program/module discloses “a cached event store”. “Although FIG. 1 illustrates ruleset 152 and memory module 150 as residing internally to global server 40, ruleset 152 and memory module 150 may reside externally or at any other location or locations accessible by global server 40 or other components of system 10.” (Ex. 1004 at 4:44-47). The memory stored detected events. “According to certain embodiments, correlation engine 140 stores each detected event 110 in detected event storage 158 according to attribute values 420 of that detected event 110. Detected event storage 158 refers to a portion or portions of memory module 150 used to store detected events 110 according to attribute values 420 assigned by correlation engine 140.” (Ex. 1004 at 6:35-41). Both the Global Server and Manager Server may have a correlation engine and associated memory. “It will be understood by those skilled in the art that correlation engine 140 may reside locally in manager server 30, global server 40, remotely on another computer server, or distributed across servers.” (Ex. 1004 at 4:60-64).

419. Rockwood explains that “Global server 40 represents any hardware, software, firmware, or combination thereof operable to process, control, and monitor system 10 at the highest

logical level. Global server 40 may comprise a general-purpose personal computer (PC), a workstation, a Unix-based computer, a server computer, or any other suitable processing device.” (Ex. 1004 at 4:38-43). A POSITA would have understood that such a global server would have contained a program or module that handles tasks of the server and its components such as memory. For one example, Rockwood explains that a global server can contain a correlation engine and that “Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.” (See Ex. 1004 at 4:54-58, 4:48-67). Rockwood also explains, “correlation engine 140 is communicatively connected to a memory module 150. Memory module 150 stores detected events 110 received by sensors 20 for later processing, retrieval, or searches. Memory module 150 may include any memory or database module . . . Correlation engine 140 is operable to process event information of detected events 110 stored in memory module 150 to detect the presence of a substantially long-term or multi-staged attack that had previously gone undetected by sensors 20 and manager servers 30.” (Ex. 1004 at 5:1-15). Thus, a POSITA would have understood that a global server would have contained memory with a correlation engine or a similar or related program/module that discloses “a cached event store,” and that the global server with its associated memory and relevant program/module would have performed event-related tasks of a global server including the processes discussed in the limitations below, or would have been motivated to configure it in such a manner.

420. Thus, Rockwood discloses limitation 20[c].

4. Limitation 20[d]

421. Limitation 20[d] reads:

receive a first request for a global perspective for the event;

422. Rockwood discloses limitation 20[d]. As I explained with reference to limitation 20[a], Rockwood discloses that the Global Server could provide attacker profile data to Manager Servers. Rockwood discloses that Manager Server may contain the correlation engine: “It will be understood by those skilled in the art that correlation engine 140 may reside locally in manager server 30,” (Ex. 1004 at 4:60-62). Rockwood also discloses that the Global Server has attacker profile data, which can be externally accessed, which is a bridge.

According to certain embodiments, memory module 150 comprises profile set 156. Profile set 156 refers to one or more portions of memory module 150 used for storing attacker profiles 224. According to certain embodiments, correlation engine 140 identifies one or more existing attacker profiles 224a in profile set 156 that have characteristics that match or are similar to attributes 410 of identified detected events 222. (Ex. 1004 at 10:23-31).

Memory module 150 may include any memory or database module and may take the form of volatile or non-volatile memory comprising, without limitation, magnetic media, optical media, random access memory (RAM), read-only memory (ROM), removable media, or any other suitable local or **remote memory component**. (Ex. 1004 at 5:4-10, emphasis added).

423. A POSITA would have understood that for a Manager Server to access the remote correlation engine and memory components in the Global Server would involve a request. Thus, the Global Server receives a (first) request for event correlation data, which with correlated attacker profiles, discloses a global perspective for the at least one event. Rockwood discloses that correlation engine is designed to process communications from multiple nodes. “Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.” (Ex. 1004 at 4:54-54). Because the Global

Server (together with its associated memory and a relevant program/module) can receive such requests, it discloses this limitation of “the cached event store”.

424. Thus, Rockwood discloses limitation 20[d].

5. Limitation 20[e]

425. Limitation 20[e] reads:

receive a second request for a global perspective for another event, the second request originating from another device:

426. Rockwood discloses limitation 20[e]. Rockwood discloses that many Manager Servers are connected to a Global Server. “Global server 40 receives detected events 110, such as from manager servers 30.” (Ex. 1004 at 5:61-63). Because the Global Server (together with its associated memory and a relevant program/module) can receive such requests, it discloses this limitation of “the cached event store”. Also, I incorporate my analysis for limitation 20[d].

427. Thus, Rockwood discloses limitation 20[e].

6. Limitation 20[f]

428. Limitation 20[f] reads:

obtain a first global perspective for the event by obtaining information associated with one or more of the first object and the originating object;

429. Rockwood discloses limitation 20[f]. Rockwood discloses “obtaining information associated with one or more of the first object and the originating object” and “obtaining a global perspective for the event”.

430. As I explained with reference to limitations 1[c] and 1[d], Rockwood obtains information related to whether the source of the attack is malware, which is information associated

with the originating object. Rockwood's disclosure aggregates data from many events (e.g., network packets or transmissions) and attacker profiles to identify the source of the attack:

Correlation engine 140 is operable to correlate detected events 110 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to correlate detected events 110 with attacker profiles 224 (illustrated in FIG. 2) to identify the source of an attack.
(Ex. 1004 at 4:49-54).

An "attack" may be any malicious, destructive, or suspicious activity communicated from a source external and/or internal to the portion of the enterprise protected by system 10. Attacks may include viruses, Trojan horses, worms, or any other piece of code or data that represents at least a portion of an unwanted attempt to access the protected portion of the enterprise.
(Ex. 1004 at 2:61-3:4).

431. Additionally, I also explained that Rockwood discloses obtaining information related to the source IP address of a transmission, which is information associated with the OS of the sending device or the packets containing the copied malware, either of which is the first object:

Referring back to FIG. 2, correlation engine 140 is operable to correlate detected events 110 in detected event storage 158 to detect an attack occurring upon or within the enterprise. Correlation engine 140 is further operable to determine the likely identity of an attacker.
(Ex. 1004 at 6:63-67).

Each existing attacker profile 224a comprises a plurality of characteristics of an attacker of an enterprise's computer network. In certain embodiments, existing attacker profile 224a includes name and background information such as, for example, last known residence or last known employer. For example, a particular existing attacker profile 224a may correspond to an individual who typically launches attacks between 5:00 p.m. and 5:30 p.m. using external computers with IP addresses between 205.252.48.160 and 205.252.48.200. These characteristics are included in existing attacker profile 224a for this individual. By determining the identity of an attacker, operator 60 may be able to preempt future attacks.
(Ex. 1004 at 10:9-22).

432. Moreover, Rockwood discloses that the Global Server could provide attacker profile data to Manager Servers. Rockwood discloses that Manager Server may contain the

correlation engine: “It will be understood by those skilled in the art that correlation engine 140 may reside locally in manager server 30” (Ex. 1004 at 4:60-62). Rockwood also discloses that the Global Server has attacker profile data, which can be externally accessed. A POSITA would understand this system to be a “bridge”.

According to certain embodiments, memory module 150 comprises profile set 156. Profile set 156 refers to one or more portions of memory module 150 used for storing attacker profiles 224. According to certain embodiments, correlation engine 140 identifies one or more existing attacker profiles 224a in profile set 156 that have characteristics that match or are similar to attributes 410 of identified detected events 222. (Ex. 1004 at 10:23-31).

Memory module 150 may include any memory or database module and may take the form of volatile or non-volatile memory comprising, without limitation, magnetic media, optical media, random access memory (RAM), read-only memory (ROM), removable media, or any other suitable local or **remote memory component**. (Ex. 1004 at 5:4-10, emphasis added).

433. As I explained with reference to limitation 1[d], Rockwood-Xie system/method’s correlated detected events (over a network/enterprise) that are also correlated with attacker profiles include a determination as to whether an originating object is malware, and thus discloses a “global perspective”. The Global Server, by “obtaining information associated with one or more of the first object and the originating object”, including detected events and attacker profiles, as described above, also “obtains a first global perspective for the event”.

434. In summary, the Global Server, which contains associated memory and a correlation engine and/or a relevant program/module, receives detected events from manager servers, calculates a Global Perspective by associating an attacker profile. The Global Server also makes its correlation engine and/or memory components available to Manager Servers. This creates a first global perspective for the at least one event.

435. Thus, I conclude that Rockwood discloses limitation 20[f].

7. Limitation 20[g]

436. Limitation 20[g] reads:

obtain a second global perspective for the another event:

437. Rockwood discloses limitation 20[g]. As I have already explained, Rockwood discloses the Global Server interfacing with multiple Manager Servers. The same analysis with respect to limitation 20[f] and the first request apply equally to the second request and I incorporate my analysis of limitation 20[f] into this section by reference.

438. Thus, I conclude that Rockwood discloses limitation 20[g].

8. Limitation 20[h]

439. Limitation 20[h] reads:

provide the first global perspective to the device; and

440. Rockwood discloses limitation 20[h]. As I explained with reference to limitation 20[a], Rockwood discloses that the Global Server could provide attacker profile data to Manager Servers. Rockwood discloses that Manager Server may contain the correlation engine: “It will be understood by those skilled in the art that correlation engine 140 may reside locally in manager server 30,” (Ex. 1004 at 4:60-62). Rockwood also discloses that the Global Server has attacker profile data, which can be externally accessed. The Global Server is a “bridge.”

According to certain embodiments, memory module 150 comprises profile set 156. Profile set 156 refers to one or more portions of memory module 150 used for storing attacker profiles 224. According to certain embodiments, correlation engine 140 identifies one or more existing attacker profiles 224a in profile set 156 that have characteristics that match or are similar to attributes 410 of identified detected events 222. (Ex. 1004 at 10:23-31).

Memory module 150 may include any memory or database module and may take the form of volatile or non-volatile memory comprising, without limitation, magnetic media, optical media, random access memory (RAM),

read-only memory (ROM), removable media, or any other suitable local or **remote memory component**.
(Ex. 1004 at 5:4-10, emphasis added).

441. A POSITA would have understood that for a Manager Server to access the remote correlation engine and memory components in the Global Server would involve a response. Thus, the Global Server provides a (first) global perspective to the device. Rockwood discloses that correlation engine is designed to process communications from multiple nodes. “Correlation engine 140 may be any software or logic operable to process multiple communications from servant nodes and may use any suitable detection or comparison technique to process packet headers, packet payloads, and/or any other data.” (Ex. 1004 at 4:54-54). Because the Global Server (together with its associated memory and a relevant program/module) can send such responses, it discloses this limitation of “the cached event store”.

442. Thus, I conclude that Rockwood discloses limitation 20[h].

9. Limitation 20[i]

443. Limitation 20[i] reads:

provide the second global perspective to the another device; and

444. Rockwood discloses limitation 20[i]. As I have already explained, Rockwood discloses the Global Server interfacing with multiple Manager Servers. The same analysis with respect to limitation 20[h] and the first request apply equally to the second request and I incorporate my analysis of limitation 20[h] into this section by reference.

445. Thus, I conclude that Rockwood discloses limitation 20[i].

446. Thus, I conclude that Rockwood and Xie render obvious Claim 20.

X. THERE ARE NO OBJECTIVE INDICIA OF NONOBVIOUSNESS.

447. I am not aware of any objective indicia of non-obviousness related specifically to the '224 Patent. Indeed, I am not aware of any evidence of commercial success, long felt but unsolved need, failure of others, industry praise, or unexpected results specifically related to the '224 Patent.

XI. CONCLUSION.

448. I declare that all statements made herein are true to the best of my knowledge and that all statements made on information and belief are believed to be true, and that this Declaration is made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Executed on March 3, 2023

By: 
Seth James Nielson