

Ex. I-7 - Invalidity of U.S. Patent No. 8,635,697 against U.S. Patent Pub. No. 2010/0161795 to Deridder et al. ("Deridder")

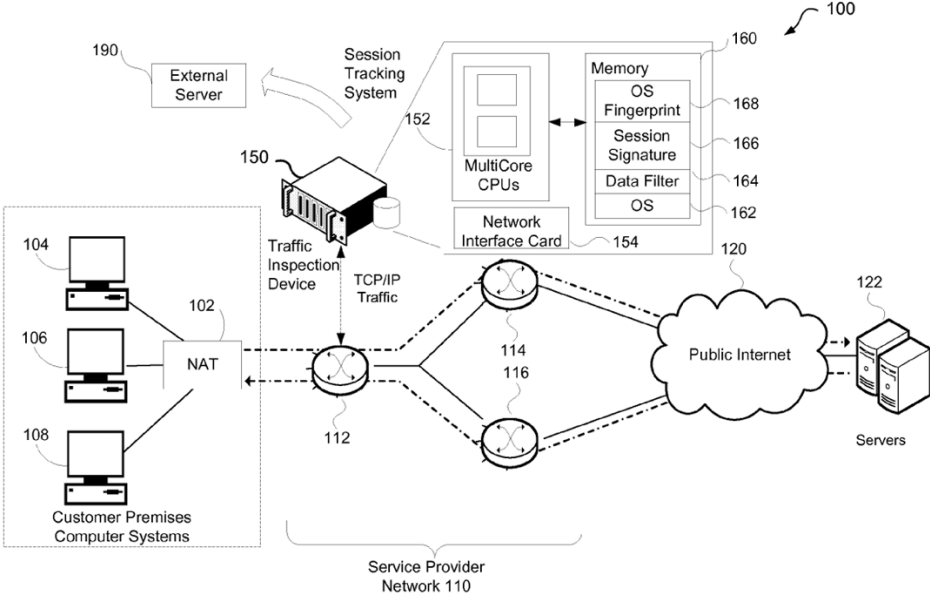
Fortinet, Inc. ("Fortinet") provides this chart subject to all reservations, objections, statements, and disclaimers set forth herein and in Fortinet's Preliminary Invalidity Contentions Cover Pleading, as well as any amendment, supplement, or modification thereof, which are incorporated herein by reference in their entirety.

On information and belief, and subject to further investigation and discovery, Deridder was filed on December 22, 2009, and published June 24, 2010, and is thus available as prior art under at least §102(a) and (e).

As illustrated in the chart below, Deridder anticipates the asserted claims of the '697 Patent. To the extent the Deridder is found not to expressly disclose certain limitations in the asserted claims, such limitations are inherent. To the extent the Deridder is found not to anticipate any asserted claims or claim elements of the '697 Patent, the reference nevertheless renders those claims or claim elements obvious under 35 U.S.C. § 103, either alone or in combination with other art identified in the cover pleading or herein. These Preliminary Invalidity Contentions are not an admission by Fortinet that the accused products, including any current or past versions of the accused products, are covered by, or infringe the asserted claims, but are based instead on the recognition that if the claims are interpreted to be broad enough to encompass the accused products, the claims must also be construed to have that same scope when considering whether they are invalid.

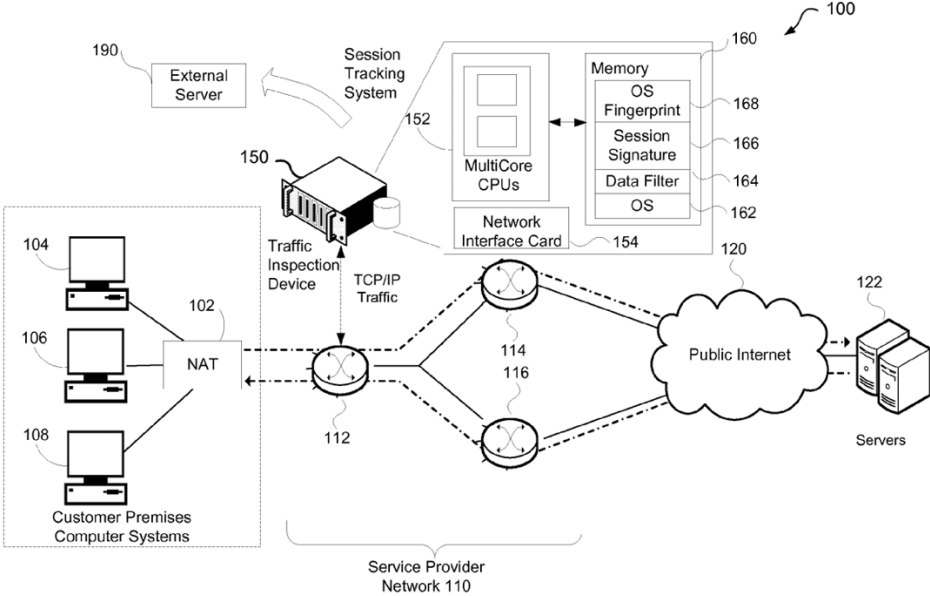
The following chart is partially based on, but is not limited by, the claim constructions implicit in Plaintiff's Infringement Contentions, to the extent that such constructions are apparent from the Infringement Contentions. Fortinet notes that in many instances, Plaintiff's Infringement Contentions fail to provide adequate notice of Plaintiff's construction of the asserted claims and fail to comply with the Court's scheduling order and other applicable rules. Fortinet does not accept the assumptions concerning the scope and meaning implicit in Plaintiff's Infringement Contentions, to the extent those assumptions are discernible, and reserves the right to challenge Plaintiff's proposed (or implied) constructions. Fortinet also reserves the right to revise and supplement these charts if and when Plaintiff is permitted to provide revised Infringement Contentions or otherwise make its positions known. To the extent that these Preliminary Invalidity Contentions rely on or otherwise embody particular constructions of terms or phrases in the asserted claims, Fortinet does not necessarily advocate any such construction as proper constructions of those terms or phrase. Fortinet also reserves the right to revise and supplement these charts after the Court construes the claims. Citations given in the chart below are merely representative of the respective elements and are not meant to be exhaustive.

Claim	Exemplary Citation from Deridder
<p>[1PRE] A method of network based malware detection in a service provider network, the method comprising:</p>	<p>Deridder discloses and/or renders obvious a method of network based malware detection in a service provider network with the following steps.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 911 1142 940">Figure 1</p> <p data-bbox="598 992 816 1021">Deridder at ¶ 21:</p> <p data-bbox="598 1029 1894 1427">"FIG. 1 is a schematic representation of a session tracking system 100 for multi-user NAT session identification and tracking in a typical operating environment. The network traffic on the service provider network 110 is passed to a traffic inspection device 150 with a high-speed network interface that is capable of receiving all network packets for computational analysis. The service provider network comprises multiple routing devices, such as 112, 114 and 116 connected to additional networks such as the public internet 120 which external servers 122 are connected to. The session tracking system 100 can be implemented as part of the traffic inspection device (e.g. data filter) 150. The inspection system 150 can be implemented at the edge of the ISP network at router 112 to ensure all traffic between a user computer system (104 to 108) is inspected. Alternatively the inspection device may be located within the service provider network 110 with packets received from the NAT device 102 being copied or forwarded to the inspection device 150. Alternatively, the traffic</p>

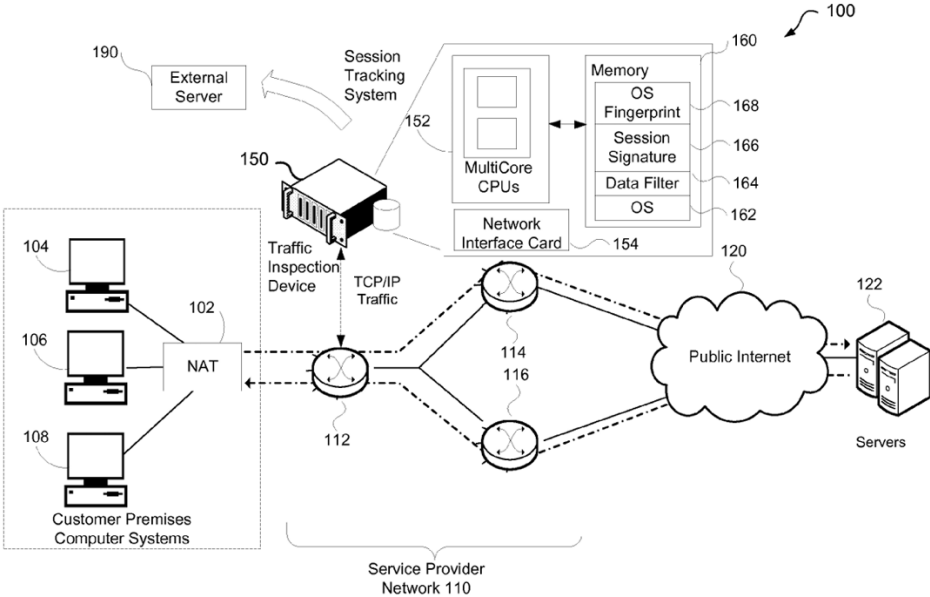
Claim	Exemplary Citation from Deridder
	<p>inspection device 150 may function inline with the network traffic. The system can optionally be operated between a first mode in which NAT session tracking is enabled and a second mode in which NAT session tracking is disabled. When NAT session tracking is disabled, all traffic to or from a single client Internet Protocol (IP) address is viewed as belonging to a single user or session. When NAT session tracking is enabled, the sessions associated with multiple users (104 to 108) behind a NAT device 102 can be detected, and their traffic can be differentiated."</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p>
[1A] receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network,	Deridder discloses and/or renders obvious receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network.

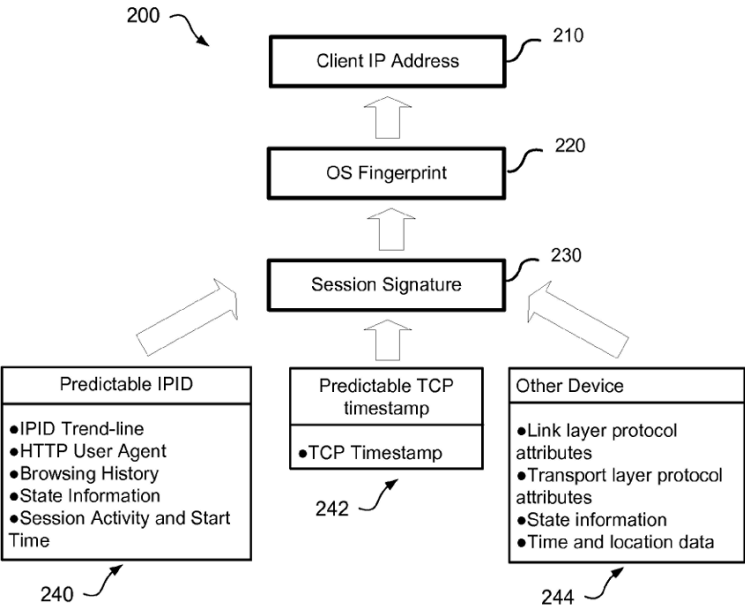
Claim	Exemplary Citation from Deridder
<p>the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;</p>	<p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 911 1142 940">Figure 1</p> <p data-bbox="598 992 816 1021">Deridder at ¶ 21:</p> <p data-bbox="598 1029 1894 1427">"FIG. 1 is a schematic representation of a session tracking system 100 for multi-user NAT session identification and tracking in a typical operating environment. The network traffic on the service provider network 110 is passed to a traffic inspection device 150 with a high-speed network interface that is capable of receiving all network packets for computational analysis. The service provider network comprises multiple routing devices, such as 112, 114 and 116 connected to additional networks such as the public internet 120 which external servers 122 are connected to. The session tracking system 100 can be implemented as part of the traffic inspection device (e.g. data filter) 150. The inspection system 150 can be implemented at the edge of the ISP network at router 112 to ensure all traffic between a user computer system (104 to 108) is inspected. Alternatively the inspection device may be located within the service provider network 110 with packets received from the NAT device 102 being copied or forwarded to the inspection device 150. Alternatively, the traffic</p>

Claim	Exemplary Citation from Deridder
	<p>inspection device 150 may function inline with the network traffic. The system can optionally be operated between a first mode in which NAT session tracking is enabled and a second mode in which NAT session tracking is disabled. When NAT session tracking is disabled, all traffic to or from a single client Internet Protocol (IP) address is viewed as belonging to a single user or session. When NAT session tracking is enabled, the sessions associated with multiple users (104 to 108) behind a NAT device 102 can be detected, and their traffic can be differentiated."</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p>
<p>[1B] determining an operating system identifier (OS ID) associated with the TCP session and the computing device;</p>	<p>Deridder discloses and/or renders obvious determining an operating system identifier (OS ID) associated with the TCP session and the computing device.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data</p>

Claim	Exemplary Citation from Deridder
	<p>packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4:</p> <p>"The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

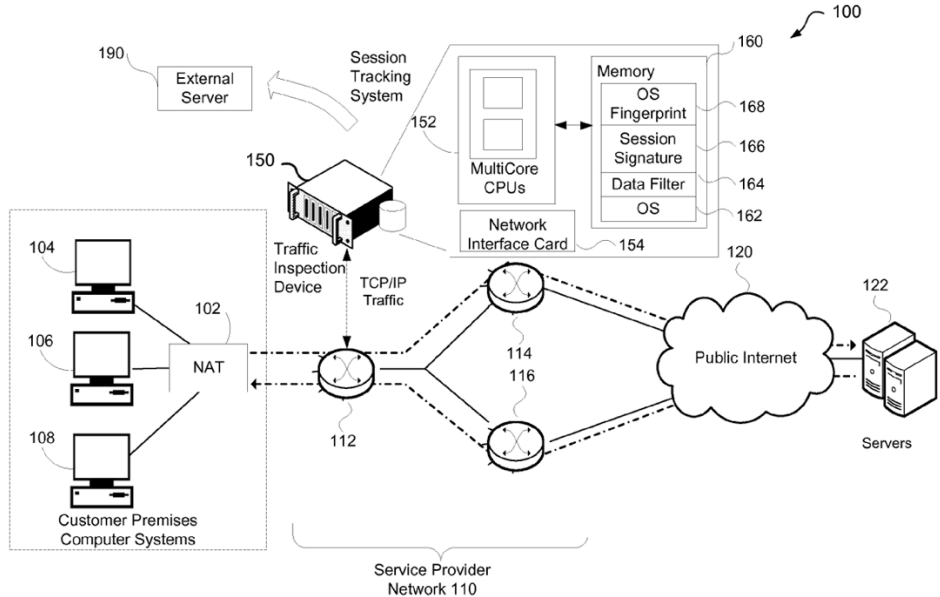
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1047 911 1142 940">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a NAT (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic' (112). The Traffic Inspection Device is connected to an 'External Server' (190) via a 'Session Tracking System' (152). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (114, 116), which is connected to the 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162).</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 2 and corresponding text:</p>  <pre> graph BT 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 240[Predictable IPID] --> 230 242[Predictable TCP timestamp] --> 230 244[Other Device] --> 230 </pre> <p>Predictable IPID (240):</p> <ul style="list-style-type: none"> •IPID Trend-line •HTTP User Agent •Browsing History •State Information •Session Activity and Start Time <p>Predictable TCP timestamp (242):</p> <ul style="list-style-type: none"> •TCP Timestamp <p>Other Device (244):</p> <ul style="list-style-type: none"> •Link layer protocol attributes •Transport layer protocol attributes •State information •Time and location data <p>Figure 2</p>

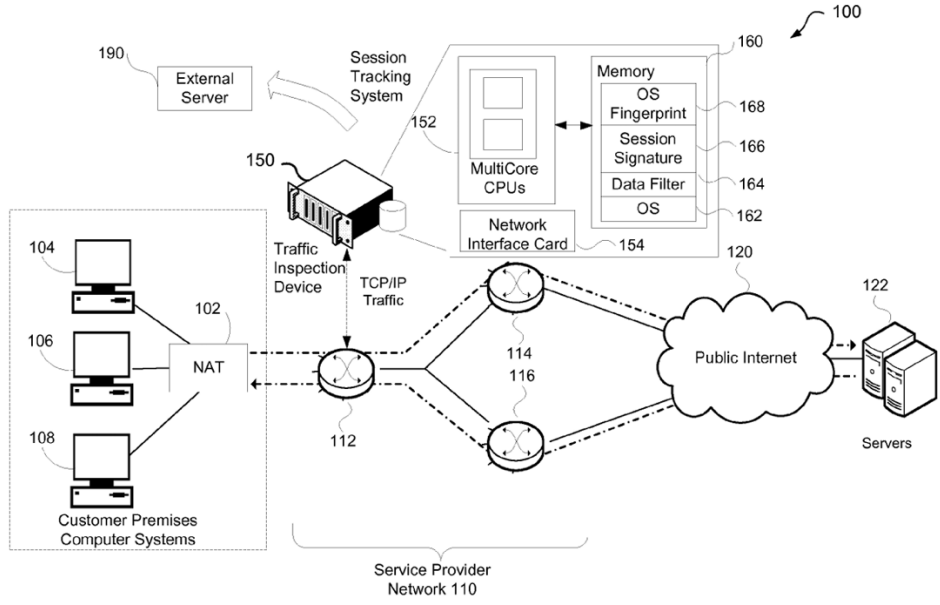
Claim	Exemplary Citation from Deridder																																										
	<p>Deridder at Fig. 3 and corresponding text:</p> <div><p>Figure 3</p><table><thead><tr><th>Session ID</th><th>Flow</th><th>OS Fingerprint</th><th>HTTP User Agent</th><th>IPID Trendline</th><th>TCP Timestamp</th><th>Session activity</th></tr></thead><tbody><tr><td>1</td><td>192.168.2.41; 8080; 200.1.2.1; 8080; TCP</td><td>13254</td><td>1</td><td>232</td><td>-</td><td>http://...</td></tr><tr><td>2</td><td>192.168.2.41; 8080; 145.77.2.1; 8080; TCP</td><td>42537</td><td>3</td><td>-</td><td>1234</td><td>-</td></tr><tr><td>3</td><td>192.168.2.117; 8080; 95.44.3.1; 8080; TCP</td><td>42538</td><td>3</td><td>-</td><td>1005</td><td>-</td></tr><tr><td>4</td><td>192.168.2.117; 8080; 112.15.7.1; 8080; TCP</td><td>13259</td><td>1</td><td>153</td><td>-</td><td>http://...</td></tr><tr><td>5</td><td>192.168.2.117; 8000; 243.45.17.34; 8000; TCP</td><td>13278</td><td>4</td><td>587</td><td>-</td><td>http://...</td></tr></tbody></table></div> <p>Deridder at ¶ 31: "An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>	Session ID	Flow	OS Fingerprint	HTTP User Agent	IPID Trendline	TCP Timestamp	Session activity	1	192.168.2.41; 8080; 200.1.2.1; 8080; TCP	13254	1	232	-	http://...	2	192.168.2.41; 8080; 145.77.2.1; 8080; TCP	42537	3	-	1234	-	3	192.168.2.117; 8080; 95.44.3.1; 8080; TCP	42538	3	-	1005	-	4	192.168.2.117; 8080; 112.15.7.1; 8080; TCP	13259	1	153	-	http://...	5	192.168.2.117; 8000; 243.45.17.34; 8000; TCP	13278	4	587	-	http://...
Session ID	Flow	OS Fingerprint	HTTP User Agent	IPID Trendline	TCP Timestamp	Session activity																																					
1	192.168.2.41; 8080; 200.1.2.1; 8080; TCP	13254	1	232	-	http://...																																					
2	192.168.2.41; 8080; 145.77.2.1; 8080; TCP	42537	3	-	1234	-																																					
3	192.168.2.117; 8080; 95.44.3.1; 8080; TCP	42538	3	-	1005	-																																					
4	192.168.2.117; 8080; 112.15.7.1; 8080; TCP	13259	1	153	-	http://...																																					
5	192.168.2.117; 8000; 243.45.17.34; 8000; TCP	13278	4	587	-	http://...																																					

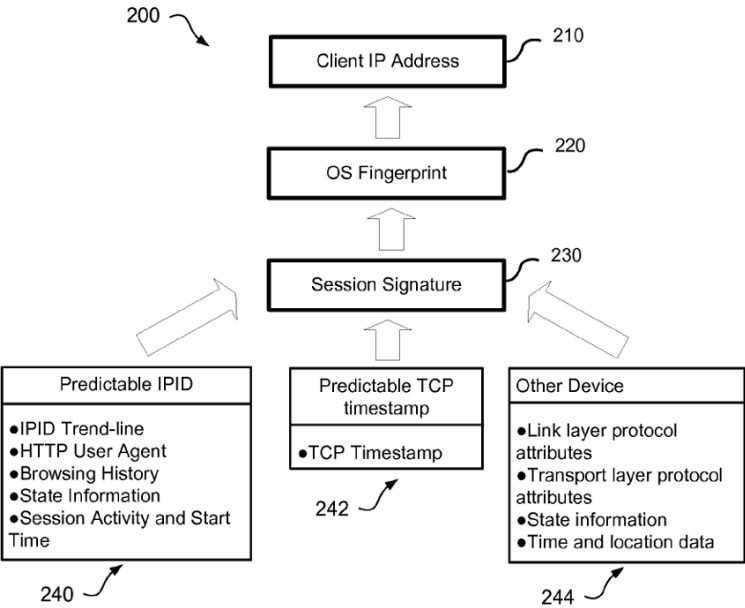
Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
<p>[1C] determining if malware is present in the TCP session and an associated malware ID by comparing a malware</p>	<p>Deridder discloses and/or renders obvious determining if malware is present in the TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets.</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core</p>

Claim	Exemplary Citation from Deridder
signature to the one or more TCP packets; and	computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."
[1D] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.	<p>Deridder discloses and/or renders obvious generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.</p> <p>Deridder at Abstract:</p> <p>"A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4:</p> <p>"The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference)</p>

Claim	Exemplary Citation from Deridder
	<p>have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p> <p>Deridder at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">Figure 1</p> <p>Deridder at ¶ 22:</p> <p>"The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with</p>

Claim	Exemplary Citation from Deridder
	external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."
[2PRE] The method of claim 1 wherein determining the operating system ID further comprises:	See claim 1.
[2A] determining a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first operating system identifier (ID);	<p>Deridder discloses and/or renders obvious determining a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first operating system identifier (ID).</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference)</p>

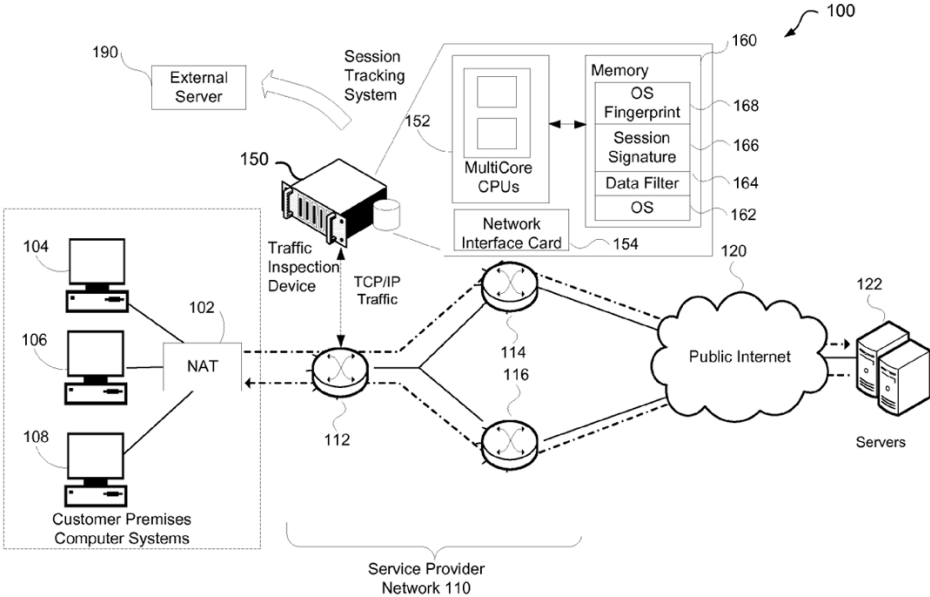
Claim	Exemplary Citation from Deridder
	<p>have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p> <p>Deridder at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">Figure 1</p>

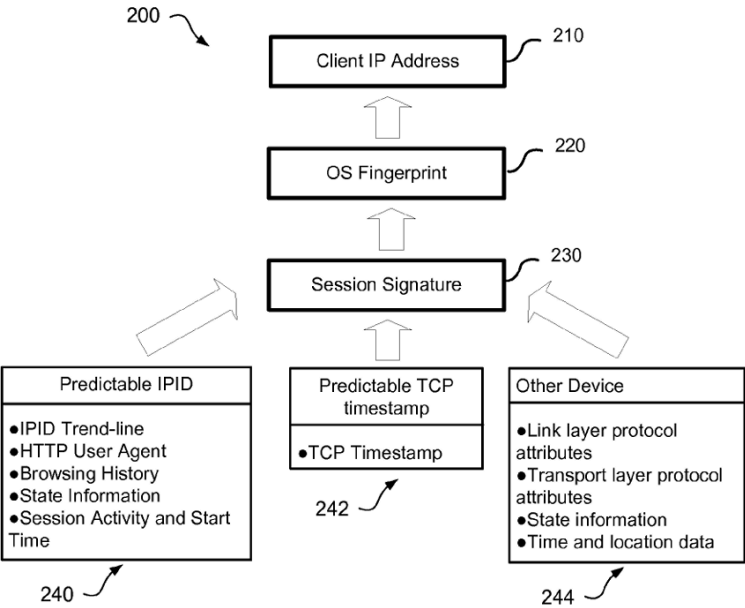
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 2 and corresponding text:</p>  <pre> graph BT 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 240[Predictable IPID] --> 230 242[Predictable TCP timestamp] --> 230 244[Other Device] --> 230 </pre> <p data-bbox="974 971 1087 1003">Figure 2</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 3 and corresponding text:</p> <p>Figure 3</p> <p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
<p>[2B] determining an application parameter associated with the TCP session if present, and identifying a second OS ID</p>	<p>Deridder discloses and/or renders obvious determining an application parameter associated with the TCP session if present, and identifying a second OS ID from the application parameter.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data</p>

Claim	Exemplary Citation from Deridder
<p>from the application parameter; and</p>	<p>packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4:</p> <p>"The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 909 1144 941">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a NAT (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic' (112). The Traffic Inspection Device is connected to an 'External Server' (190) via a 'Session Tracking System' (152). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (114, 116), which is connected to the 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162).</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 2 and corresponding text:</p>  <pre> graph TD 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 230 --> 210 240[Predictable IPID] --> 230 242[Predictable TCP timestamp] --> 230 244[Other Device] --> 230 </pre> <p data-bbox="974 971 1087 1003">Figure 2</p>

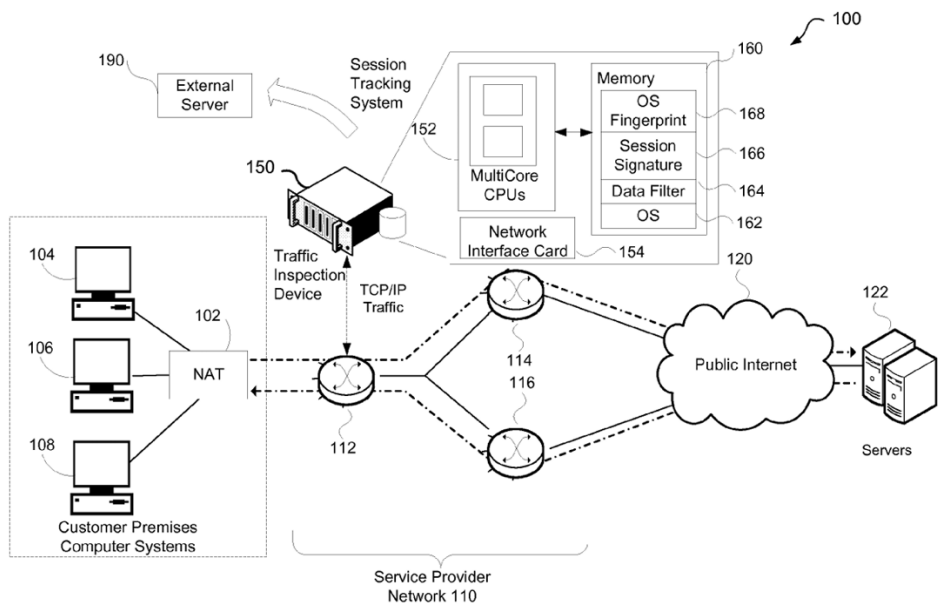
Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 3 and corresponding text:</p> <p>Figure 3</p> <p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
[2C] wherein the OS ID in the alert comprises one of the first OS ID or second OS ID associated with one or more computing devices	<p>Deridder discloses and/or renders obvious that the OS ID in the alert comprises one of the first OS ID or second OS ID associated with one or more computing devices coupled to the access device.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data</p>

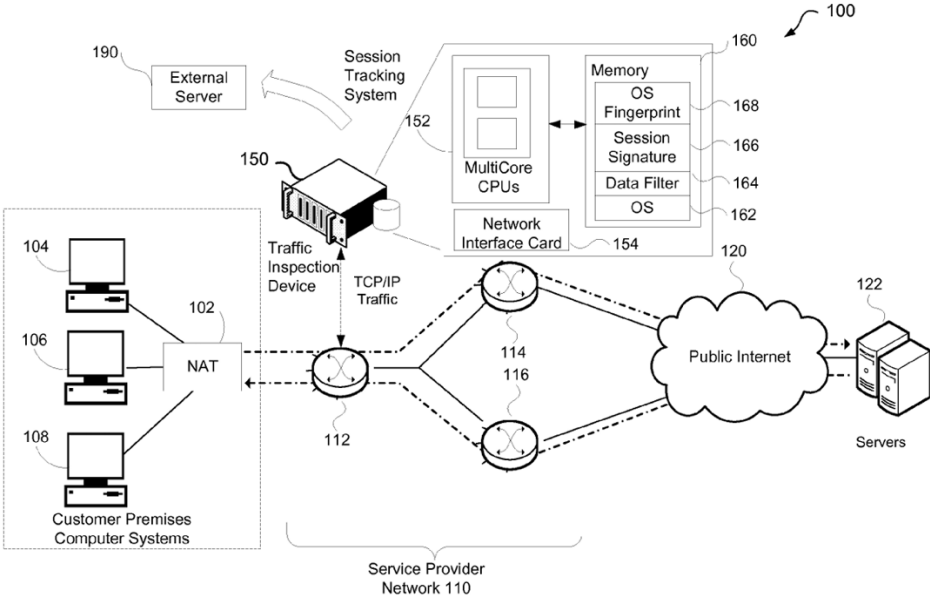
Claim	Exemplary Citation from Deridder
coupled to the access device.	<p>packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4:</p> <p>"The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 1 and corresponding text:</p> <div data-bbox="632 297 1560 894"> </div> <p data-bbox="1045 911 1142 943">Figure 1</p> <p data-bbox="598 992 814 1024">Deridder at ¶ 22:</p> <p data-bbox="598 1032 1894 1352">"The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p>

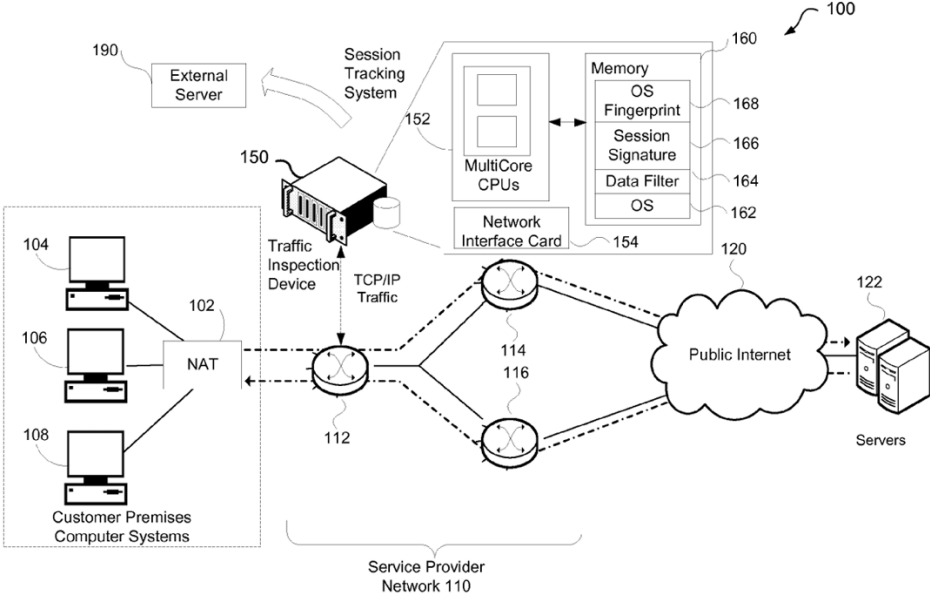
Claim	Exemplary Citation from Deridder
<p>[3] The method of claim 2 further comprising sending a notification to a subscriber associated with the network address associated with the access device, the notification identifying a remediation portal to remove the determined malware in the alert.</p>	<p>Deridder discloses and/or renders obvious sending a notification to a subscriber associated with the network address associated with the access device, the notification identifying a remediation portal to remove the determined malware in the alert.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

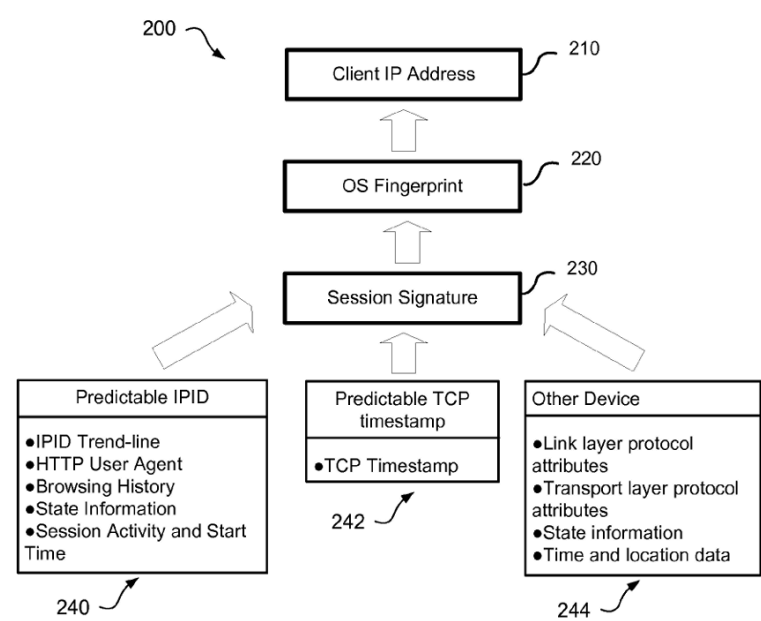
Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">Figure 1</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p>
[4PRE] The method of claim 3 further comprising:	See Claim 3.

Claim	Exemplary Citation from Deridder
<p>[4A] identifying at the remediation portal the subscriber accessing the remediation portal;</p>	<p>Deridder discloses and/or renders obvious identifying at the remediation portal the subscriber accessing the remediation portal.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">Figure 1</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p>
[4B] identifying the OS associated with the	Deridder discloses and/or renders obvious identifying the OS associated with the computing device used by the subscriber to access the remediation portal.

Claim	Exemplary Citation from Deridder
<p>computing device used by the subscriber to access the remediation portal;</p>	<p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

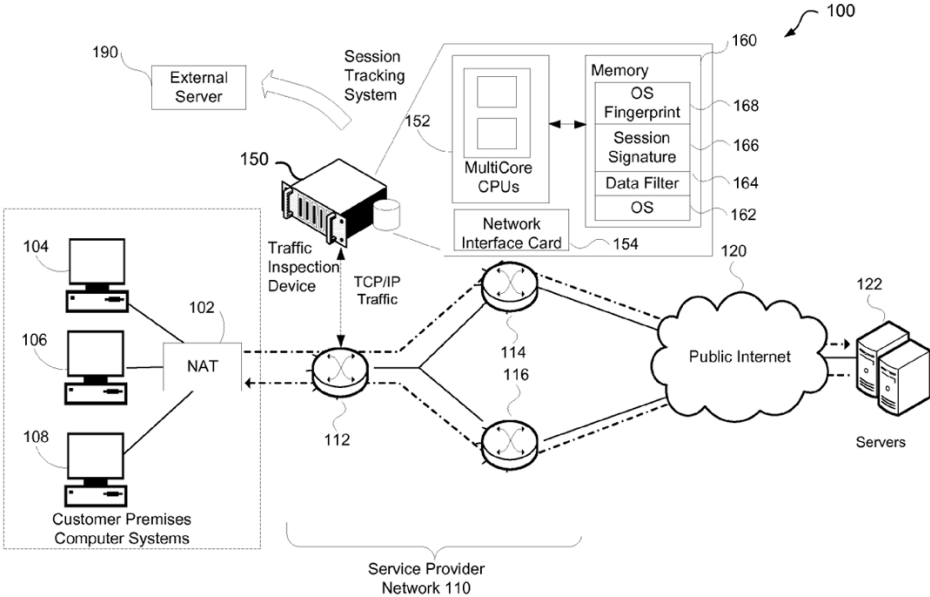
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 911 1142 940">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a NAT (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic'. The Traffic Inspection Device is connected to an 'External Server' (190). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (112, 114, 116), which is connected to a 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162). A 'Session Tracking System' (152) is also shown connected to the MultiCore CPUs.</p>

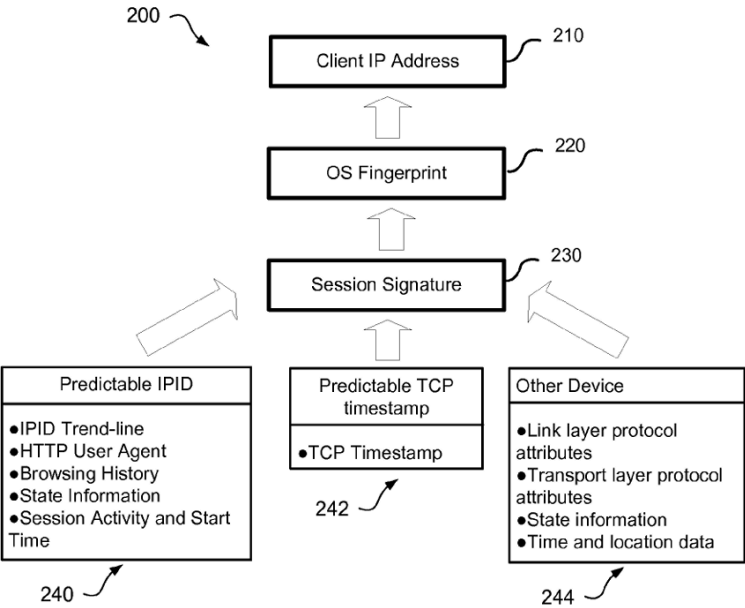
Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 2 and corresponding text:</p>  <pre> graph BT 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 240[Predictable IPID] --> 230 242[Predictable TCP timestamp] --> 230 244[Other Device] --> 230 </pre> <p>Predictable IPID (240):</p> <ul style="list-style-type: none"> •IPID Trend-line •HTTP User Agent •Browsing History •State Information •Session Activity and Start Time <p>Predictable TCP timestamp (242):</p> <ul style="list-style-type: none"> •TCP Timestamp <p>Other Device (244):</p> <ul style="list-style-type: none"> •Link layer protocol attributes •Transport layer protocol attributes •State information •Time and location data <p>Figure 2</p>

Claim	Exemplary Citation from Deridder																																										
	<p>Deridder at Fig. 3 and corresponding text:</p> <p>Figure 3</p> <table><thead><tr><th>Session ID</th><th>Flow</th><th>OS Fingerprint</th><th>HTTP User Agent</th><th>IPID Trendline</th><th>TCP Timestamp</th><th>Session activity</th></tr></thead><tbody><tr><td>1</td><td>192.168.2.41; 8080; 200.1.2.1; 8080; TCP</td><td>13254</td><td>1</td><td>232</td><td>-</td><td>http://...</td></tr><tr><td>2</td><td>192.168.2.41; 8080; 145.77.2.1; 8080; TCP</td><td>42537</td><td>3</td><td>-</td><td>1234</td><td>-</td></tr><tr><td>3</td><td>192.168.2.117; 8080; 95.44.3.1; 8080; TCP</td><td>42538</td><td>3</td><td>-</td><td>1005</td><td>-</td></tr><tr><td>4</td><td>192.168.2.117; 8080; 112.15.7.1; 8080; TCP</td><td>13259</td><td>1</td><td>153</td><td>-</td><td>http://...</td></tr><tr><td>5</td><td>192.168.2.117; 8000; 243.45.17.34; 8000; TCP</td><td>13278</td><td>4</td><td>587</td><td>-</td><td>http://...</td></tr></tbody></table>	Session ID	Flow	OS Fingerprint	HTTP User Agent	IPID Trendline	TCP Timestamp	Session activity	1	192.168.2.41; 8080; 200.1.2.1; 8080; TCP	13254	1	232	-	http://...	2	192.168.2.41; 8080; 145.77.2.1; 8080; TCP	42537	3	-	1234	-	3	192.168.2.117; 8080; 95.44.3.1; 8080; TCP	42538	3	-	1005	-	4	192.168.2.117; 8080; 112.15.7.1; 8080; TCP	13259	1	153	-	http://...	5	192.168.2.117; 8000; 243.45.17.34; 8000; TCP	13278	4	587	-	http://...
Session ID	Flow	OS Fingerprint	HTTP User Agent	IPID Trendline	TCP Timestamp	Session activity																																					
1	192.168.2.41; 8080; 200.1.2.1; 8080; TCP	13254	1	232	-	http://...																																					
2	192.168.2.41; 8080; 145.77.2.1; 8080; TCP	42537	3	-	1234	-																																					
3	192.168.2.117; 8080; 95.44.3.1; 8080; TCP	42538	3	-	1005	-																																					
4	192.168.2.117; 8080; 112.15.7.1; 8080; TCP	13259	1	153	-	http://...																																					
5	192.168.2.117; 8000; 243.45.17.34; 8000; TCP	13278	4	587	-	http://...																																					
	<p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>																																										

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
<p>[4C] comparing the OS associated with the computing device to the at least one of the first OS ID or second OS ID identified in the alert; and</p>	<p>Deridder discloses and/or renders obvious comparing the OS associated with the computing device to the at least one of the first OS ID or second OS ID identified in the alert.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data</p>

Claim	Exemplary Citation from Deridder
	<p>packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4:</p> <p>"The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

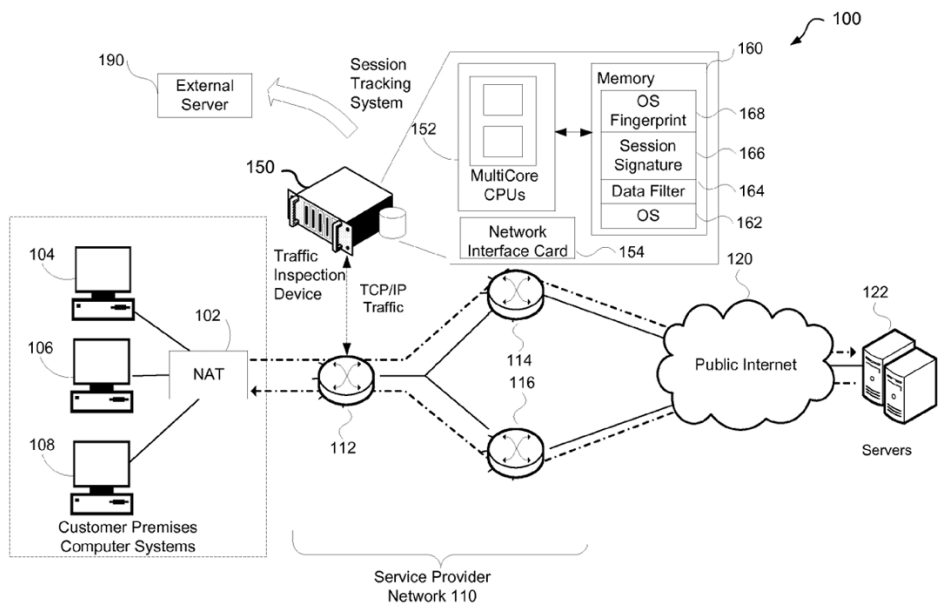
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 909 1142 938">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a NAT (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic'. The Traffic Inspection Device is connected to an 'External Server' (190). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (112, 114, 116), which is connected to the 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162). A 'Session Tracking System' (152) is also shown connected to the MultiCore CPUs.</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="596 240 1136 272">Deridder at Fig. 2 and corresponding text:</p>  <pre> graph TD 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 230 --> 220 240[Predictable IPID] --> 230 242[Predictable TCP timestamp] --> 230 244[Other Device] --> 230 240 --> 242 244 --> 242 </pre> <p data-bbox="974 971 1087 1003">Figure 2</p>

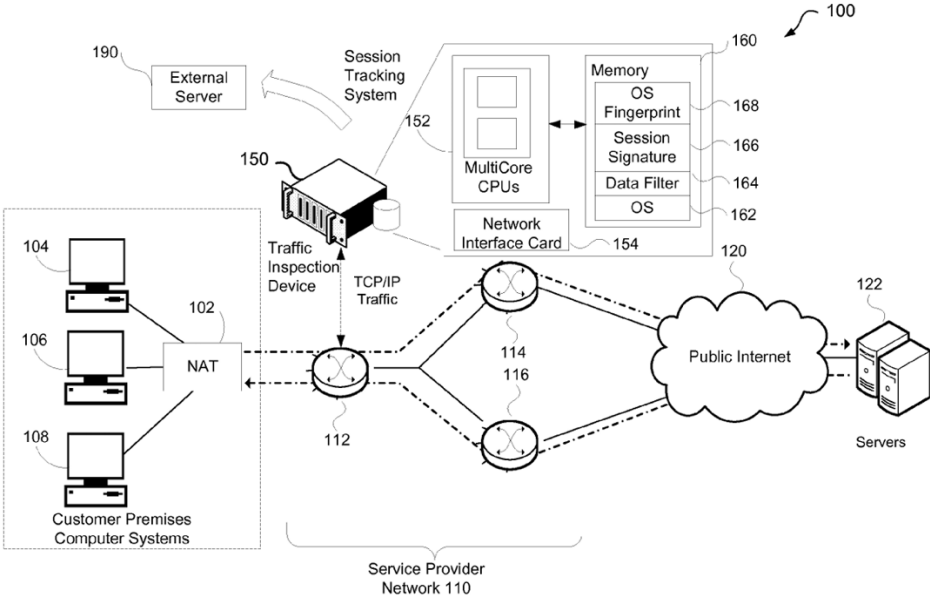
Claim	Exemplary Citation from Deridder																																										
	<p>Deridder at Fig. 3 and corresponding text:</p> <div><p>Figure 3</p><table><thead><tr><th>Session ID</th><th>Flow</th><th>OS Fingerprint</th><th>HTTP User Agent</th><th>IPID Trendline</th><th>TCP Timestamp</th><th>Session activity</th></tr></thead><tbody><tr><td>1</td><td>192.168.2.41; 8080; 200.1.2.1; 8080; TCP</td><td>13254</td><td>1</td><td>232</td><td>-</td><td>http://...</td></tr><tr><td>2</td><td>192.168.2.41; 8080; 145.77.2.1; 8080; TCP</td><td>42537</td><td>3</td><td>-</td><td>1234</td><td>-</td></tr><tr><td>3</td><td>192.168.2.117; 8080; 95.44.3.1; 8080; TCP</td><td>42538</td><td>3</td><td>-</td><td>1005</td><td>-</td></tr><tr><td>4</td><td>192.168.2.117; 8080; 112.15.7.1; 8080; TCP</td><td>13259</td><td>1</td><td>153</td><td>-</td><td>http://...</td></tr><tr><td>5</td><td>192.168.2.117; 8000; 243.45.17.34; 8000; TCP</td><td>13278</td><td>4</td><td>587</td><td>-</td><td>http://...</td></tr></tbody></table></div>	Session ID	Flow	OS Fingerprint	HTTP User Agent	IPID Trendline	TCP Timestamp	Session activity	1	192.168.2.41; 8080; 200.1.2.1; 8080; TCP	13254	1	232	-	http://...	2	192.168.2.41; 8080; 145.77.2.1; 8080; TCP	42537	3	-	1234	-	3	192.168.2.117; 8080; 95.44.3.1; 8080; TCP	42538	3	-	1005	-	4	192.168.2.117; 8080; 112.15.7.1; 8080; TCP	13259	1	153	-	http://...	5	192.168.2.117; 8000; 243.45.17.34; 8000; TCP	13278	4	587	-	http://...
Session ID	Flow	OS Fingerprint	HTTP User Agent	IPID Trendline	TCP Timestamp	Session activity																																					
1	192.168.2.41; 8080; 200.1.2.1; 8080; TCP	13254	1	232	-	http://...																																					
2	192.168.2.41; 8080; 145.77.2.1; 8080; TCP	42537	3	-	1234	-																																					
3	192.168.2.117; 8080; 95.44.3.1; 8080; TCP	42538	3	-	1005	-																																					
4	192.168.2.117; 8080; 112.15.7.1; 8080; TCP	13259	1	153	-	http://...																																					
5	192.168.2.117; 8000; 243.45.17.34; 8000; TCP	13278	4	587	-	http://...																																					
	<p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>																																										

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
[4D] providing malware remediation to the computing device when the at least one of the first OS ID or second OS ID	Deridder discloses and/or renders obvious providing malware remediation to the computing device when the at least one of the first OS ID or second OS ID matches the OS associated with the computing device.

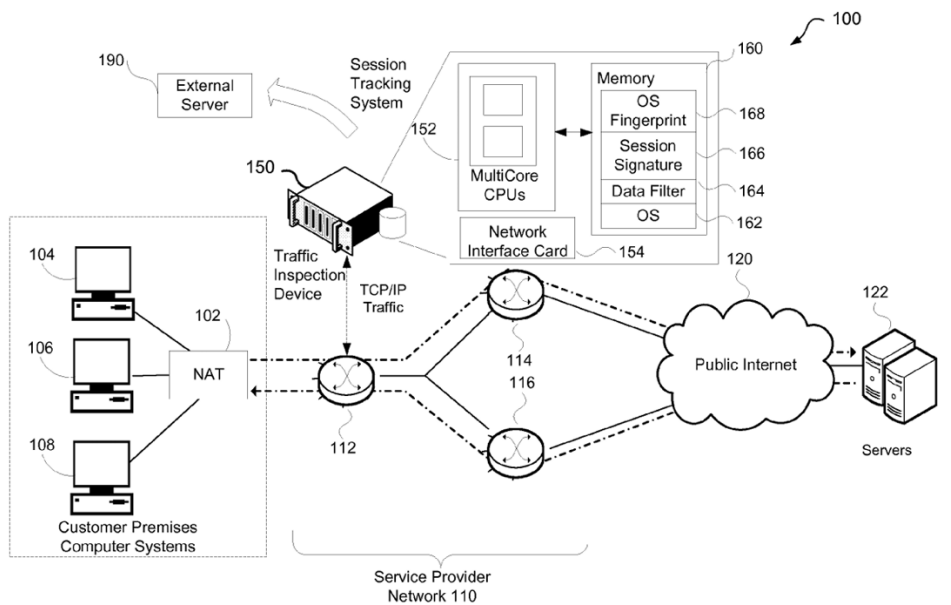
Claim	Exemplary Citation from Deridder
<p>matches the OS associated with the computing device.</p>	<p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">Figure 1</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p>
<p>[5] The method of claim 3 wherein if the OS of the</p>	<p>Deridder discloses and/or renders obvious that, if the OS of the computing device is not the at least one of the first OS ID or second OS ID identified in the alert an indication is provided to the</p>

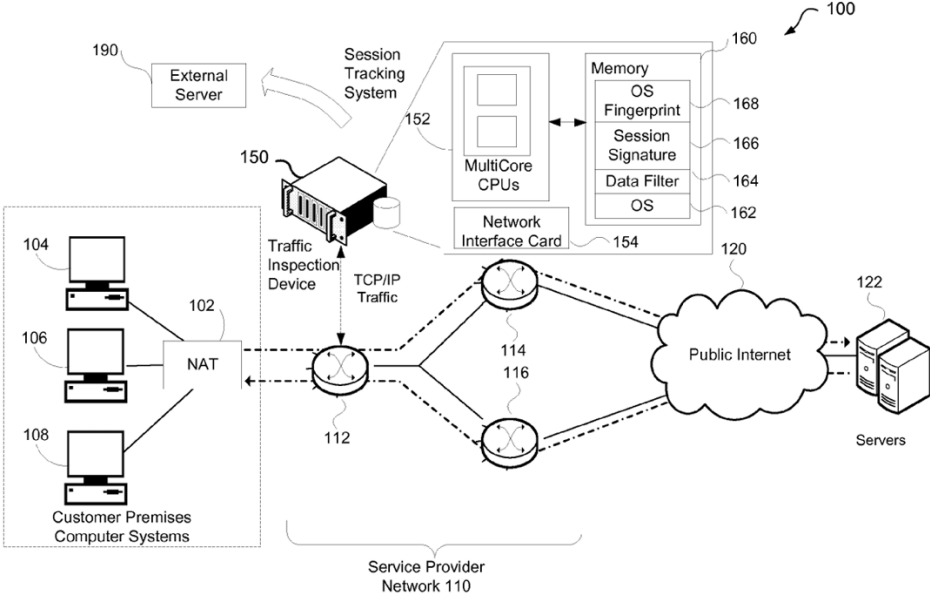
Claim	Exemplary Citation from Deridder
<p>computing device is not the at least one of the first OS ID or second OS ID identified in the alert an indication is provided to the subscriber to connect to the remediation portal with an other computing device connected to the access device.</p>	<p>subscriber to connect to the remediation portal with an other computing device connected to the access device.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 911 1142 943">Figure 1</p> <p data-bbox="598 995 814 1027">Deridder at ¶ 22:</p> <p data-bbox="598 1032 1894 1352">"The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p>

Claim	Exemplary Citation from Deridder
<p>[6] The method of claim 3 wherein the alert is cleared when all computing devices associated with the access device and the computing device having the at least one of the first OS ID or second OS ID have accessed the remediation portal.</p>	<p>Deridder discloses and/or renders obvious that the alert is cleared when all computing devices associated with the access device and the computing device having the at least one of the first OS ID or second OS ID have accessed the remediation portal.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">Figure 1</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p>
[7] The method of claim 2 wherein determining the	Deridder discloses and/or renders obvious that determining the first OS ID is performed by comparing one or more protocol parameters of the one or more TCP packets having a SYN flag

Claim	Exemplary Citation from Deridder
<p>first OS ID is performed by comparing one or more protocol parameters of the one or more TCP packets having a SYN flag associated with the session, wherein the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS ID associated with parameters.</p>	<p>associated with the session, wherein the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS ID associated with parameters.</p> <p>Deridder at Abstract:</p> <p>"A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4:</p> <p>"The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

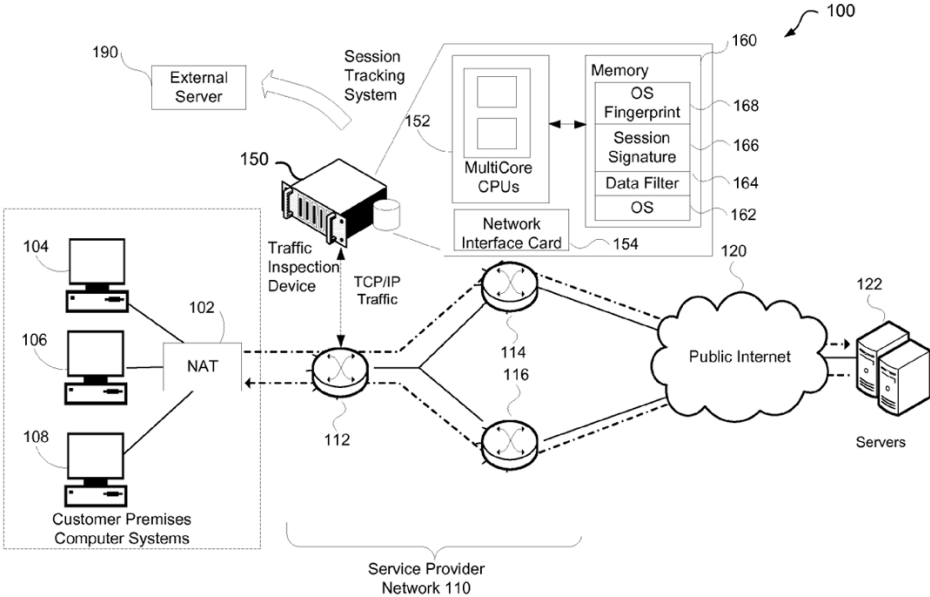
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1047 911 1142 940">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a NAT (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic'. The Traffic Inspection Device is connected to an 'External Server' (190). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (112, 114, 116), which is connected to the 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162). A 'Session Tracking System' (152) is also shown connected to the MultiCore CPUs.</p>

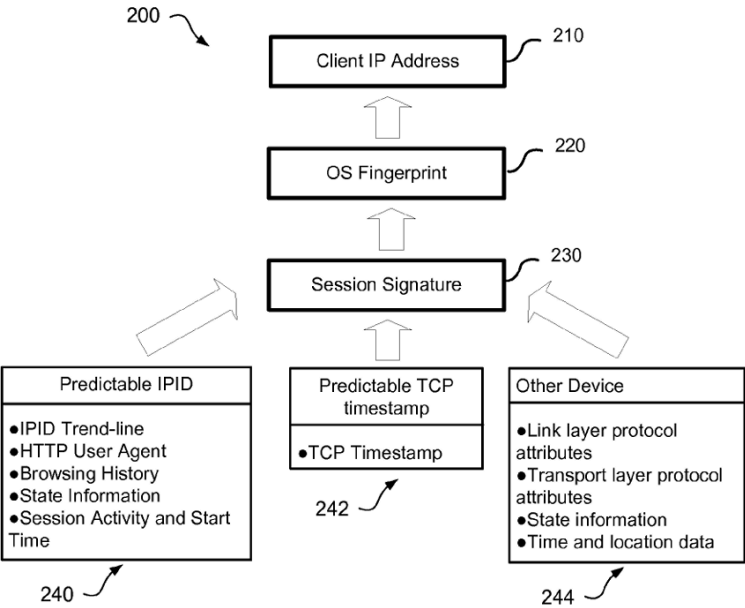
Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 2 and corresponding text:</p> <pre> graph TD 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 240[Predictable IPID] --> 230 242[Predictable TCP timestamp] --> 230 244[Other Device] --> 230 </pre> <p style="text-align: center;">Figure 2</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 3 and corresponding text:</p> <p>Figure 3</p> <p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
<p>[8] The method of claim 2 wherein determining the application parameter associated with the session and identifying the second OS ID from the application</p>	<p>Deridder discloses and/or renders obvious that determining the application parameter associated with the session and identifying the second OS ID from the application parameter by inspecting an HTTP user agent string to determine the second OS ID.</p>

Claim	Exemplary Citation from Deridder
<p>parameter by inspecting an HTTP user agent string to determine the second OS ID.</p>	<p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

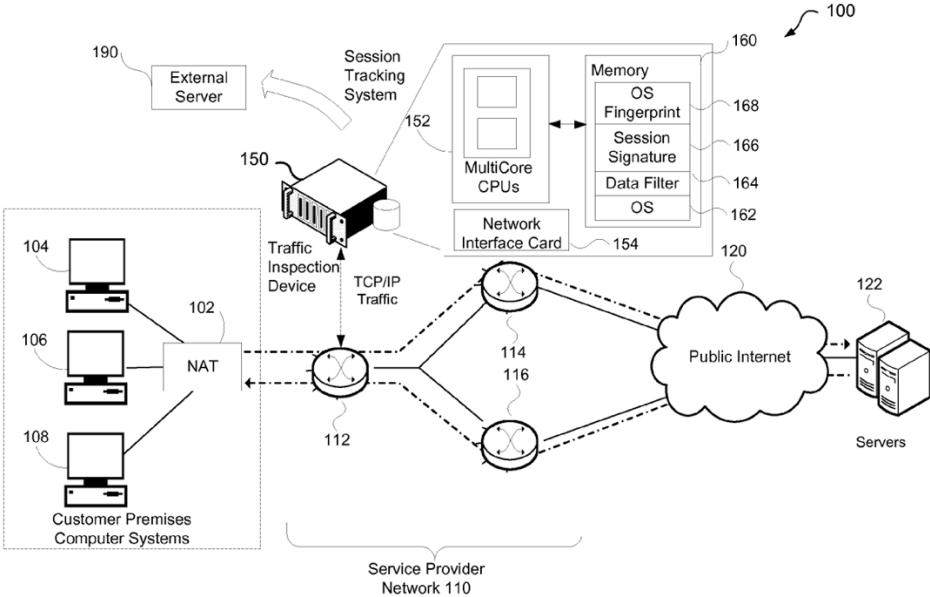
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1047 911 1142 940">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a NAT (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic'. The Traffic Inspection Device is connected to an 'External Server' (190) via a 'Session Tracking System' (152). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (112, 114, 116), which is connected to the 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162). The entire system is labeled 100.</p>

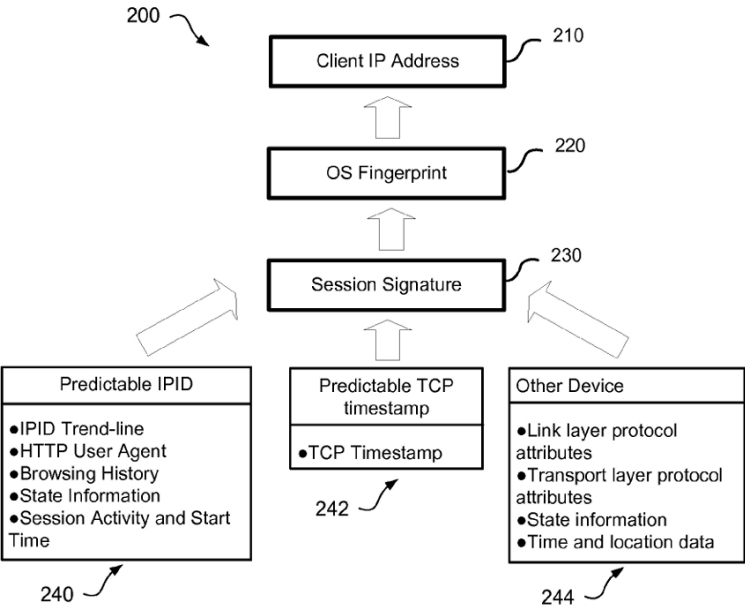
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 2 and corresponding text:</p>  <pre> graph BT 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 240[Predictable IPID] --> 230 242[Predictable TCP timestamp] --> 230 244[Other Device] --> 230 </pre> <p data-bbox="974 971 1087 1003">Figure 2</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 3 and corresponding text:</p> <p>Figure 3</p> <p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
<p>[9] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the</p>	<p>Deridder discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the session is identified by the OS ID identified as a more reliable OS ID.</p>

Claim	Exemplary Citation from Deridder
<p>session is identified by the OS ID identified as a more reliable OS ID.</p>	<p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

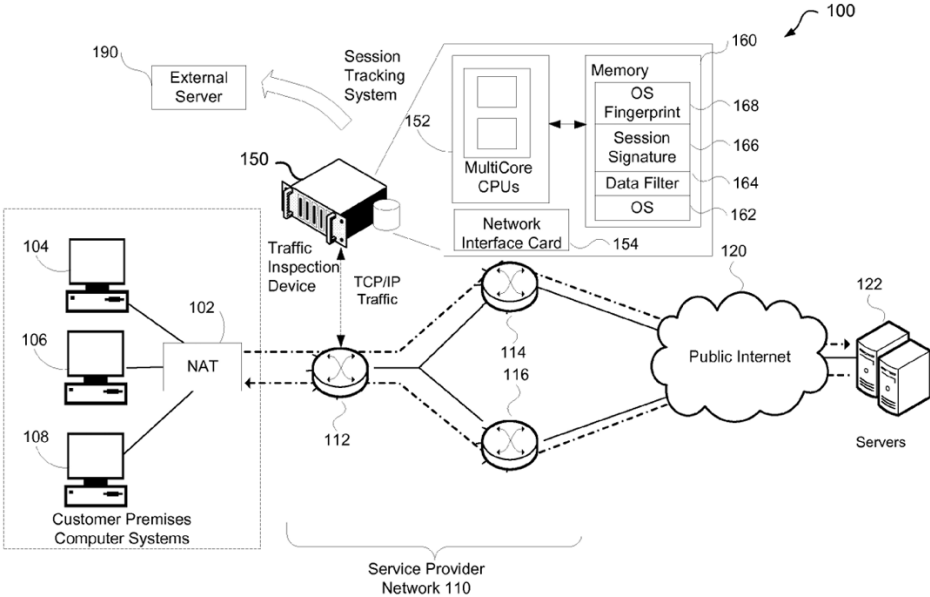
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1047 911 1142 940">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a NAT (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic'. The Traffic Inspection Device is connected to an 'External Server' (190). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (112, 114, 116), which is connected to the 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162). A 'Session Tracking System' (152) is also shown connected to the MultiCore CPUs.</p>

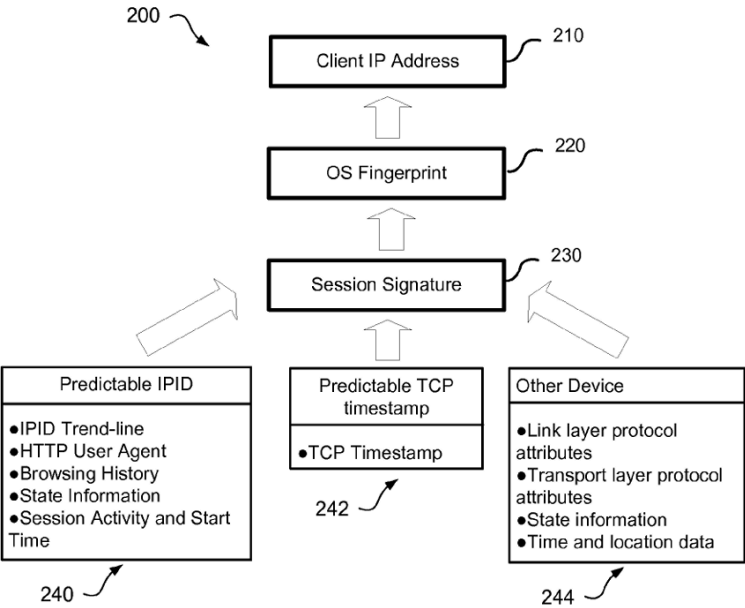
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 2 and corresponding text:</p>  <pre> graph TD 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 230 --> 240[Predictable IPID] 230 --> 242[Predictable TCP timestamp] 230 --> 244[Other Device] 240 --> 230 242 --> 230 244 --> 230 </pre> <p data-bbox="974 971 1087 1003">Figure 2</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 3 and corresponding text:</p> <p>Figure 3</p> <p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
<p>[10] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID matches</p>	<p>Deridder discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID either OS ID is used.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data</p>

Claim	Exemplary Citation from Deridder
<p>the first OS ID either OS ID is used.</p>	<p>packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4:</p> <p>"The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

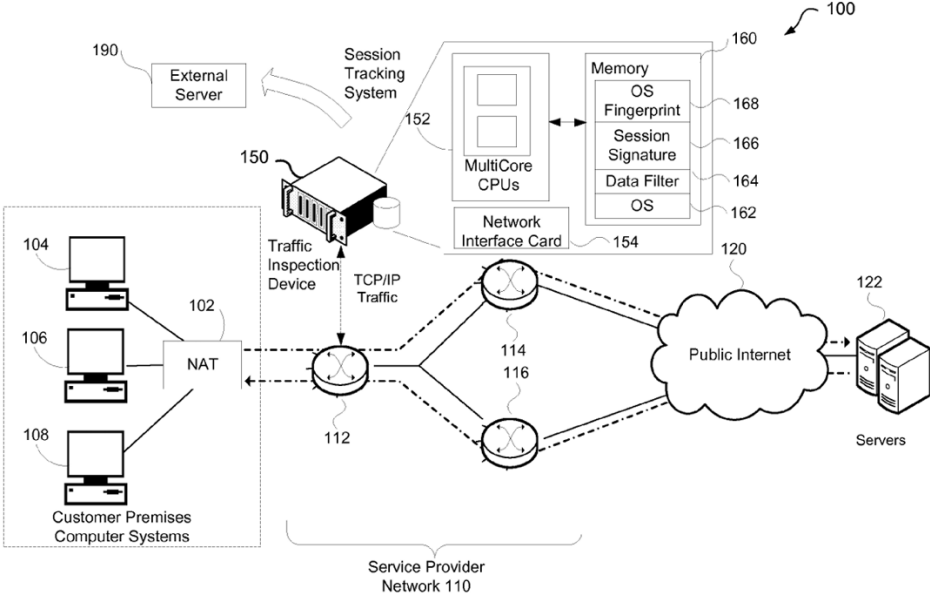
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 909 1142 941">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a NAT (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic'. The Traffic Inspection Device is connected to an 'External Server' (190). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (112, 114, 116), which is connected to the 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162). A 'Session Tracking System' (152) is also shown connected to the MultiCore CPUs.</p>

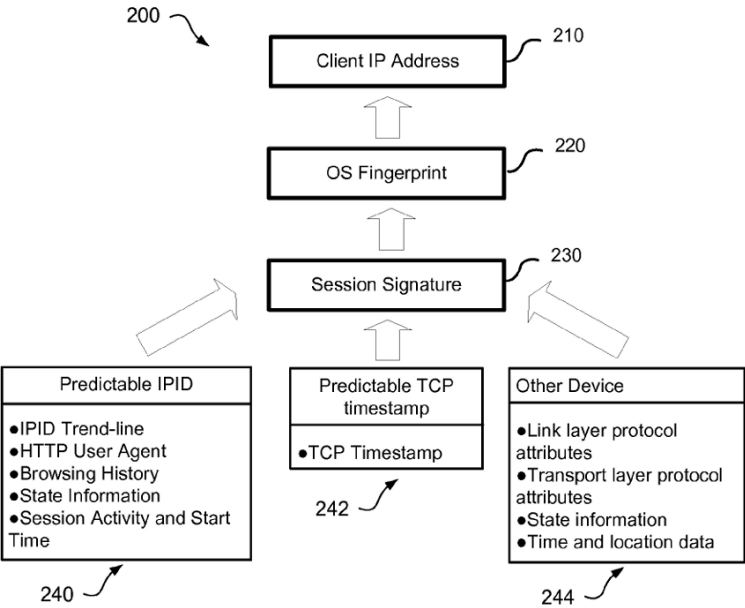
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 2 and corresponding text:</p>  <pre> graph TD 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 230 --> 210 230 --> 242[Predictable TCP timestamp] 242 --> 240[Predictable IPID] 242 --> 244[Other Device] 240 --> 230 244 --> 230 </pre> <p data-bbox="974 971 1087 1003">Figure 2</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 3 and corresponding text:</p> <p>Figure 3</p> <p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
<p>[11] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID but the</p>	<p>Deridder discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID but the second OS ID provides additional information, the second OS ID is used.</p>

Claim	Exemplary Citation from Deridder
<p>second OS ID provides additional information, the second OS ID is used.</p>	<p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1047 911 1142 940">Figure 1</p> <p>The diagram illustrates a network architecture for traffic inspection. On the left, 'Customer Premises Computer Systems' (104, 106, 108) are connected to a 'NAT' (102). The NAT is connected to a 'Traffic Inspection Device' (150) via 'TCP/IP Traffic'. The Traffic Inspection Device is connected to an 'External Server' (190). The Traffic Inspection Device is also connected to a 'Service Provider Network 110' (112, 114, 116), which is connected to the 'Public Internet' (120). The Public Internet is connected to 'Servers' (122). A detailed view of the Traffic Inspection Device (150) shows its internal components: 'MultiCore CPUs' (152), 'Network Interface Card' (154), and 'Memory' (160). The Memory contains an 'OS' (168), 'Fingerprint' (166), 'Session Signature' (164), 'Data Filter' (162), and another 'OS' (162). A 'Session Tracking System' (152) is also shown connected to the MultiCore CPUs.</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 2 and corresponding text:</p>  <pre> graph TD 210[Client IP Address] --> 220[OS Fingerprint] 220 --> 230[Session Signature] 230 --> 210 240[Predictable IPID] --> 230 242[Predictable TCP timestamp] --> 230 244[Other Device] --> 230 240 --> 242 244 --> 242 </pre> <p data-bbox="974 971 1087 1003">Figure 2</p>

Claim	Exemplary Citation from Deridder
	<p>Deridder at Fig. 3 and corresponding text:</p> <p>Figure 3</p> <p>Deridder at ¶ 31:</p> <p>"An OS fingerprint uses IP and TCP header fields from the initial TCP synchronization (SYN) packet of a connection to provide a unique signature or "fingerprint" of the OS that generated the packet. The present invention makes use of most of the fields found in the passive OS fingerprinting tool 'p0f'. However, unlike 'p0f', the present system does not try to match the fingerprint to a list of known</p>

Claim	Exemplary Citation from Deridder
	<p>OS fingerprints, because it is concerned less with the characterization of computer systems and more with the detection and tracking of unique sessions. Therefore, the presence of multiple distinct fingerprints is enough to indicate multiple hosts behind a NAT device."</p> <p>Deridder at ¶ 33: "An HTTP user agent string is an indicator of the application that generated an HTTP application layer message. This criteria is available for use in the algorithm to distinguish sessions, for example where one session is from the Internet Explorer® web browser and another session is from the Mozilla® Firefox® web browser."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p> <p>Deridder at ¶ 37: "For Mac® and Linux OS variants, the TCP timestamp option is used to estimate the uptime of the host. Although this value may be pseudo-random at boot time, it has been shown to increment in a highly predictable manner and is used to distinguish multiple Mac and/or Linux clients behind a NAT device. TCP timestamps can increment at different rates for different operating systems. This can be accommodated in the session matching algorithm by not requiring an exact match, but rather by allowing a timestamp to fall within a configurable range of an expected value."</p>
<p>[12] The method of claim 3 wherein sending the notification to the subscriber associated with the access device further comprises replacing the</p>	<p>Deridder discloses and/or renders obvious that sending the notification to the subscriber associated with the access device further comprises replacing the network address of the access device with a subscriber ID.</p>

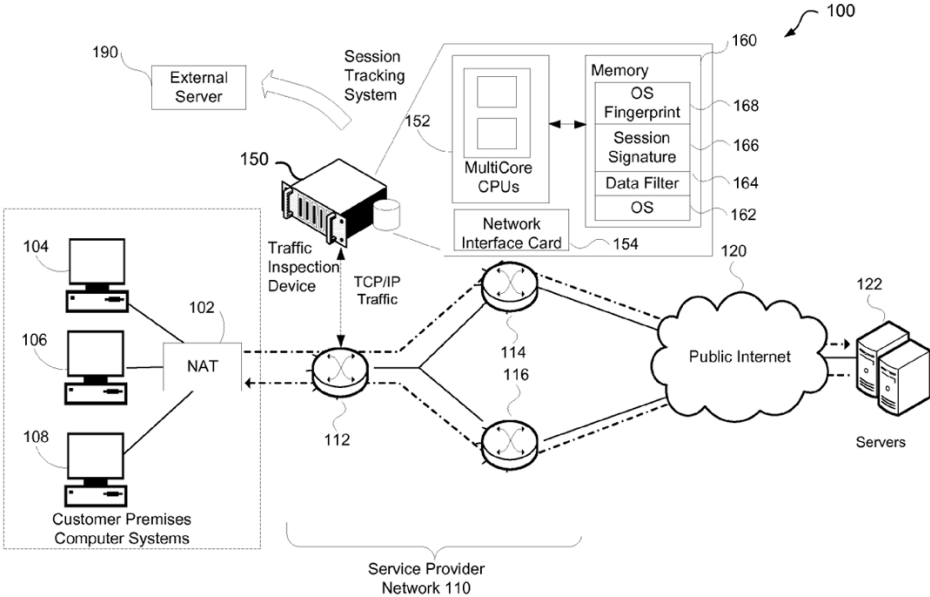
Claim	Exemplary Citation from Deridder
<p>network address of the access device with a subscriber ID.</p>	<p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 1 and corresponding text:</p> <div data-bbox="632 297 1560 894"> </div> <p data-bbox="1045 911 1142 943">Figure 1</p> <p data-bbox="598 992 816 1024">Deridder at ¶ 22:</p> <p data-bbox="598 1032 1894 1351">"The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p>

Claim	Exemplary Citation from Deridder
<p>[13] The method of claim 12 wherein sending the notification to the subscriber associated with the access device further comprises aggregating one or more detailed alerts having the subscriber ID into a single alert summary associated with the subscriber ID.</p>	<p>Deridder discloses and/or renders obvious that sending the notification to the subscriber associated with the access device further comprises aggregating one or more detailed alerts having the subscriber ID into a single alert summary associated with the subscriber ID.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

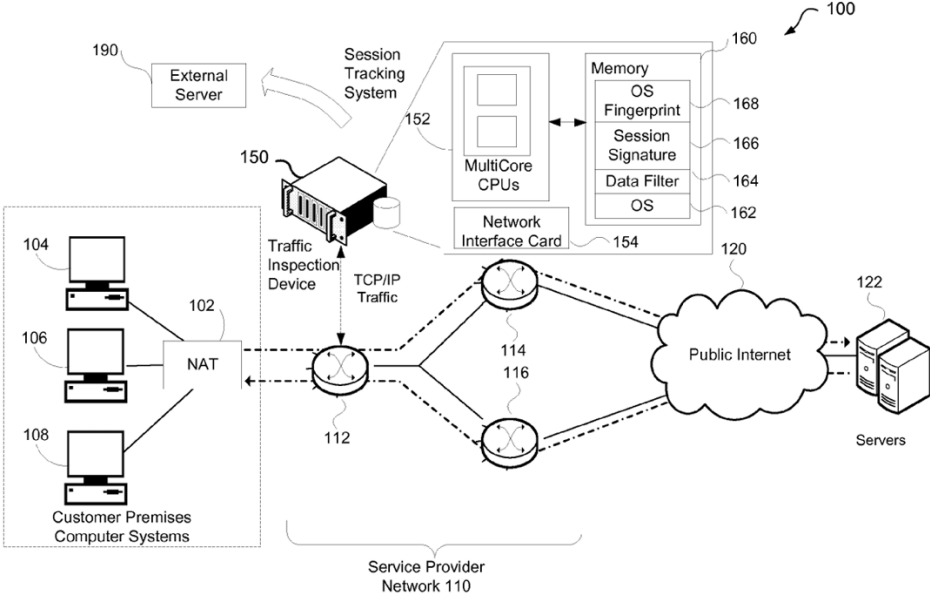
Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 1 and corresponding text:</p> <div data-bbox="632 297 1560 893"> </div> <p data-bbox="1045 911 1142 943">Figure 1</p> <p data-bbox="598 992 814 1024">Deridder at ¶ 22:</p> <p data-bbox="598 1032 1894 1357">"The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p>

Claim	Exemplary Citation from Deridder
<p>[14] The method of claim 12 wherein the one or more computing devices are coupled to the access device through a local area network (LAN), the access device providing network address translation (NAT) or is coupled to NAT device to share the network address of the access device.</p>	<p>Deridder discloses and/or renders obvious that the one or more computing devices are coupled to the access device through a local area network (LAN), the access device providing network address translation (NAT) or is coupled to NAT device to share the network address of the access device.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcamtuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 272">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 911 1142 943">Figure 1</p> <p data-bbox="598 992 816 1024">Deridder at ¶ 21:</p> <p data-bbox="598 1032 1894 1427">"FIG. 1 is a schematic representation of a session tracking system 100 for multi-user NAT session identification and tracking in a typical operating environment. The network traffic on the service provider network 110 is passed to a traffic inspection device 150 with a high-speed network interface that is capable of receiving all network packets for computational analysis. The service provider network comprises multiple routing devices, such as 112, 114 and 116 connected to additional networks such as the public internet 120 which external servers 122 are connected to. The session tracking system 100 can be implemented as part of the traffic inspection device (e.g. data filter) 150. The inspection system 150 can be implemented at the edge of the ISP network at router 112 to ensure all traffic between a user computer system (104 to 108) is inspected. Alternatively the inspection device may be located within the service provider network 110 with packets received from the NAT device 102 being copied or forwarded to the inspection device 150. Alternatively, the traffic</p>

Claim	Exemplary Citation from Deridder
	<p>inspection device 150 may function inline with the network traffic. The system can optionally be operated between a first mode in which NAT session tracking is enabled and a second mode in which NAT session tracking is disabled. When NAT session tracking is disabled, all traffic to or from a single client Internet Protocol (IP) address is viewed as belonging to a single user or session. When NAT session tracking is enabled, the sessions associated with multiple users (104 to 108) behind a NAT device 102 can be detected, and their traffic can be differentiated."</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p>
[15PRE] A system for network based malware detection in a service provider network, the system comprising:	See [1PRE].

Claim	Exemplary Citation from Deridder
<p>[15A] a plurality of network sensors coupled to the service provider network, each of the network sensors for:</p>	<p>Deridder discloses and/or renders obvious a plurality of network sensors coupled to the service provider network, each of the network sensors.</p> <p>Deridder at Abstract: "A method and apparatus are provided for identifying and tracking multiple computer user sessions associated with a network address translation (NAT) device coupled to a computer network. Data packets originating from the NAT device containing a request to a destination server coupled to the computer network are received. A client source address associated with the data packets is determined where the source address associated with the NAT device. One or more session entries associated with the client source address can then be determined, wherein the session entries are stored in system state information. A session signature can then be determined based one or more characteristics defined by the data packet header information and/or request header information in the one or more data packets, the session signature uniquely identifying one of the multiple computer user sessions coupled to the NAT device."</p> <p>Deridder at ¶ 3-4: "The popularity of network address translation (NAT) devices such as home wireless routers has given rise to an increasing number of households with multiple computing devices sharing a single network connection. These devices make it difficult to distinguish the network traffic of multiple clients within a local area network, as their traffic appears to emanate from a single source Internet Protocol (IP) address. Earlier proposed solutions have been based on operating system (OS) fingerprinting, Internet Protocol Identifier (IPID) analysis, and snooping network traffic headers and content. Tools such as p0f (refer to: 'passive OS fingerprinting tool', Michal Zalewski, <lcantuf@coredump.cx>, 2000-2006; the entirety of which is hereby incorporated by reference) have been used in a non-real-time manner to gain information about distinct hosts behind a NAT. These solutions do not address the problem of tracking multiple active hosts, behind a NAT device, simultaneously in real-time."</p>

Claim	Exemplary Citation from Deridder
	<p data-bbox="598 240 1136 269">Deridder at Fig. 1 and corresponding text:</p>  <p data-bbox="1045 911 1142 940">Figure 1</p> <p data-bbox="598 992 816 1021">Deridder at ¶ 21:</p> <p data-bbox="598 1029 1894 1427">"FIG. 1 is a schematic representation of a session tracking system 100 for multi-user NAT session identification and tracking in a typical operating environment. The network traffic on the service provider network 110 is passed to a traffic inspection device 150 with a high-speed network interface that is capable of receiving all network packets for computational analysis. The service provider network comprises multiple routing devices, such as 112, 114 and 116 connected to additional networks such as the public internet 120 which external servers 122 are connected to. The session tracking system 100 can be implemented as part of the traffic inspection device (e.g. data filter) 150. The inspection system 150 can be implemented at the edge of the ISP network at router 112 to ensure all traffic between a user computer system (104 to 108) is inspected. Alternatively the inspection device may be located within the service provider network 110 with packets received from the NAT device 102 being copied or forwarded to the inspection device 150. Alternatively, the traffic</p>

Claim	Exemplary Citation from Deridder
	<p>inspection device 150 may function inline with the network traffic. The system can optionally be operated between a first mode in which NAT session tracking is enabled and a second mode in which NAT session tracking is disabled. When NAT session tracking is disabled, all traffic to or from a single client Internet Protocol (IP) address is viewed as belonging to a single user or session. When NAT session tracking is enabled, the sessions associated with multiple users (104 to 108) behind a NAT device 102 can be detected, and their traffic can be differentiated."</p> <p>Deridder at ¶ 22: "The inspection device 150, implements the session tacking system 100 configured on a computing platform. The computing platform comprises at least a processing unit 152 such as multi-core computer processing units (CPUs) for executing instructions retrieved from a memory 160. The memory 160 provides instructions for implementing an operating system (OS) 162, a data filter function 164, an OS fingerprint function 166 and a session signature function 168. The memory 160 may also store and track identified sessions. A network interface card 154 provides an interface to the Service Provider Network 110 for receiving packets for processing and enables communication with external servers 190 which provide services for utilizing the session tracking functionality for services such as network monitoring, targeted advertising, malware detection etc."</p> <p>Deridder at ¶ 23: "Sessions (i.e. end-users on different computer systems (104 to 108)) are modeled as a combination of an operating system (OS) fingerprint and a session signature. The first stage of session detection involves the detection of an OS fingerprint from captured packets that provides a unique signature for a particular operating system. Multiple OS fingerprints emanating from a single source IP address serve to differentiate disparate computers (104 to 108) behind a NAT device 102. The OS fingerprint stage leverages a degree of packet reassembly to construct the TCP/IP flow and OS fingerprint information belonging to application-layer messages."</p>
<p>[15B] receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network,</p>	<p>See [1A].</p>

Claim	Exemplary Citation from Deridder
the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;	
[15C] determining an operating system identifier (OS ID) associated with the TCP session and the computing device associated;	<i>See</i> [1B].
[15D] determining if malware is present in the TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets; and	<i>See</i> [1C].
[15E] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.	<i>See</i> [1D].
[16PRE] The system of claim 15 wherein the plurality of network sensors in determining the OS ID further:	<i>See</i> [2PRE].

Claim	Exemplary Citation from Deridder
[16A] determine a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first OS ID;	<i>See</i> [2A].
[16B] determine an application parameter associated with the TCP session if present, and identifying a second OS ID from the application parameter; and	<i>See</i> [2B].
[16C] wherein the identified OS ID in the alert comprises one of the first OS ID or the second OS ID associated with one or more computing devices coupled to the access device.	<i>See</i> [2C].
[17] The system of claim 15 wherein the plurality of network sensors further: send a notification to a subscriber associated with the network address associated with the access device, the notification identifying a remediation portal to remove the	<i>See</i> claim 3

Claim	Exemplary Citation from Deridder
determined malware in the alert.	
[18] The system of claim 17 wherein the plurality of network sensors send the notification to the subscriber associated with the access device by retrieving a subscriber ID from a subscriber ID repository and replacing the network address of the access device with the subscriber ID.	<i>See claim 12.</i>
[19] The system of claim 16 wherein the plurality of network sensors determine the first OS ID by comparing one or more protocol parameters of the one or more TCP packets having a SYN flag associated with the session, the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS ID associated with parameters.	<i>See claim 7.</i>
[20] The system of claim 19 wherein the plurality of network sensors determine the application parameter	<i>See claim 8.</i>

Claim	Exemplary Citation from Deridder
associated with the session and identifies the second OS ID from the application parameter by inspecting an HTTP user agent string to determine the second OS ID.	
[21] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the session is identified by the OS ID providing a more reliable OS ID.	<i>See claim 9.</i>
[22] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID either OS ID is used.	<i>See claim 10.</i>
[23] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID but the	<i>See claim 11.</i>

Claim	Exemplary Citation from Deridder
second OS ID provides additional information, the second OS ID is used.	
[24PRE] A computer readable memory containing instructions for network based malware detection in a service provider network, the instructions when executed by a processor performing:	<i>See</i> [1PRE].
[24A] receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;	<i>See</i> [1A].
[24B] determining an operating system identifier (OS ID) associated with the TCP session and the computing device;	<i>See</i> [1B].
[24C] determining if malware is present in the TCP session and an	<i>See</i> [1C].

Claim	Exemplary Citation from Deridder
associated malware ID by comparing a malware signature to the one or more TCP packets; and	
[24D] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.	<i>See</i> [1D].
[25PRE] The system of claim 16 further comprising:	<i>See</i> [4PRE].
[25A] a remediation portal coupled to the service provider network, the remediation portal for:	<i>See</i> [4A].
[25B] identifying a subscriber accessing the remediation portal;	<i>See</i> [4A].
[25C] identifying the OS associated with the computing device used by the subscriber to access the remediation portal;	<i>See</i> [4B].
[25D] comparing the OS associated with the computing device to the OS ID identified in the alert; and	<i>See</i> [4C].

Claim	Exemplary Citation from Deridder
[25E] providing malware remediation to the computing device.	<i>See</i> [4D].

Ex. I-1 - Invalidity of U.S. Patent No. 8,635,697 against U.S. Patent Pub. No. 2008/0086773 to Tuvell et al. ("Tuvell")

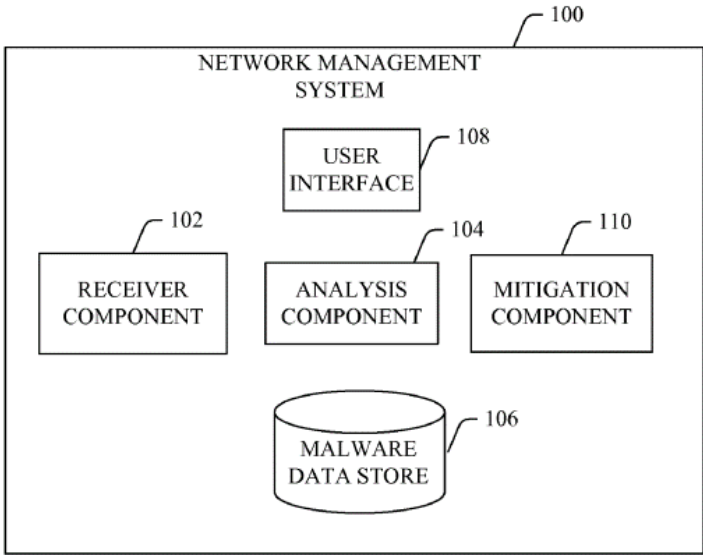
Fortinet, Inc. ("Fortinet") provides this chart subject to all reservations, objections, statements, and disclaimers set forth herein and in Fortinet's Preliminary Invalidity Contentions Cover Pleading, as well as any amendment, supplement, or modification thereof, which are incorporated herein by reference in their entirety.

On information and belief, and subject to further investigation and discovery, Tuvell was published April 10, 2008, and is thus available as prior art under at least §102(a), (b), and (e).

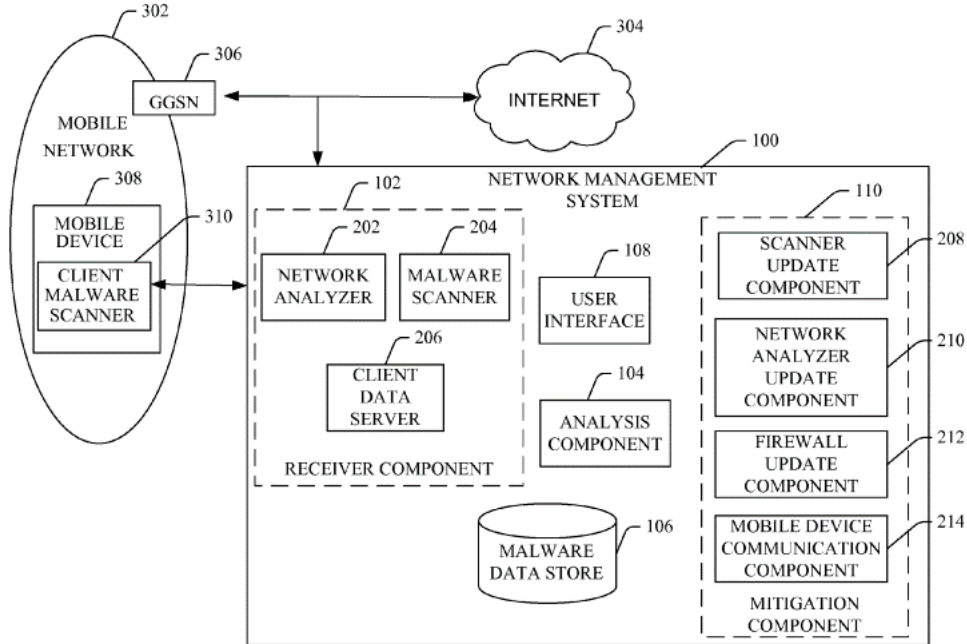
As illustrated in the chart below, Tuvell anticipates the asserted claims of the '697 Patent. To the extent the Tuvell is found not to expressly disclose certain limitations in the asserted claims, such limitations are inherent. To the extent the Tuvell is found not to anticipate any asserted claims or claim elements of the '697 Patent, the reference nevertheless renders those claims or claim elements obvious under 35 U.S.C. § 103, either alone or in combination with other art identified in the cover pleading or herein. These Preliminary Invalidity Contentions are not an admission by Fortinet that the accused products, including any current or past versions of the accused products, are covered by, or infringe the asserted claims, but are based instead on the recognition that if the claims are interpreted to be broad enough to encompass the accused products, the claims must also be construed to have that same scope when considering whether they are invalid.

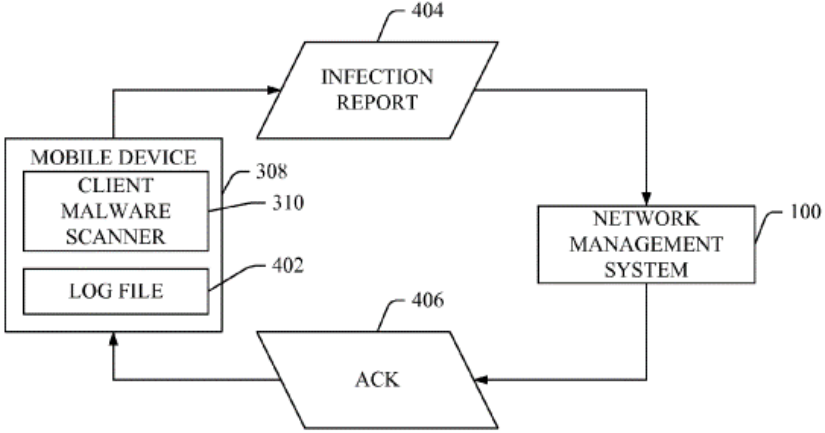
The following chart is partially based on, but is not limited by, the claim constructions implicit in Plaintiff's Infringement Contentions, to the extent that such constructions are apparent from the Infringement Contentions. Fortinet notes that in many instances, Plaintiff's Infringement Contentions fail to provide adequate notice of Plaintiff's construction of the asserted claims and fail to comply with the Court's scheduling order and other applicable rules. Fortinet does not accept the assumptions concerning the scope and meaning implicit in Plaintiff's Infringement Contentions, to the extent those assumptions are discernible, and reserves the right to challenge Plaintiff's proposed (or implied) constructions. Fortinet also reserves the right to revise and supplement these charts if and when Plaintiff is permitted to provide revised Infringement Contentions or otherwise make its positions known. To the extent that these Preliminary Invalidity Contentions rely on or otherwise embody particular constructions of terms or phrases in the asserted claims, Fortinet does not necessarily advocate any such construction as proper constructions of those terms or phrase. Fortinet also reserves the right to revise and supplement these charts after the Court construes the claims. Citations given in the chart below are merely representative of the respective elements and are not meant to be exhaustive.

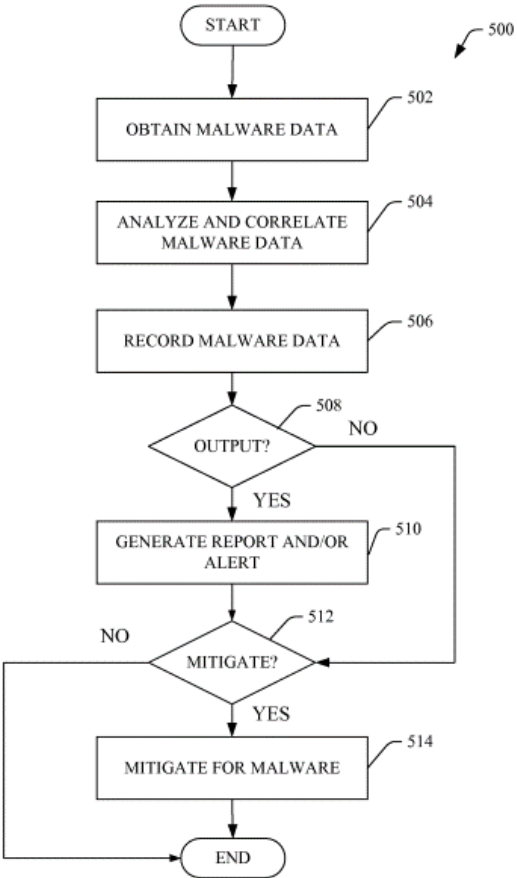
Claim	Exemplary Citation from Tuvell
<p>[1PRE] A method of network based malware detection in a service provider network, the method comprising:</p>	<p>Tuvell discloses and/or renders obvious a method of network based malware detection in a service provider network with the following steps.</p> <p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 2: "The present invention relates generally to systems, devices, and methods for detecting malware in mobile networks and mobile devices."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting,</p>

Claim	Exemplary Citation from Tuvell
	<p>reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p> <p>Tuvell at Fig. 1 and corresponding text:</p>  <p>The diagram, labeled FIG. 1, illustrates a Network Management System (100). It is a rectangular box containing several components. At the top center is the 'USER INTERFACE' (108). Below it, there are three rectangular boxes arranged horizontally: 'RECEIVER COMPONENT' (102) on the left, 'ANALYSIS COMPONENT' (104) in the center, and 'MITIGATION COMPONENT' (110) on the right. Below these three components is a cylindrical shape representing a 'MALWARE DATA STORE' (106). Each component is labeled with its name and a reference numeral (100, 102, 104, 106, 108, 110) with a line pointing to the respective component.</p> <p>FIG. 1</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 25:</p> <p>"FIG. 1 depicts an exemplary network management system 100, also referred to herein as CoreStats, that provides for reporting and visualizing viruses on mobile networks. As used herein, the term "exemplary" indicates a sample or example. It is not indicative of preference over other aspects or embodiments. The network management system 100 includes a receiver component 102 that obtains or receives malware data for a mobile network (not shown). As used herein, the term "component" refers to hardware, software, firmware or any combination thereof."</p> <p>Tuvell at Fig. 2 and corresponding text:</p> <pre> graph TD subgraph 100 [NETWORK MANAGEMENT SYSTEM] subgraph 102 [RECEIVER COMPONENT] 202[NETWORK ANALYZER] 204[MALWARE SCANNER] 206[CLIENT DATA SERVER] end 108[USER INTERFACE] 104[ANALYSIS COMPONENT] 106[(MALWARE DATA STORE)] subgraph 110 [MITIGATION COMPONENT] 208[SCANNER UPDATE COMPONENT] 210[NETWORK ANALYZER UPDATE COMPONENT] 212[FIREWALL UPDATE COMPONENT] 214[MOBILE DEVICE COMMUNICATION COMPONENT] end end </pre> <p>FIG. 2</p>

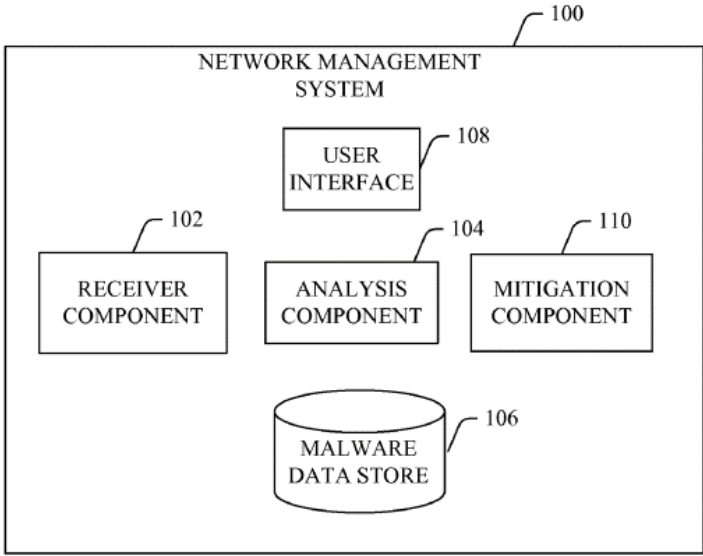
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

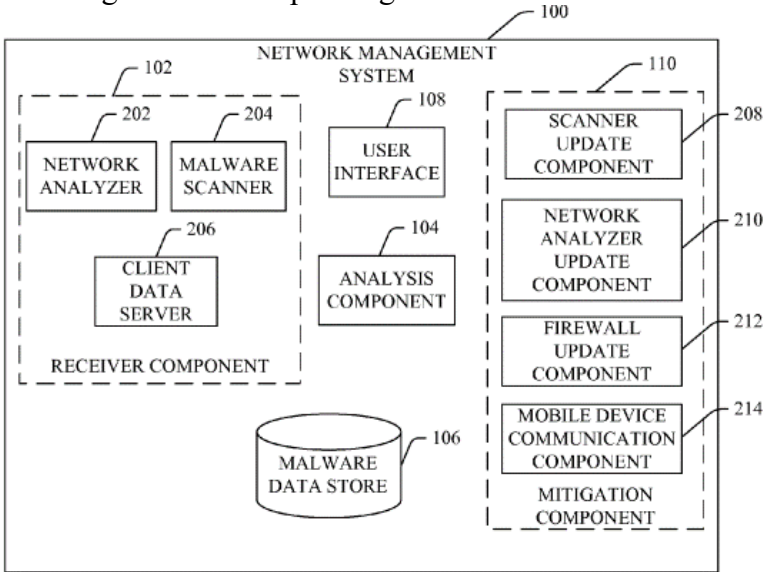
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre>

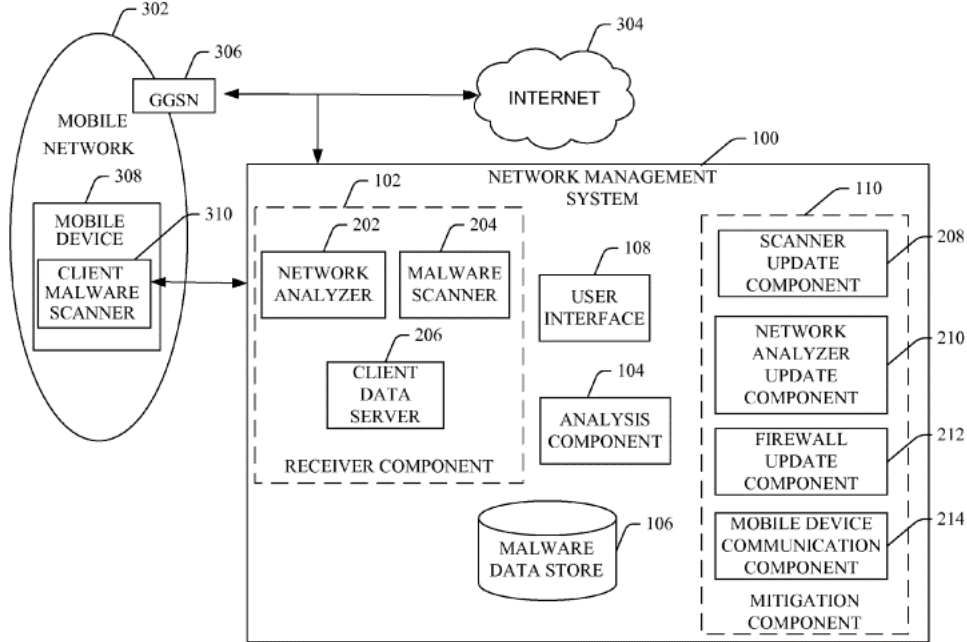
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 510 --> 512{MITIGATE?} 508 -- NO --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 514 --> END([END]) 512 -- NO --> END </pre> <p>FIG. 5</p> <p>Tuvell at ¶ 43-44: "Referring now to FIG. 3, an exemplary deployment of a sample CoreStats system 100 in a network environment is depicted. FIG. 3 illustrates a deployment of CoreStats 100 between the edge of a mobile network 302 and the Internet 304, although it can also be deployed effectively at various other points in the mobile network 302 depending upon the network topology and desired coverage.</p>

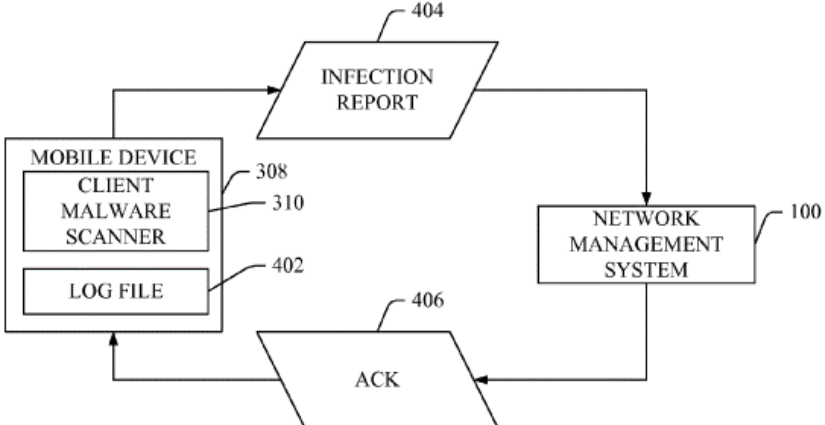
Claim	Exemplary Citation from Tuvell
	<p>The network analyzer 202 monitors and evaluates all traffic going from the mobile network 302 to the public data networks (e.g., the Internet 304) and vice-versa. The network analyzer 202 can intercept packets on either side of a firewall (not shown). In an embodiment, CoreStats 100 monitors a mobile network (or operator's network) 302 by monitoring or packet sniffing IP packets passing from the Gateway General Packet Radio Service ("GPRS") Support Node or Gateway GPRS Support Node ("GGSN") 306 and the Internet 304. In an embodiment, CoreStats 100 is deployed between the edge of the mobile network 302 and the Internet 304. The GGSN 306 links the access dependent Radio Access Network (RAN), shown on the figure as the mobile network 302, to the access independent Internet 304. RAN comprises the entire radio/wireless network with a variety of protocols for data transfer (e.g., CDMA, GPRS, 802.11). The GGSN 306 acts as a gateway between the mobile network 302 and the Internet 304, converting access-specific packet data to IP packets and vice-versa. As discussed above, the intercepted packets are processed by the malware scanner 204 and the resulting malware data is provided to the analysis component 104."</p>
<p>[1A] receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;</p>	<p>Tuvell discloses and/or renders obvious receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network.</p> <p>Tuvell at Abstract:</p> <p>"A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5:</p> <p>"Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new</p>

Claim	Exemplary Citation from Tuvell
	<p>malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p>

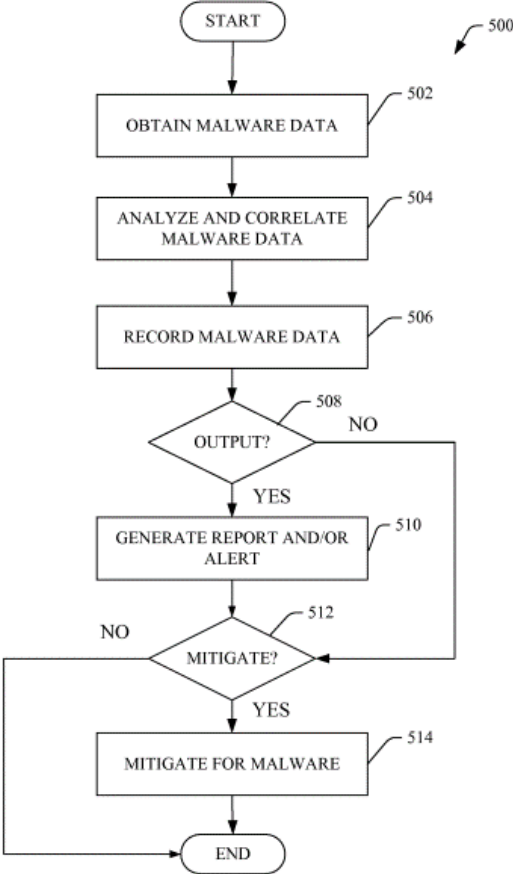
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 26: "In another embodiment, the receiver component 102 obtains malware data from a plurality of sources, such as individual mobile devices, mobile network traffic and/or computer network traffic (e.g., Internet Protocol (IP) packets). Malware data obtained from multiple sources provides a more</p>

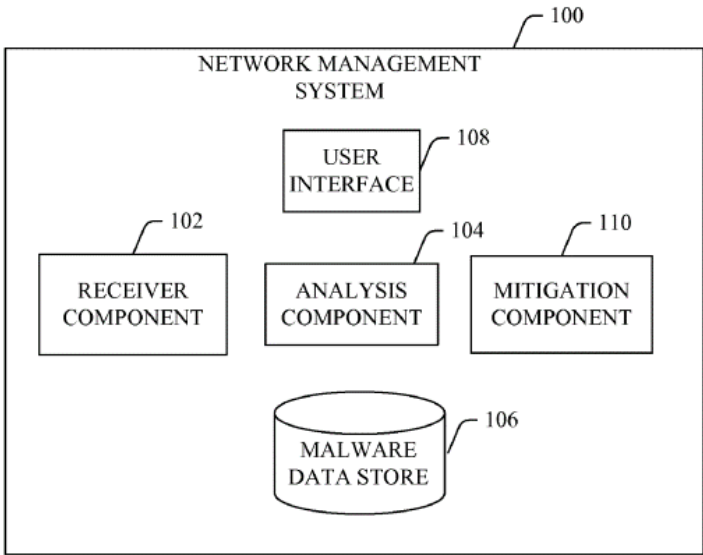
Claim	Exemplary Citation from Tuvell
	<p>complete picture of the current state of a mobile network. Consequently, collection or receipt of information from a variety of sources facilitates the detection and analysis of the spread of malware."</p> <p>Tuvell at Fig. 2 and corresponding text:</p>  <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 33-34: "Turning now to FIG. 2, a network management system or CoreStats 100 is illustrated in greater detail. In an embodiment, the receiver component 102 includes a network analyzer 202 or packet sniffer that monitors network traffic. The network analyzer 202 can be implemented as software</p>

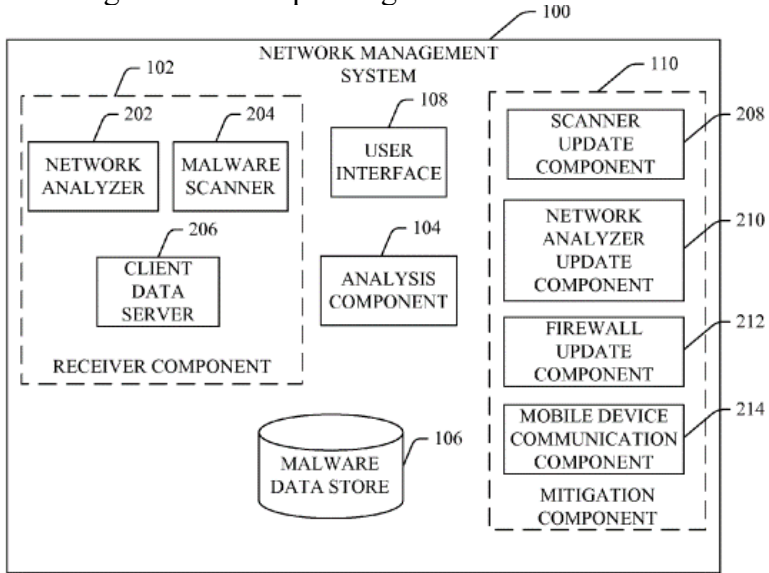
Claim	Exemplary Citation from Tuvell
	<p>and/or hardware that intercepts and logs traffic passing over a network or a portion of a network. In an embodiment, the network analyzer 202 intercepts communications between the mobile network and a data network (e.g., the Internet). In an alternate embodiment, the network analyzer 202 intercepts data within the mobile network. Intercepted or "sniffed" data packets are analyzed by a data stream scanner or malware scanner 204 to identify malware present in the data packets and the addresses of the transmitting and/or receiving mobile device. Data is analyzed in real time packet by packet or stored and analyzed non-linearly. In some instances, the data packets may need to be reassembled in the proper order and the contents extracted before analysis can be done."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

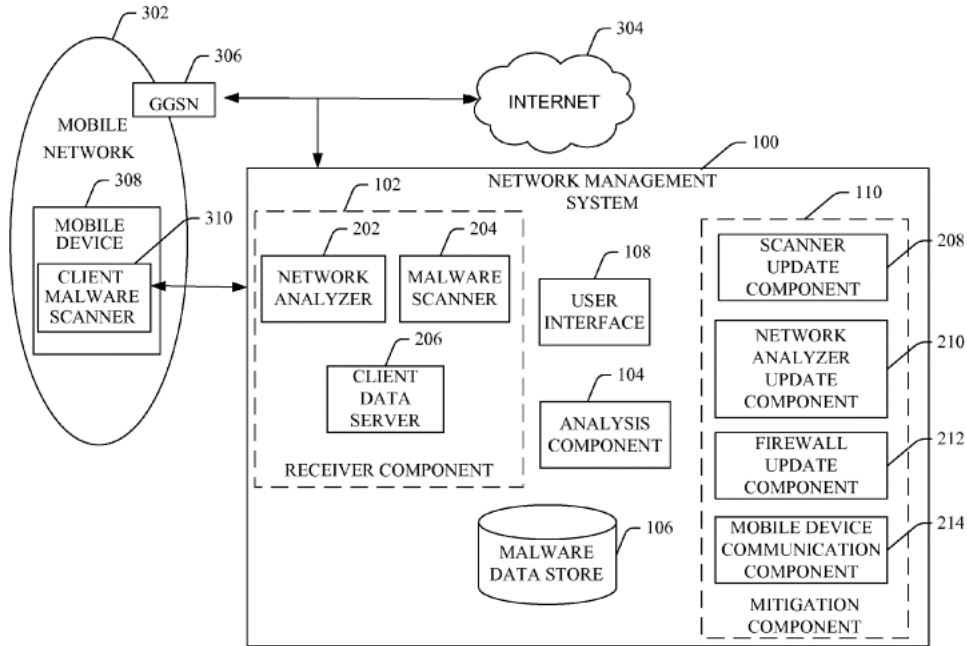
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD MD[MOBILE DEVICE 308] --> IR[/INFECTION REPORT 404/] IR --> NMS[NETWORK MANAGEMENT SYSTEM 100] NMS --> ACK[/ACK 406/] ACK --> MD subgraph MD_Box [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end </pre> <p>Tuvell at ¶ 43-44:</p> <p>"Referring now to FIG. 3, an exemplary deployment of a sample CoreStats system 100 in a network environment is depicted. FIG. 3 illustrates a deployment of CoreStats 100 between the edge of a mobile network 302 and the Internet 304, although it can also be deployed effectively at various other points in the mobile network 302 depending upon the network topology and desired coverage. The network analyzer 202 monitors and evaluates all traffic going from the mobile network 302 to the public data networks (e.g., the Internet 304) and vice-versa. The network analyzer 202 can intercept packets on either side of a firewall (not shown). In an embodiment, CoreStats 100 monitors a mobile network (or operator's network) 302 by monitoring or packet sniffing IP packets passing from the Gateway General Packet Radio Service ("GPRS") Support Node or Gateway GPRS Support Node ("GGSN") 306 and the Internet 304. In an embodiment, CoreStats 100 is deployed between the edge of the mobile network 302 and the Internet 304. The GGSN 306 links the access dependent Radio Access Network (RAN), shown on the figure as the mobile network 302, to the access independent Internet 304. RAN comprises the entire radio/wireless network with a variety of protocols for data transfer (e.g., CDMA, GPRS, 802.11). The GGSN 306 acts as a gateway between</p>

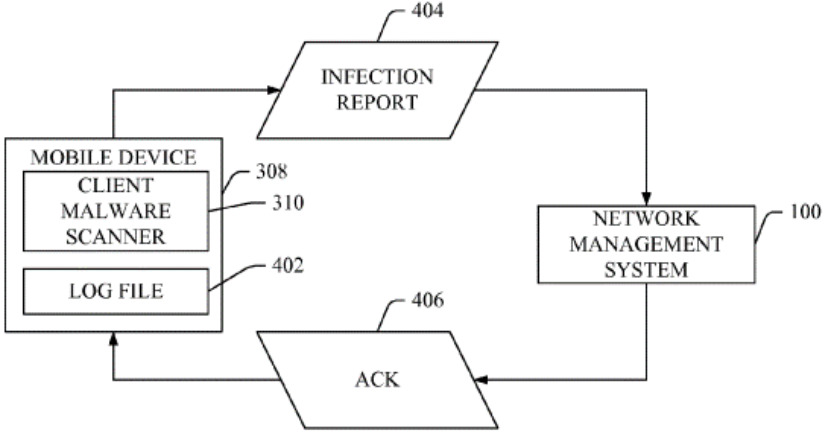
Claim	Exemplary Citation from Tuvell
	<p>the mobile network 302 and the Internet 304, converting access-specific packet data to IP packets and vice-versa. As discussed above, the intercepted packets are processed by the malware scanner 204 and the resulting malware data is provided to the analysis component 104."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 vulnerability to malware. Early detection of the vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP, Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, such as malware that spreads using the mobile network 302 via TCP/IP, Messaging, and/or other protocols. Furthermore, CoreStats 100 can generate spreading statistics of short range malware, such as malware that spreads over Bluetooth, memory cards, or other means without being transported across the mobile network 302."</p> <p>Tuvell at ¶ 56: "In another embodiment, CoreStats 100 provides information relating to the infected network traffic itself, such as malware identification, traffic patterns and topologies, and the like. In yet another embodiment, CoreStats 100 computes vulnerability of particular mobile devices 308 based on acquired heuristic data about infected mobile devices 308, protocols used, type of malware and the like. In still another embodiment, CoreStats 100 determines vulnerability of a mobile network 302 to certain kinds of malware."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> 512 514 --> END([END]) </pre> <p>FIG. 5</p>
<p>[1B] determining an operating system identifier (OS ID) associated with the TCP session and the computing device;</p>	<p>Tuvell discloses and/or renders obvious determining an operating system identifier (OS ID) associated with the TCP session and the computing device.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

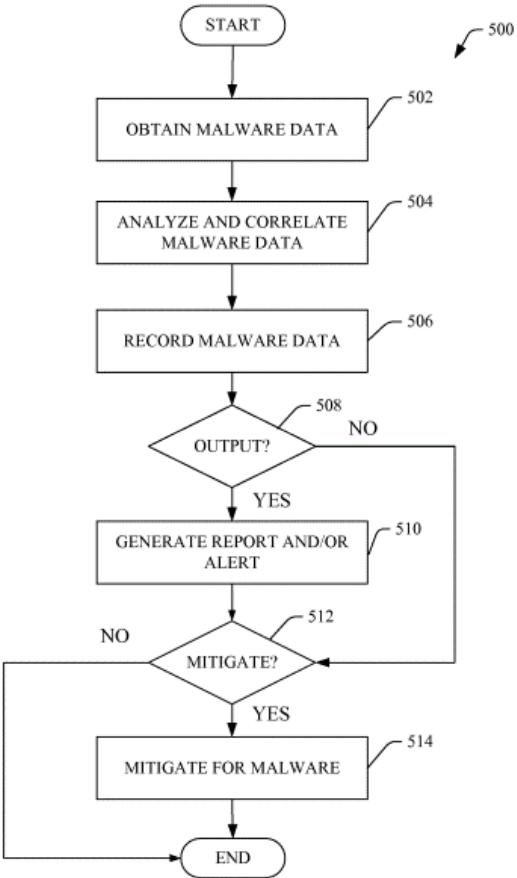
Claim	Exemplary Citation from Tuvell
	 <p data-bbox="646 251 1344 803">The diagram illustrates a Network Management System (100). It includes a User Interface (108) at the top, a Receiver Component (102) on the left, an Analysis Component (104) in the center, a Mitigation Component (110) on the right, and a Malware Data Store (106) at the bottom. All components are interconnected within the system boundary.</p> <p data-bbox="991 1084 1117 1128">FIG. 1</p>

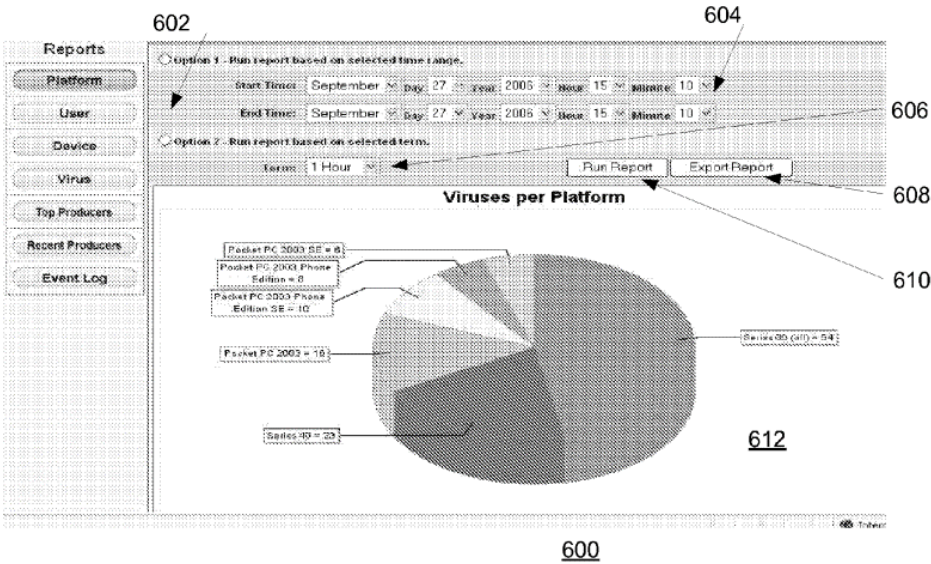
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) is connected to the system. An Analysis Component (104) is also present. A Malware Data Store (106) is shown as a database. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

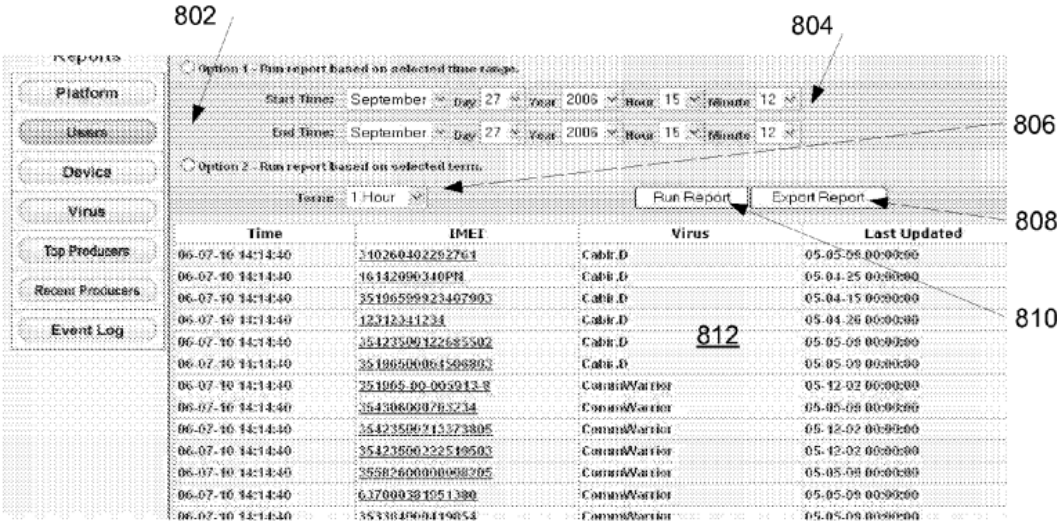
Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] MD --> IR IR --> NMS NMS --> ACK ACK --> MD </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

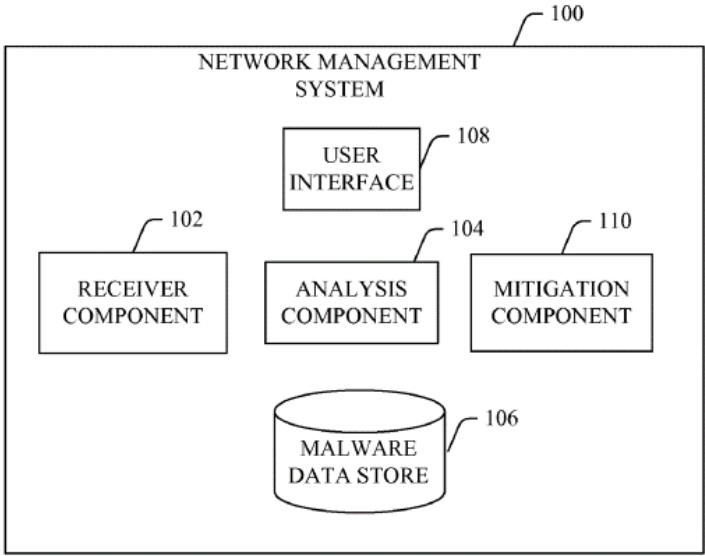
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[1C] determining if malware is present in the TCP session and an associated malware ID by comparing a malware</p>	<p>Tuvell discloses and/or renders obvious determining if malware is present in the TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets.</p> <p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained</p>

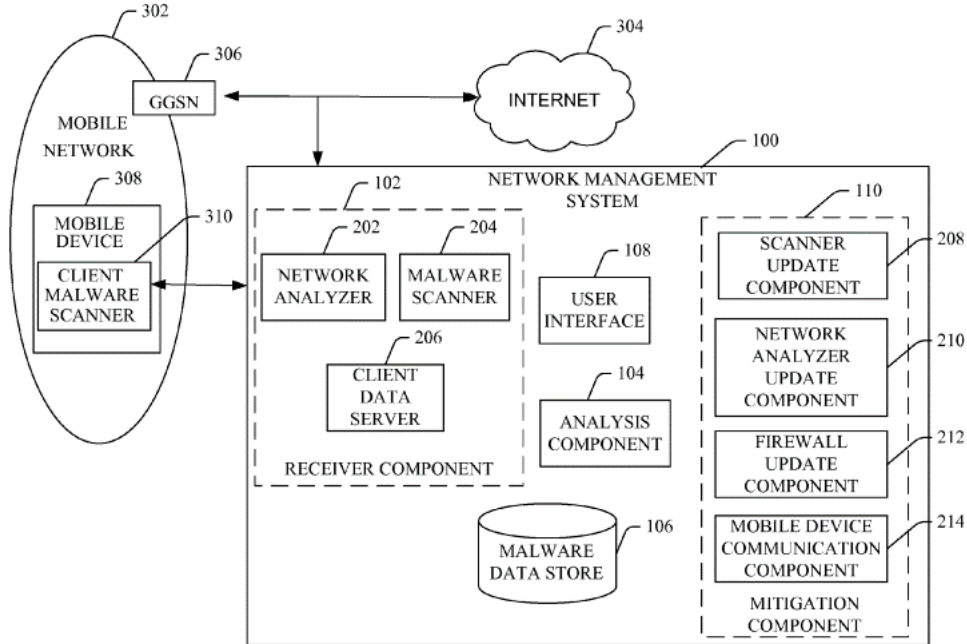
Claim	Exemplary Citation from Tuvell
signature to the one or more TCP packets; and	<p>from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

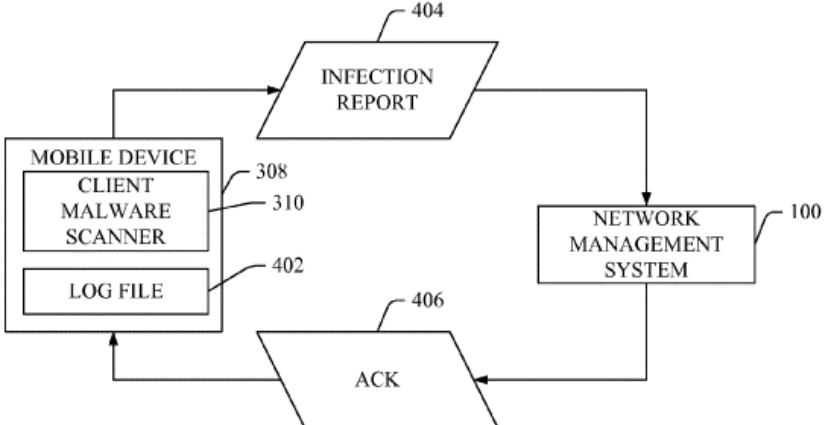
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <pre> graph TD 100[NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 108 --- 100 102 --- 100 104 --- 100 110 --- 100 106 --- 100 </pre> <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 26: "In another embodiment, the receiver component 102 obtains malware data from a plurality of sources, such as individual mobile devices, mobile network traffic and/or computer network traffic (e.g., Internet Protocol (IP) packets). Malware data obtained from multiple sources provides a more</p>

Claim	Exemplary Citation from Tuvell
	<p>complete picture of the current state of a mobile network. Consequently, collection or receipt of information from a variety of sources facilitates the detection and analysis of the spread of malware."</p> <p>Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104 generates a malware analysis or report that details and describes instances of malware in the mobile network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 30: "In an embodiment, the mitigation component 110 uses this information to dynamically change the parameters of the scanning algorithms utilized to detect the presence of malware either in network traffic or on individual mobile devices and to modify the malware detection algorithms used to identify malware. Some representative malware algorithms include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647 "Malware Detection System and Method for Mobile Platforms"; and malware detection in headers and compressed parts of mobile messages as described in U.S. patent application Ser. No. 11/697,658 "Malware Detection System and Method for Compressed Data on Mobile Platforms"."</p>

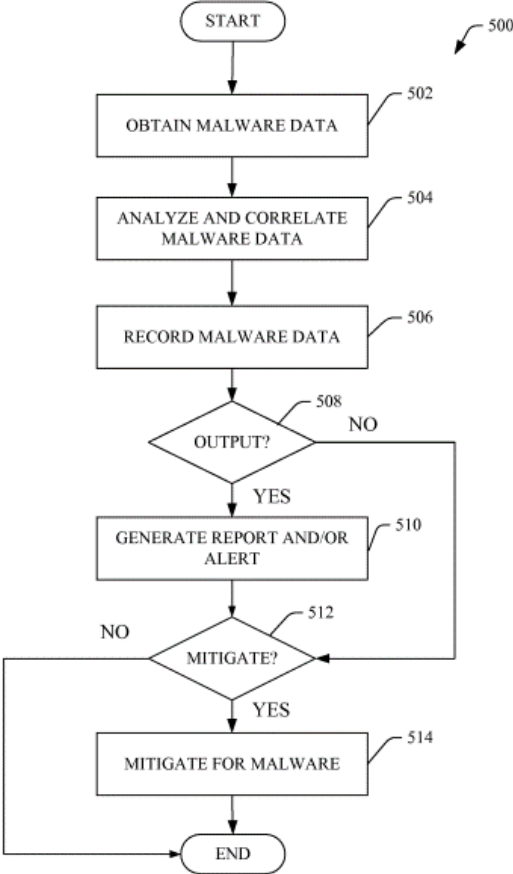
Claim	Exemplary Citation from Tuvell
	<p data-bbox="598 240 1102 267">Tuvell at Fig. 2 and corresponding text:</p> <div data-bbox="646 267 1407 836"> <p>The diagram illustrates a Network Management System (100) with the following components and connections:</p> <ul style="list-style-type: none"> Receiver Component (102): A dashed box containing: <ul style="list-style-type: none"> Network Analyzer (202) and Malware Scanner (204) connected to a Client Data Server (206). User Interface (108) and Analysis Component (104) are connected to the Receiver Component (102). Malware Data Store (106) is connected to the Analysis Component (104). Update/Mitigation Components (110): A dashed box containing: <ul style="list-style-type: none"> Scanner Update Component (208), Network Analyzer Update Component (210), Firewall Update Component (212), Mobile Device Communication Component (214), and Mitigation Component. </div> <p data-bbox="966 1071 1081 1112" style="text-align: center;">FIG. 2</p> <p data-bbox="598 1161 787 1193">Tuvell at ¶ 35:</p> <p data-bbox="598 1201 1890 1404">"In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 37: "The analysis component 104 processes or analyzes malware data received via the client data server 206, the malware scanner 204, and/or any other source. Malware may use a variety of techniques to spread and may even be designed to avoid detection. Monitoring a plurality of sources increases the likelihood of early detection of malware, before infection becomes widespread. In addition, use of data from multiple sources, as well as, historical data retrieved from the malware data store 106, increases accuracy of the malware analysis. The resulting malware analysis is stored in a malware data store 106 and/or presented to operators via the user interface 108."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

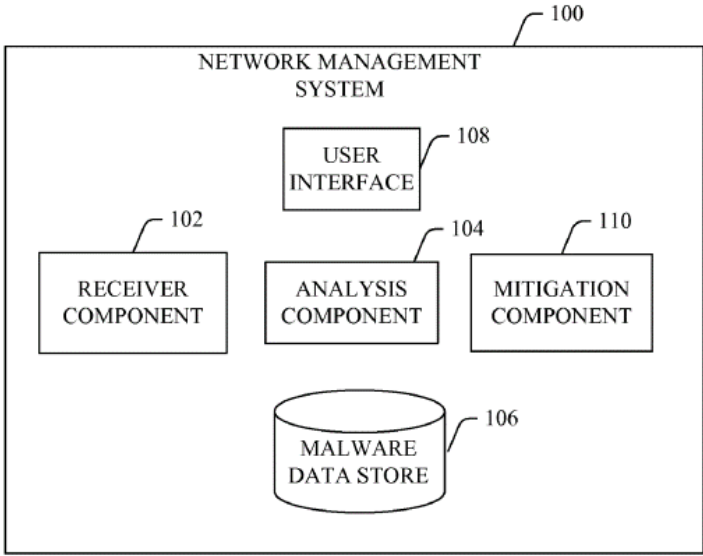
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD MD[MOBILE DEVICE 308] --> IR[/INFECTION REPORT 404/] IR --> NMS[NETWORK MANAGEMENT SYSTEM 100] NMS --> ACK[/ACK 406/] ACK --> MD subgraph MD_Box [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end </pre> <p>Tuvell at ¶ 43-44:</p> <p>"Referring now to FIG. 3, an exemplary deployment of a sample CoreStats system 100 in a network environment is depicted. FIG. 3 illustrates a deployment of CoreStats 100 between the edge of a mobile network 302 and the Internet 304, although it can also be deployed effectively at various other points in the mobile network 302 depending upon the network topology and desired coverage. The network analyzer 202 monitors and evaluates all traffic going from the mobile network 302 to the public data networks (e.g., the Internet 304) and vice-versa. The network analyzer 202 can intercept packets on either side of a firewall (not shown). In an embodiment, CoreStats 100 monitors a mobile network (or operator's network) 302 by monitoring or packet sniffing IP packets passing from the Gateway General Packet Radio Service ("GPRS") Support Node or Gateway GPRS Support Node ("GGSN") 306 and the Internet 304. In an embodiment, CoreStats 100 is deployed between the edge of the mobile network 302 and the Internet 304. The GGSN 306 links the access dependent Radio Access Network (RAN), shown on the figure as the mobile network 302, to the access independent Internet 304. RAN comprises the entire radio/wireless network with a variety of protocols for data transfer (e.g., CDMA, GPRS, 802.11). The GGSN 306 acts as a gateway between</p>

Claim	Exemplary Citation from Tuvell
	<p>the mobile network 302 and the Internet 304, converting access-specific packet data to IP packets and vice-versa. As discussed above, the intercepted packets are processed by the malware scanner 204 and the resulting malware data is provided to the analysis component 104."</p> <p>Tuvell at ¶ 45: "In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46: "Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p>

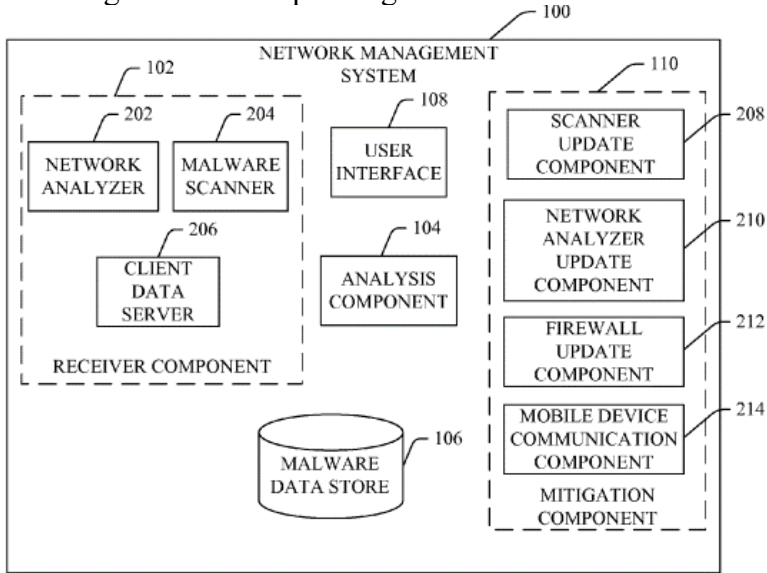
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 510 --> 512{MITIGATE?} 508 -- NO --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 514 --> END([END]) 512 -- NO --> END </pre> <p>FIG. 5</p> <p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the</p>

Claim	Exemplary Citation from Tuvell
	<p>malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 59: "At reference number 504 , the malware data is analyzed and/or correlated. An analysis component 104 generates a malware analysis and/or statistics describing malware activity as well as other pertinent network statistics useful in quantifying relative levels of malware activity."</p>
<p>[1D] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.</p>	<p>Tuvell discloses and/or renders obvious generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.</p> <p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p>

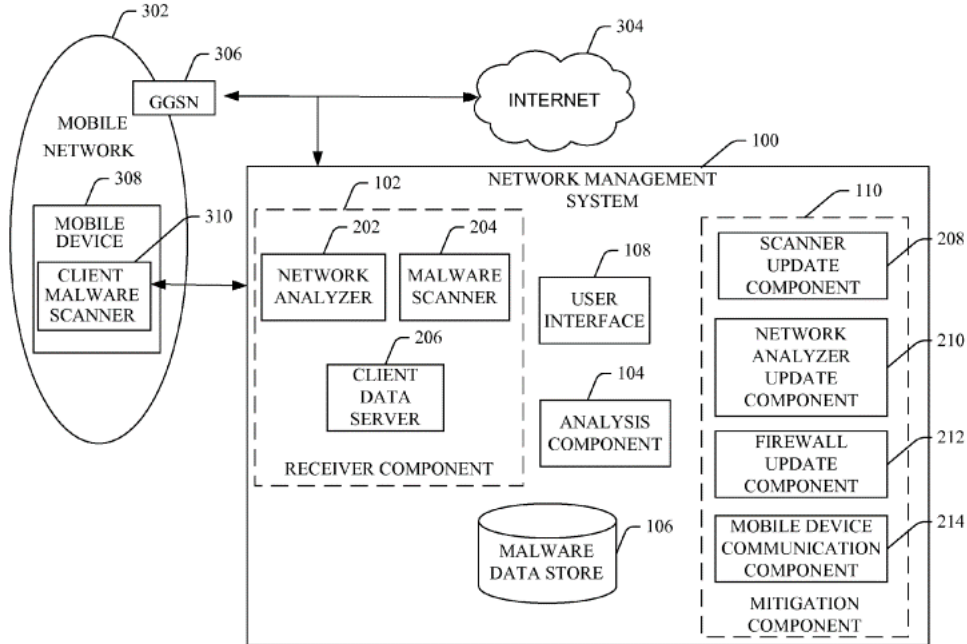
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 9:</p> <p>"The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

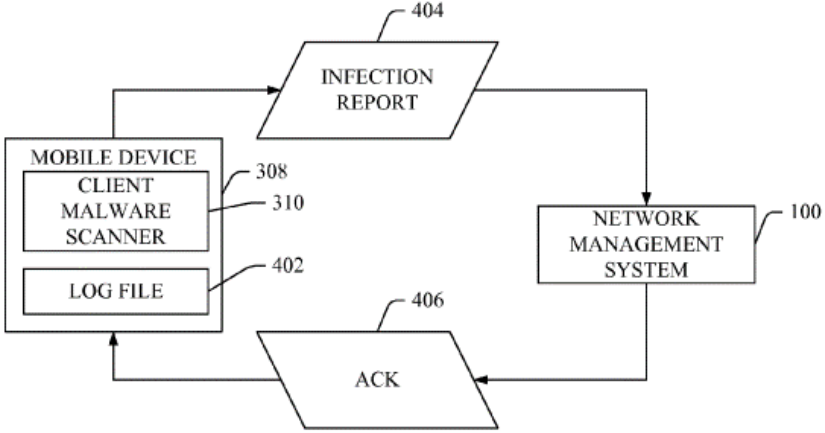
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104</p>

Claim	Exemplary Citation from Tuvell
	<p>generates a malware analysis or report that details and describes instances of malware in the mobile network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 31: "One way is to associate CoreStats 100 with the mobile network administrator's firewall so that the administrator can block identified malicious content. Another way is to report alarms upstream to operational support systems or OAM&P (Operations, Administration, Maintenance, and Provisioning) systems used by network service providers to manage mobile networks."</p> <p>Tuvell at ¶ 32: "In another embodiment, CoreStats 100 facilitates malware prevention by informing mobile device users and/or taking preventative steps at the mobile device. Once CoreStats 100 identifies an infected user, network administrators or operators send messages to a user to alert them to the problem, force an update of the user's mobile device's anti-virus software and definitions, or even disable the mobile device's data connections altogether."</p>

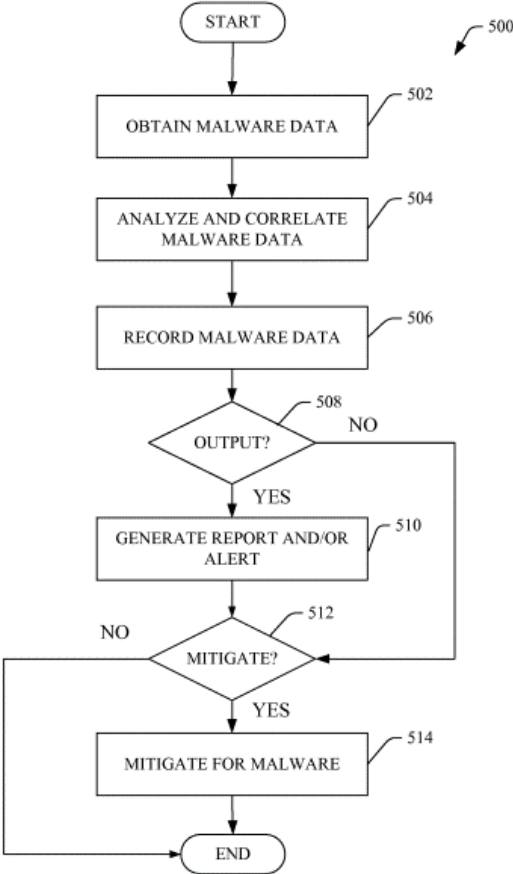
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) and an Analysis Component (104) are also shown. A Malware Data Store (106) is connected to the system. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 35: "In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p>

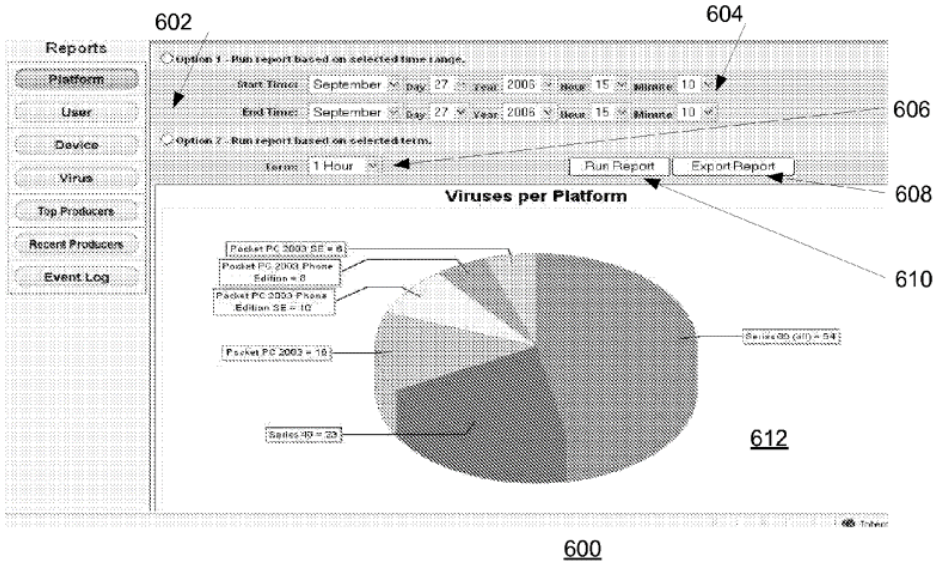
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

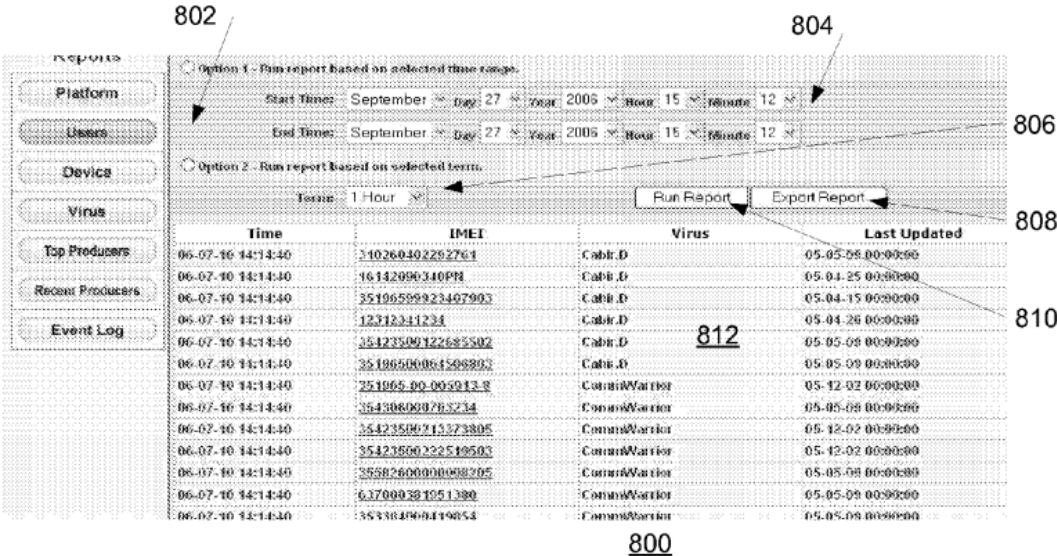
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph LR subgraph 308 [MOBILE DEVICE] 310[CLIENT MALWARE SCANNER] 402[LOG FILE] end 404[/INFECTION REPORT/] 100[NETWORK MANAGEMENT SYSTEM] 406[/ACK/] 310 --> 404 404 --> 100 100 --> 406 406 --> 308 </pre> <p>Tuvell at ¶ 45:</p> <p>"In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46:</p> <p>"Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p>

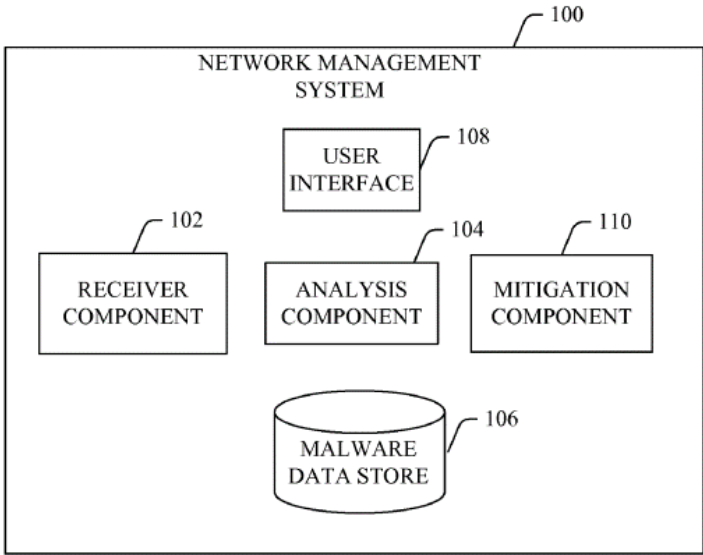
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 48: "In a further embodiment, the client malware scanner 310 generates an infection report 404 that contains information about the detected malware and transmits the infection report 404 to the client data server 206 of CoreStats 100. Report generation transmission is automatically triggered (pushed) upon detection of malware or based upon a periodic fixed time interval. Alternatively, infection reports 404 are maintained in the client device internal log file 402 until queried (pulled) by CoreStats 100. In yet another embodiment, infection reports 404 are delivered to CoreStats 100 using some combination of pulling and pushing. Infection reports 404 are transmitted, for example, using hypertext transfer protocol (http), file transfer protocol (ftp), or any packet data transmission method as would be generally known in the art."</p> <p>Tuvell at ¶ 49: "Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g.,</p>

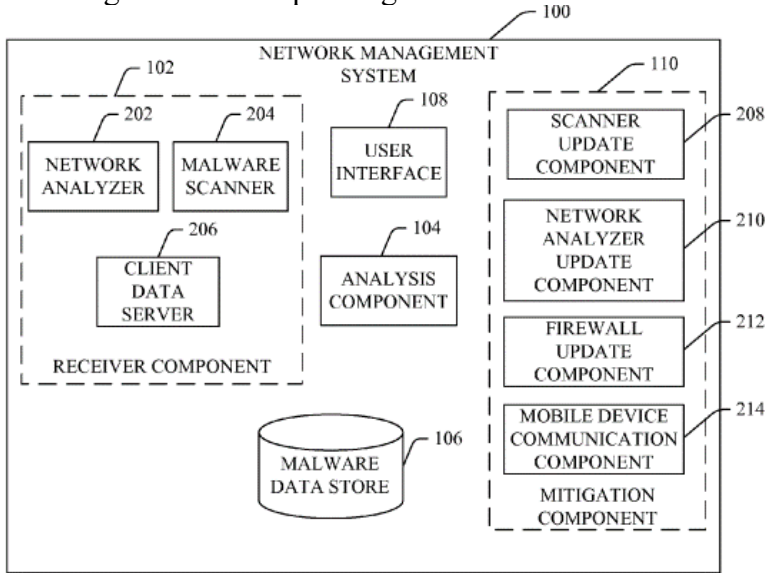
Claim	Exemplary Citation from Tuvell
	<p>phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 510 --> 512{MITIGATE?} 508 -- NO --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 514 --> END([END]) 512 -- NO --> END </pre> <p>FIG. 5</p>

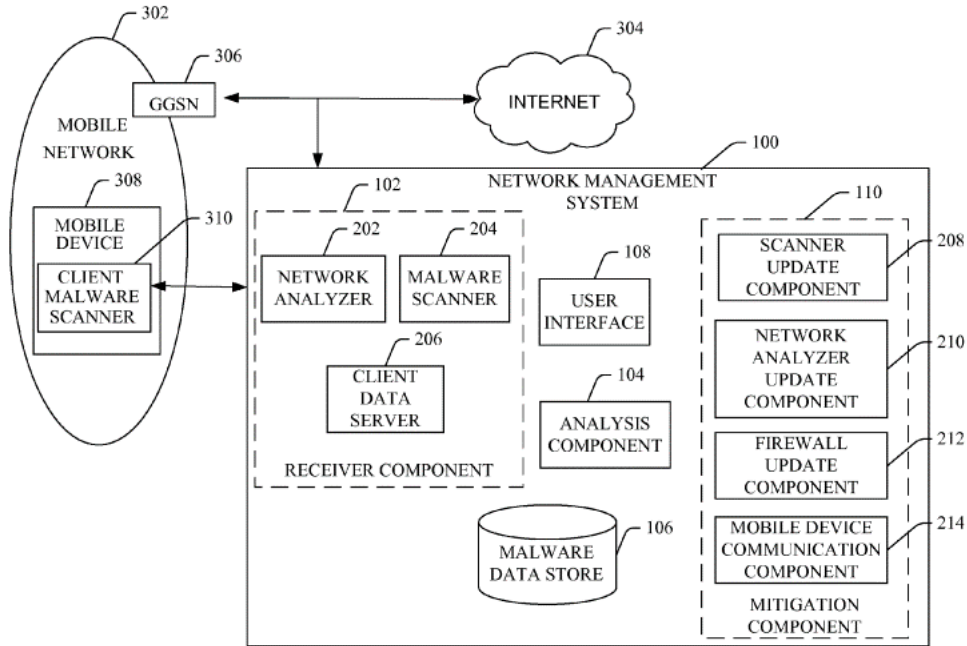
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 60: "At reference number 508 a determination is made as to whether to generate output, such as a report or alert. The determination can be based in whole or in part upon the malware data obtained from various sources. For example, if analysis indicates high levels of malware activity or significant impact on mobile network 302 performance, the determination is made to generate a report and alert or notify network administrators. Alternatively, reports are triggered periodically or upon operator request. In particular, operators can request particular reports via a user interface 108."</p> <p>Tuvell at ¶ 62: "In still other embodiments, a mitigation component 110 notifies a mobile device 308 user, force an update of mobile device 308 software, or even disable the mobile device's 308 data connections."</p> <p>Tuvell at ¶ 63: "Referring now to FIGS. 6-10, exemplary user interface displays are illustrated. As discussed above, CoreStats 100 also performs report generating functions. The analysis component 104 uses both stored and real-time information, including network traffic and individual user information, to generate statistics and dynamic graphs depicting malware activity and network statistics necessary to quantify relative levels of malware activity. For example, the analysis component 104 generates malware analyses, which can be presented by a user interface 108 as straightforward visual reports to alert managers and operators as to which platforms are infected with the most viruses, which viruses are spreading the fastest, the most recently infected mobile devices 308, and which infected mobile devices 308 are spreading the most viruses."</p>

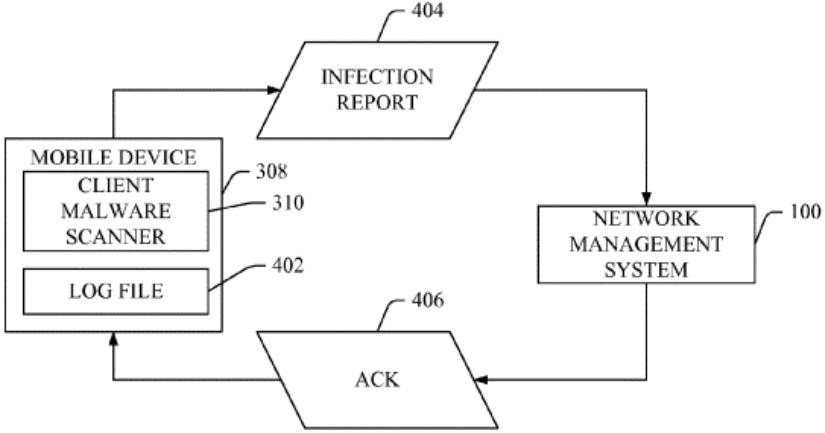
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p>FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
[2PRE] The method of claim 1 wherein determining the operating system ID further comprises:	See claim 1.

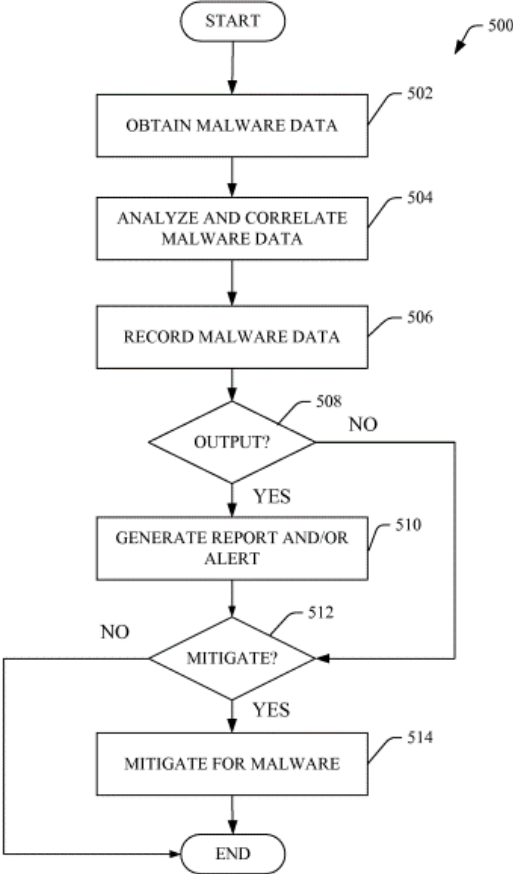
Claim	Exemplary Citation from Tuvell
<p>[2A] determining a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first operating system identifier (ID);</p>	<p>Tuvell discloses and/or renders obvious determining a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first operating system identifier (ID).</p> <p>Tuvell at Fig. 1 and corresponding text:</p>  <pre> graph TD 100[NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 108 --- 100 102 --- 100 104 --- 100 110 --- 100 106 --- 100 </pre> <p style="text-align: center;">FIG. 1</p>

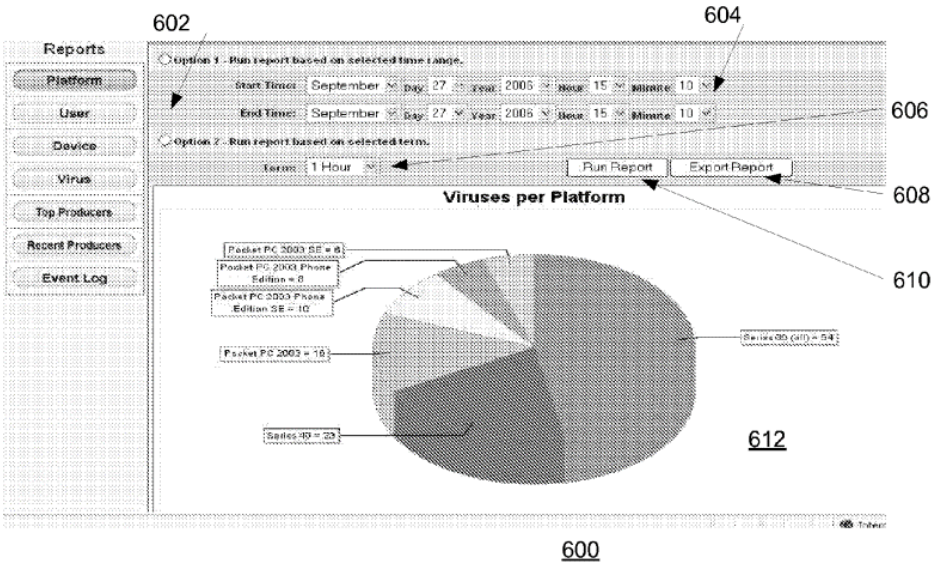
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) is connected to the system. An Analysis Component (104) is also present. A Malware Data Store (106) is shown as a database. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

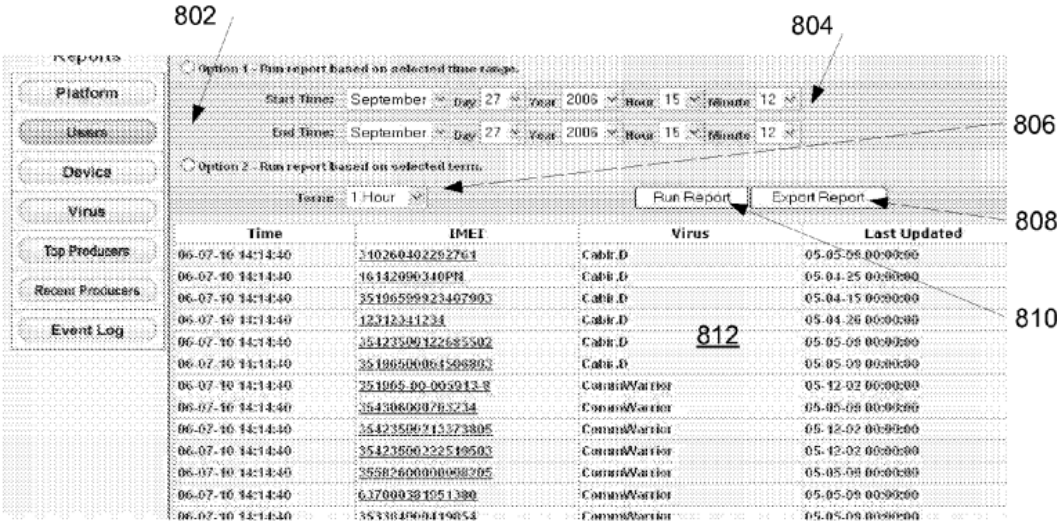
Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

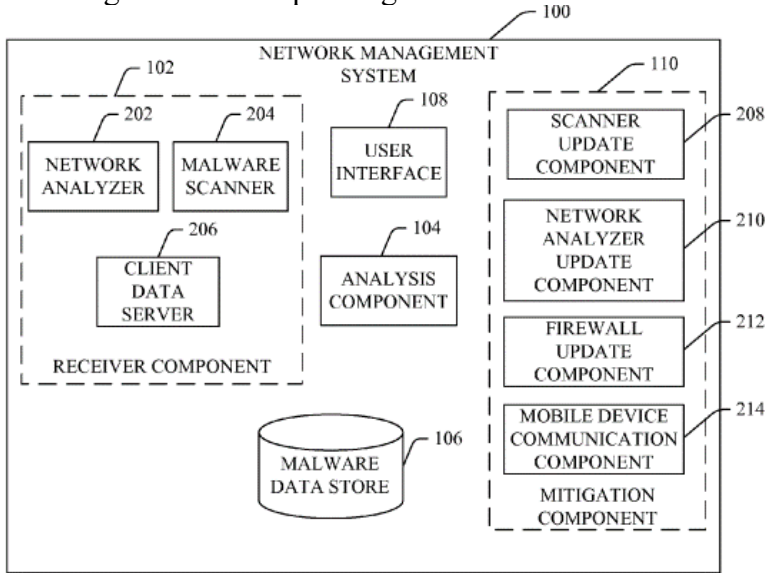
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

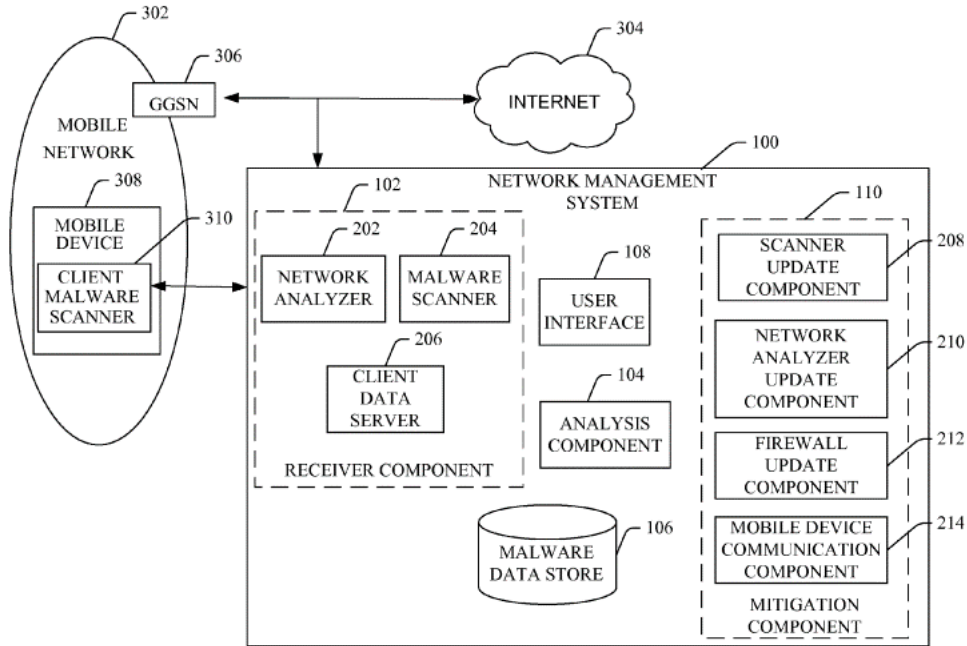
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

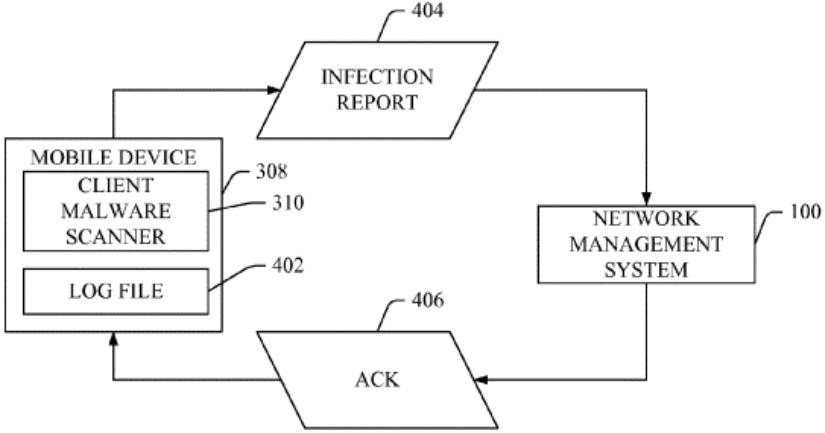
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[2B] determining an application parameter associated with the TCP session if present, and identifying a second OS ID</p>	<p>Tuvell discloses and/or renders obvious determining an application parameter associated with the TCP session if present, and identifying a second OS ID from the application parameter.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

Claim	Exemplary Citation from Tuvell
<p>from the application parameter; and</p>	<div data-bbox="646 251 1344 803"> <pre> graph TD 100[NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 108 --- 100 102 --- 100 104 --- 100 110 --- 100 106 --- 100 </pre> <p>The diagram illustrates a Network Management System (100) enclosed in a rectangular box. At the top center of the box is the label "NETWORK MANAGEMENT SYSTEM". Below this label, there are five components: a "USER INTERFACE" (108) at the top center, a "RECEIVER COMPONENT" (102) on the left, an "ANALYSIS COMPONENT" (104) in the center, a "MITIGATION COMPONENT" (110) on the right, and a "MALWARE DATA STORE" (106) represented by a cylinder at the bottom center. Each component is connected to the system label by a thin line.</p> </div> <p style="text-align: center;">FIG. 1</p>

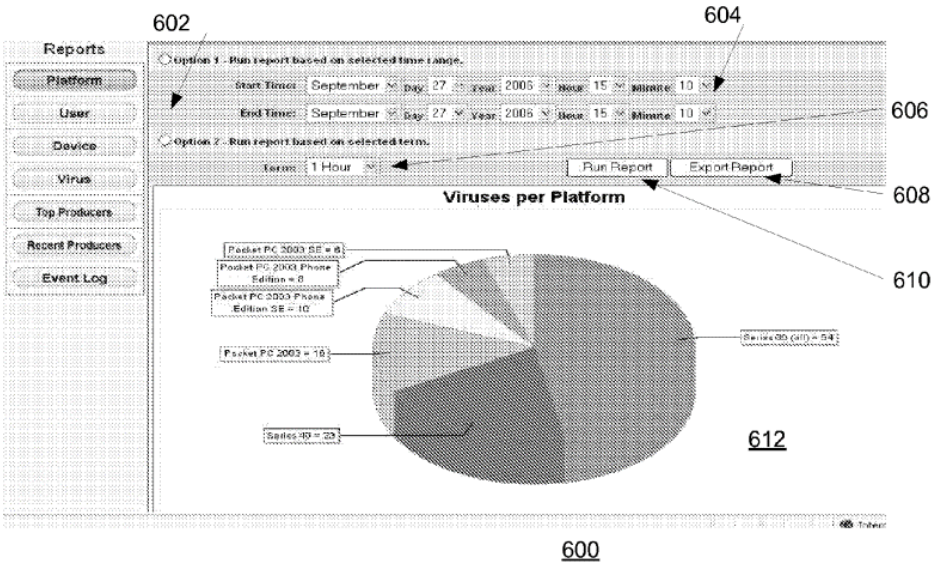
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) is connected to the system. An Analysis Component (104) is also present. A Malware Data Store (106) is shown as a database. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

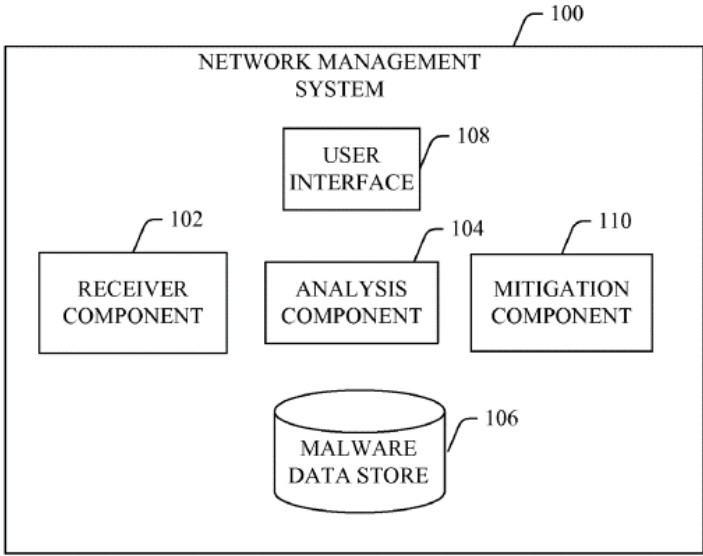
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

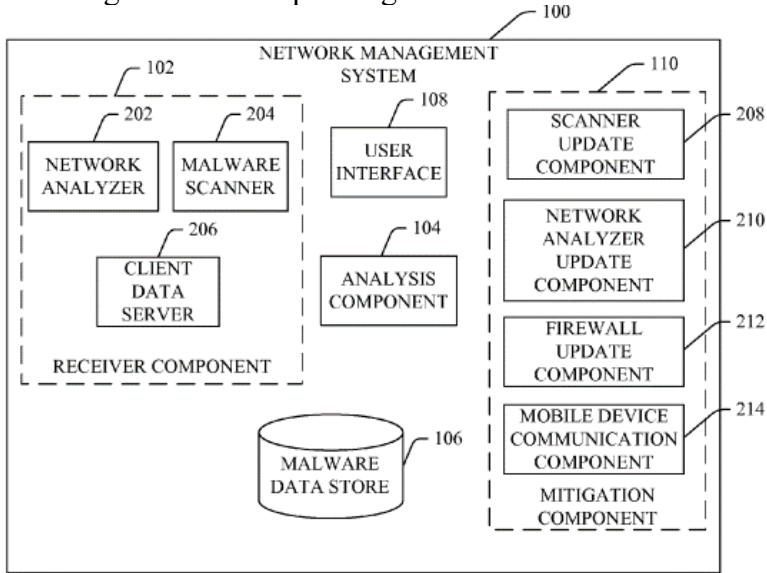
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p> <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

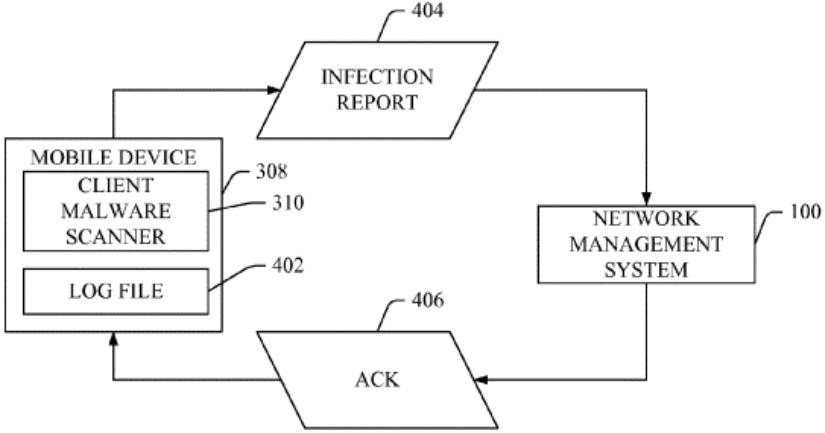
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p> <p>802</p> <p>804</p> <p>806</p> <p>808</p> <p>810</p> <p>812</p> <p>800</p> <p>FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
[2C] wherein the OS ID in the alert comprises one of the first OS ID or second OS ID associated with one or more computing devices	Tuvell discloses and/or renders obvious that the OS ID in the alert comprises one of the first OS ID or second OS ID associated with one or more computing devices coupled to the access device.

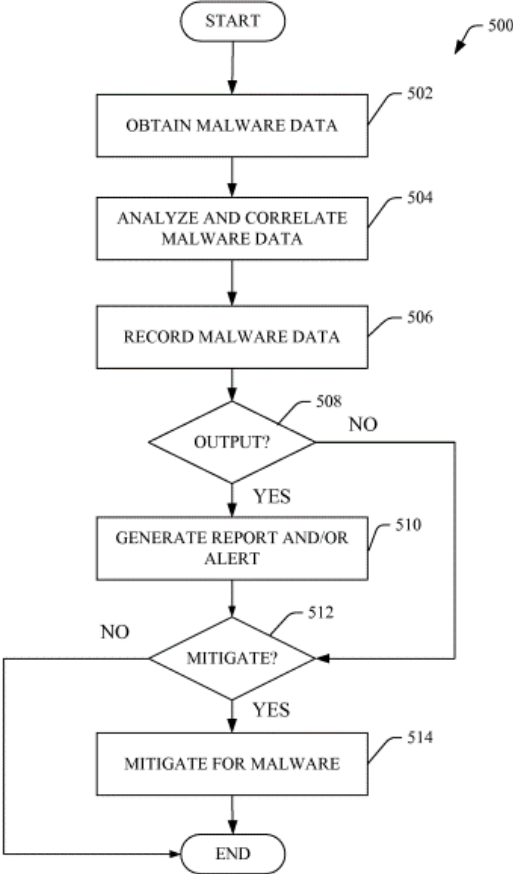
Claim	Exemplary Citation from Tuvell
<p>coupled to the access device.</p>	<p>Tuvell at Fig. 1 and corresponding text:</p>  <pre> graph TD subgraph 100 [NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] end </pre> <p>FIG. 1</p>

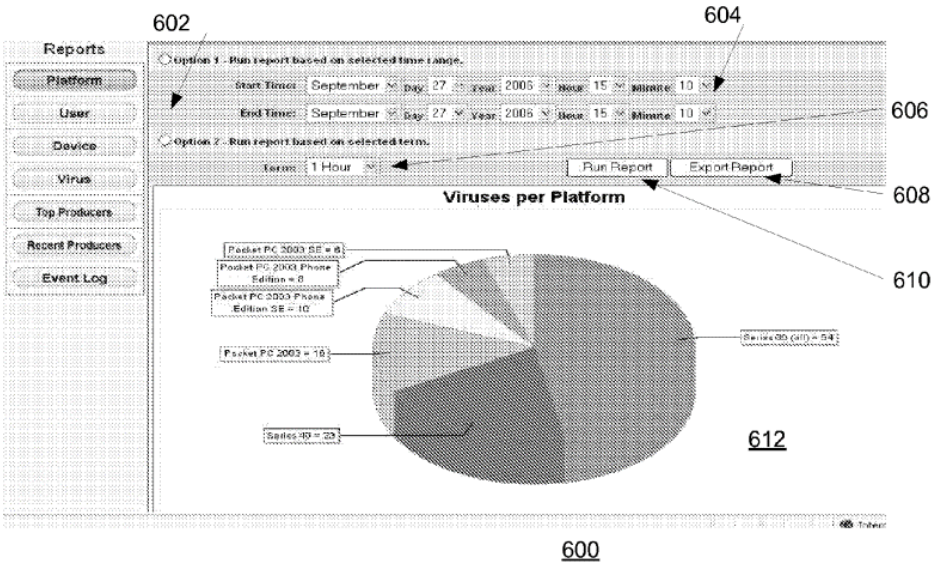
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) and an Analysis Component (104) are also shown. A Malware Data Store (106) is connected to the system. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

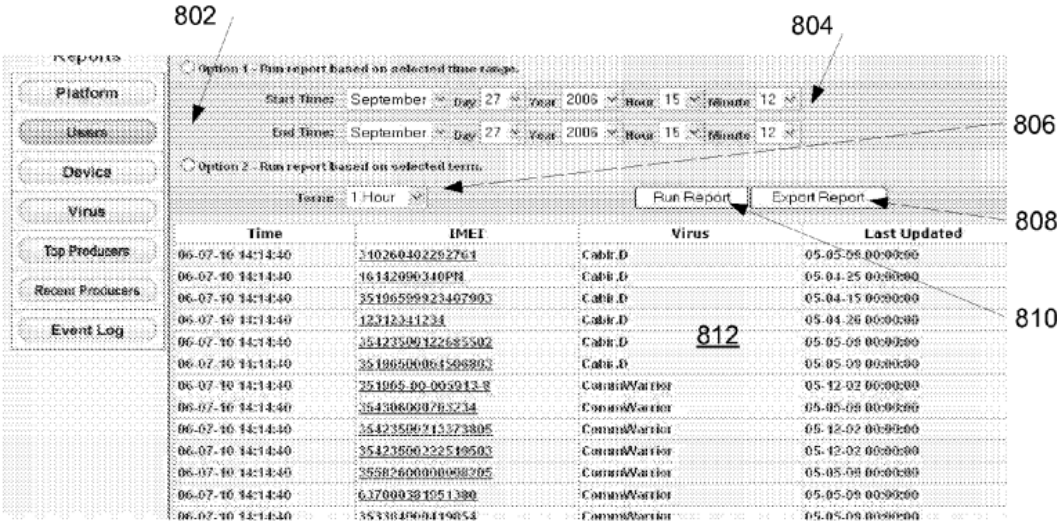
Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p> <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

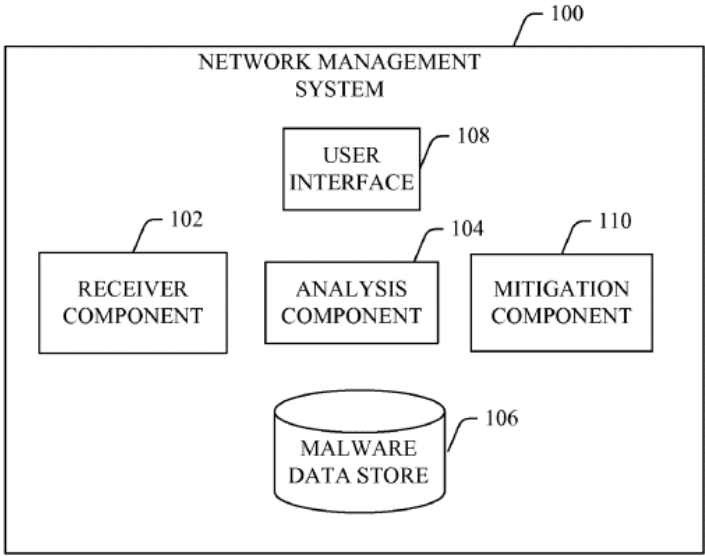
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

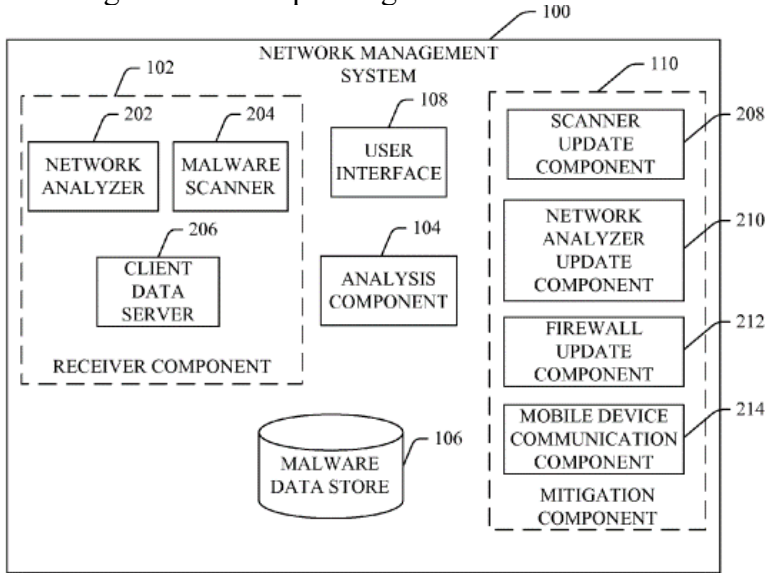
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[3] The method of claim 2 further comprising sending a notification to a subscriber associated with the network address associated with the access device, the</p>	<p>Tuvell discloses and/or renders obvious sending a notification to a subscriber associated with the network address associated with the access device, the notification identifying a remediation portal to remove the determined malware in the alert.</p>

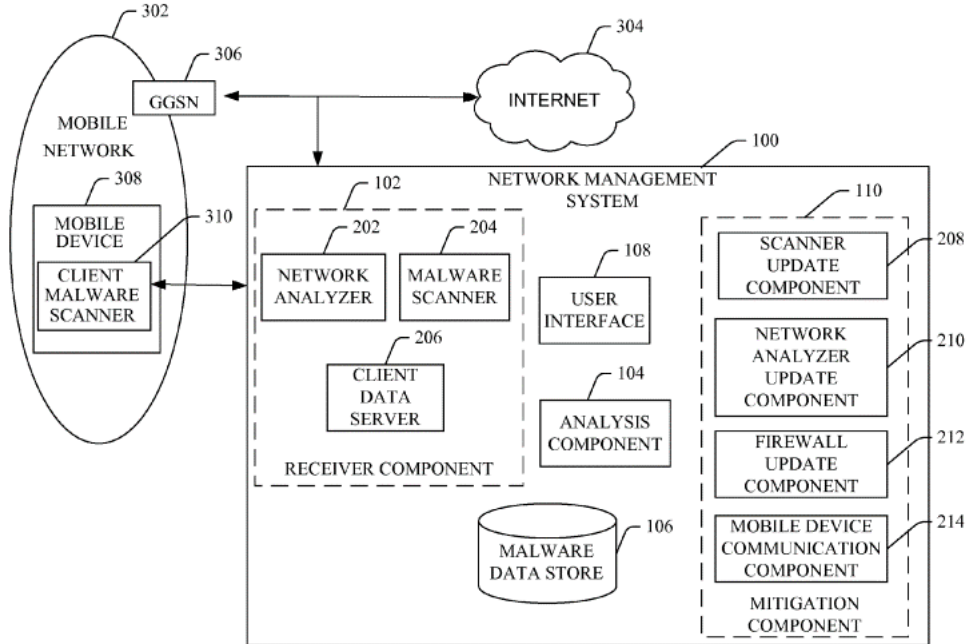
Claim	Exemplary Citation from Tuvell
<p>notification identifying a remediation portal to remove the determined malware in the alert.</p>	<p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

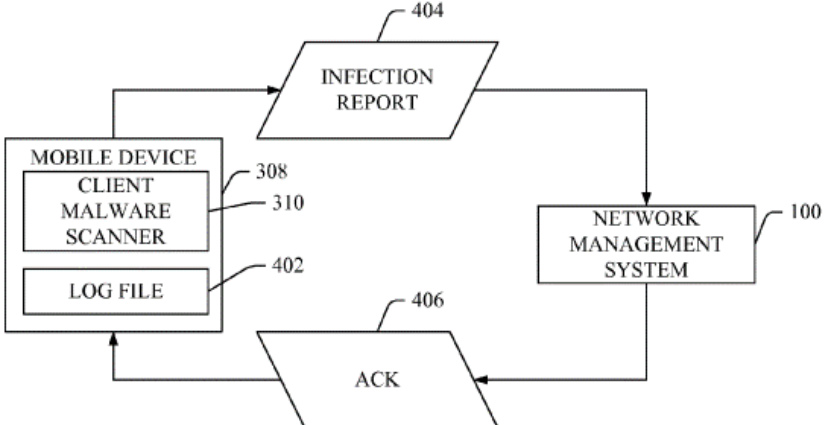
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104</p>

Claim	Exemplary Citation from Tuvell
	<p>generates a malware analysis or report that details and describes instances of malware in the mobile network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 31: "One way is to associate CoreStats 100 with the mobile network administrator's firewall so that the administrator can block identified malicious content. Another way is to report alarms upstream to operational support systems or OAM&P (Operations, Administration, Maintenance, and Provisioning) systems used by network service providers to manage mobile networks."</p> <p>Tuvell at ¶ 32: "In another embodiment, CoreStats 100 facilitates malware prevention by informing mobile device users and/or taking preventative steps at the mobile device. Once CoreStats 100 identifies an infected user, network administrators or operators send messages to a user to alert them to the problem, force an update of the user's mobile device's anti-virus software and definitions, or even disable the mobile device's data connections altogether."</p>

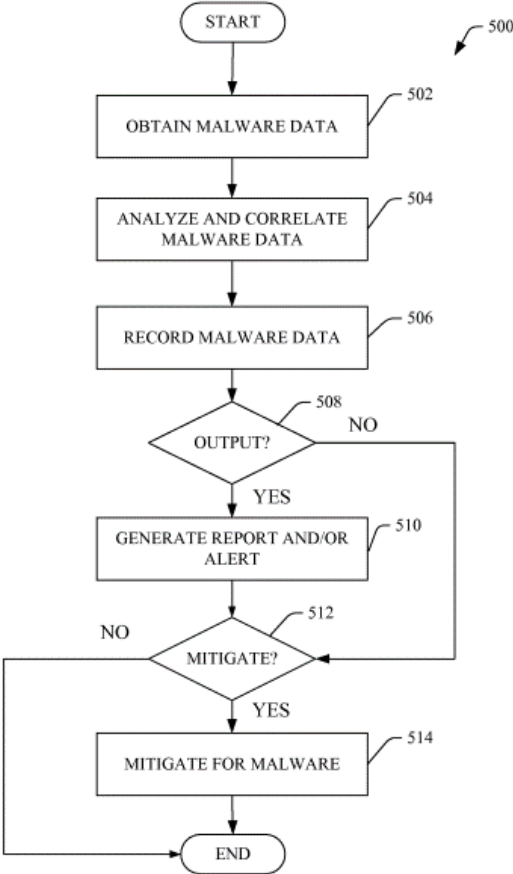
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) which contains a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) is connected to the system. An Analysis Component (104) is also present. A Malware Data Store (106) is shown as a database. A Mitigation Component (110) is shown with several sub-components: a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 35: "In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p>

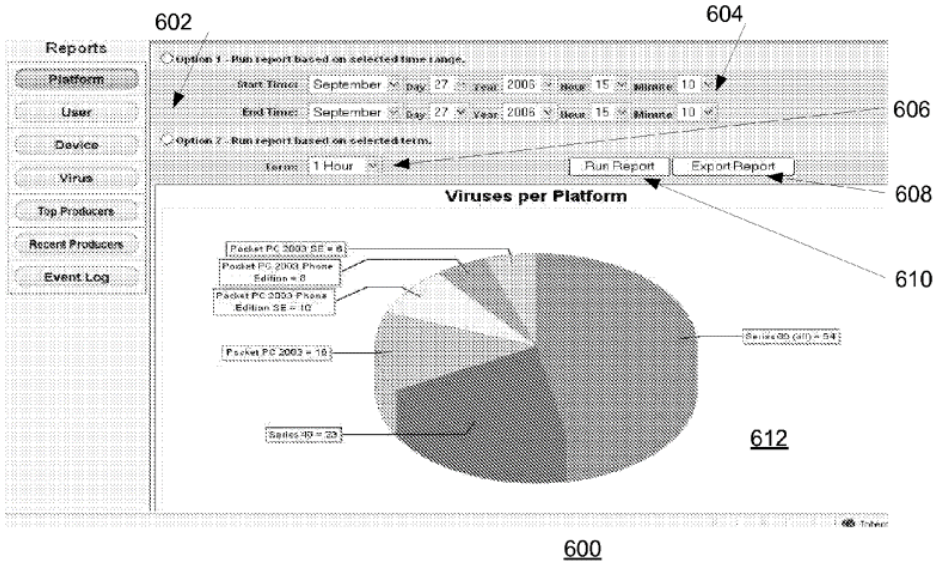
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph LR subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> MD </pre> <p>Tuvell at ¶ 45:</p> <p>"In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46:</p> <p>"Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 48: "In a further embodiment, the client malware scanner 310 generates an infection report 404 that contains information about the detected malware and transmits the infection report 404 to the client data server 206 of CoreStats 100. Report generation transmission is automatically triggered (pushed) upon detection of malware or based upon a periodic fixed time interval. Alternatively, infection reports 404 are maintained in the client device internal log file 402 until queried (pulled) by CoreStats 100. In yet another embodiment, infection reports 404 are delivered to CoreStats 100 using some combination of pulling and pushing. Infection reports 404 are transmitted, for example, using hypertext transfer protocol (http), file transfer protocol (ftp), or any packet data transmission method as would be generally known in the art."</p> <p>Tuvell at ¶ 49: "Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g.,</p>

Claim	Exemplary Citation from Tuvell
	<p>phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 60: "At reference number 508 a determination is made as to whether to generate output, such as a report or alert. The determination can be based in whole or in part upon the malware data obtained from various sources. For example, if analysis indicates high levels of malware activity or significant impact on mobile network 302 performance, the determination is made to generate a report and alert or notify network administrators. Alternatively, reports are triggered periodically or upon operator request. In particular, operators can request particular reports via a user interface 108."</p> <p>Tuvell at ¶ 62: "In still other embodiments, a mitigation component 110 notifies a mobile device 308 user, force an update of mobile device 308 software, or even disable the mobile device's 308 data connections."</p> <p>Tuvell at ¶ 63: "Referring now to FIGS. 6-10, exemplary user interface displays are illustrated. As discussed above, CoreStats 100 also performs report generating functions. The analysis component 104 uses both stored and real-time information, including network traffic and individual user information, to generate statistics and dynamic graphs depicting malware activity and network statistics necessary to quantify relative levels of malware activity. For example, the analysis component 104 generates malware analyses, which can be presented by a user interface 108 as straightforward visual reports to alert managers and operators as to which platforms are infected with the most viruses, which viruses are spreading the fastest, the most recently infected mobile devices 308, and which infected mobile devices 308 are spreading the most viruses."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p> <p>FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
[4PRE] The method of claim 3 further comprising:	See Claim 3.
[4A] identifying at the remediation portal the subscriber accessing the remediation portal;	<p>Tuvell discloses and/or renders obvious identifying at the remediation portal the subscriber accessing the remediation portal.</p> <p>Tuvell at Abstract:</p>

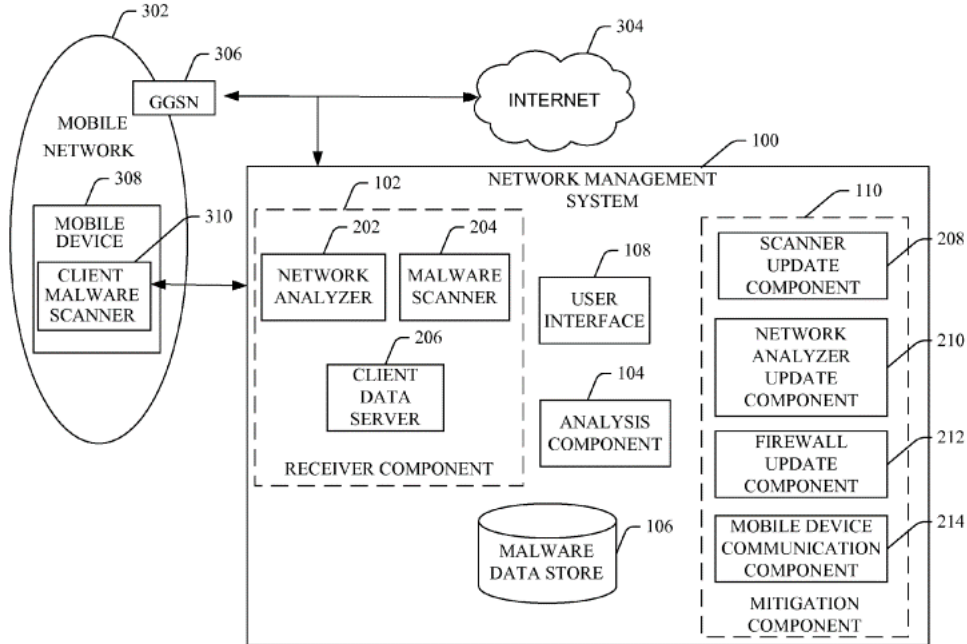
Claim	Exemplary Citation from Tuvell
	<p>"A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5:</p> <p>"Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9:</p> <p>"The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

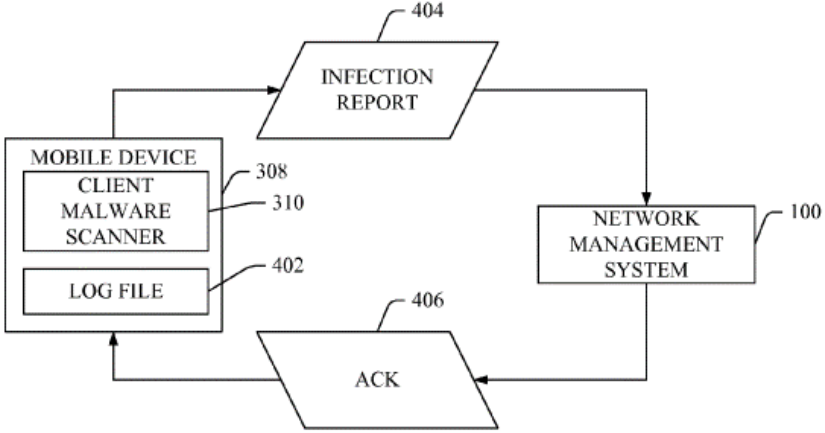
Claim	Exemplary Citation from Tuvell
	<div data-bbox="646 251 1344 803" data-label="Diagram"> <pre> graph TD 100[NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 102 --> 104 104 --> 110 104 <--> 106 </pre> </div> <p data-bbox="993 1084 1119 1128">FIG. 1</p> <p data-bbox="598 1198 1873 1414"> Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104 generates a malware analysis or report that details and describes instances of malware in the mobile </p>

Claim	Exemplary Citation from Tuvell
	<p>network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 31: "One way is to associate CoreStats 100 with the mobile network administrator's firewall so that the administrator can block identified malicious content. Another way is to report alarms upstream to operational support systems or OAM&P (Operations, Administration, Maintenance, and Provisioning) systems used by network service providers to manage mobile networks."</p> <p>Tuvell at ¶ 32: "In another embodiment, CoreStats 100 facilitates malware prevention by informing mobile device users and/or taking preventative steps at the mobile device. Once CoreStats 100 identifies an infected user, network administrators or operators send messages to a user to alert them to the problem, force an update of the user's mobile device's anti-virus software and definitions, or even disable the mobile device's data connections altogether."</p>

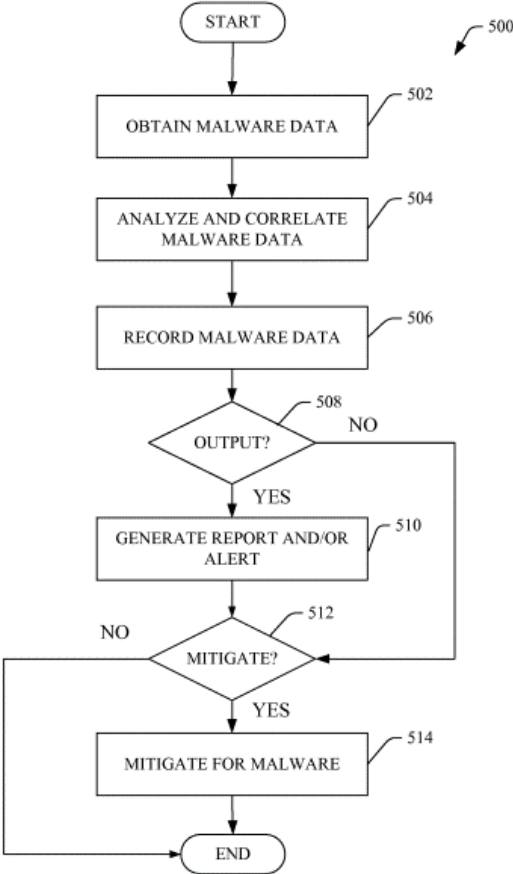
Claim	Exemplary Citation from Tuvell
	<p data-bbox="598 240 1102 267">Tuvell at Fig. 2 and corresponding text:</p> <div data-bbox="646 267 1407 836"> <p>The diagram illustrates a Network Management System (100) with the following components and connections:</p> <ul style="list-style-type: none"> Receiver Component (102): A dashed box containing: <ul style="list-style-type: none"> Network Analyzer (202) and Malware Scanner (204) connected to a Client Data Server (206). User Interface (108) and Analysis Component (104) are connected to the Receiver Component (102). Malware Data Store (106) is connected to the Analysis Component (104). Update and Mitigation Components (110): A dashed box containing: <ul style="list-style-type: none"> Scanner Update Component (208), Network Analyzer Update Component (210), Firewall Update Component (212), and Mobile Device Communication Component (214). Mitigation Component is located below the dashed box 110. </div> <p data-bbox="966 1071 1081 1112" style="text-align: center;">FIG. 2</p> <p data-bbox="598 1161 787 1193">Tuvell at ¶ 35:</p> <p data-bbox="598 1201 1890 1404">"In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p>

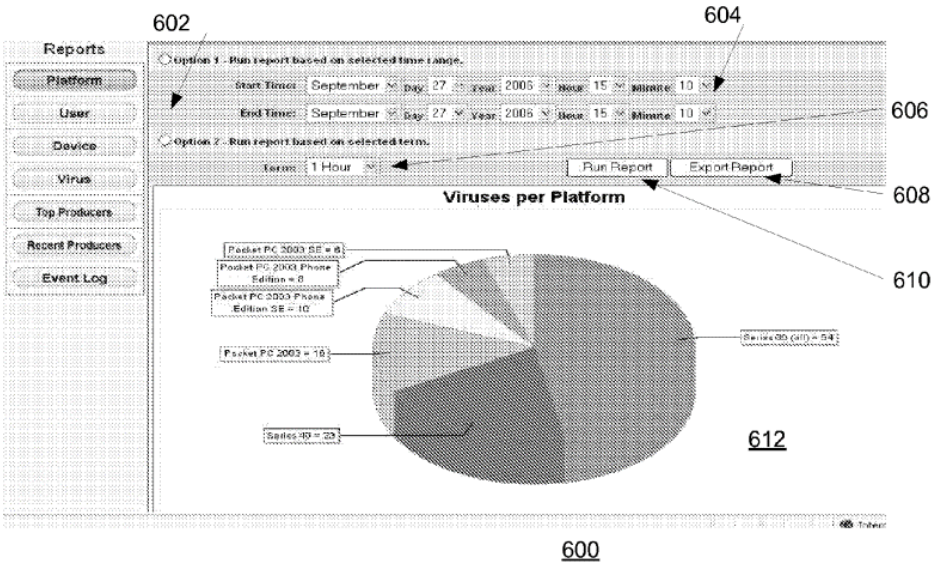
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

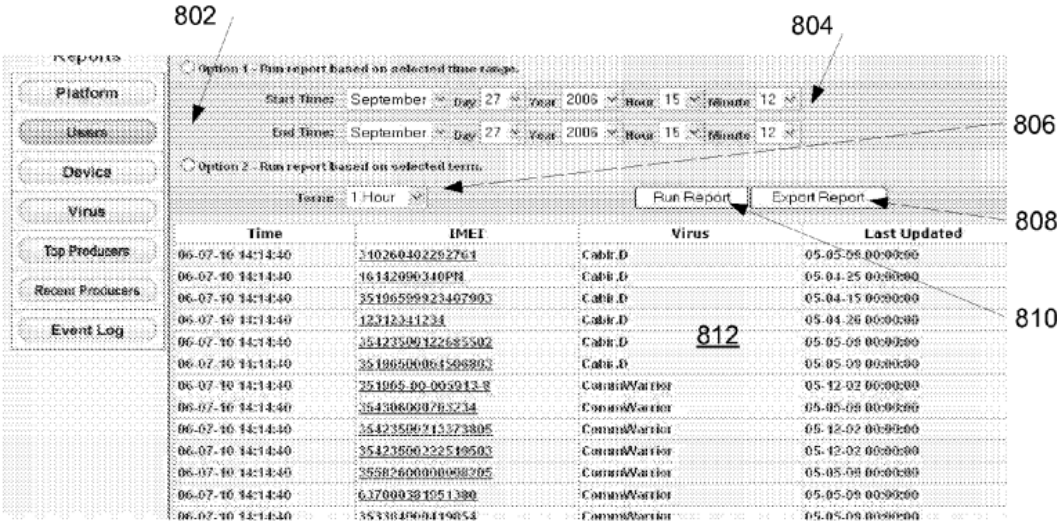
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph LR subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 45:</p> <p>"In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46:</p> <p>"Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p>

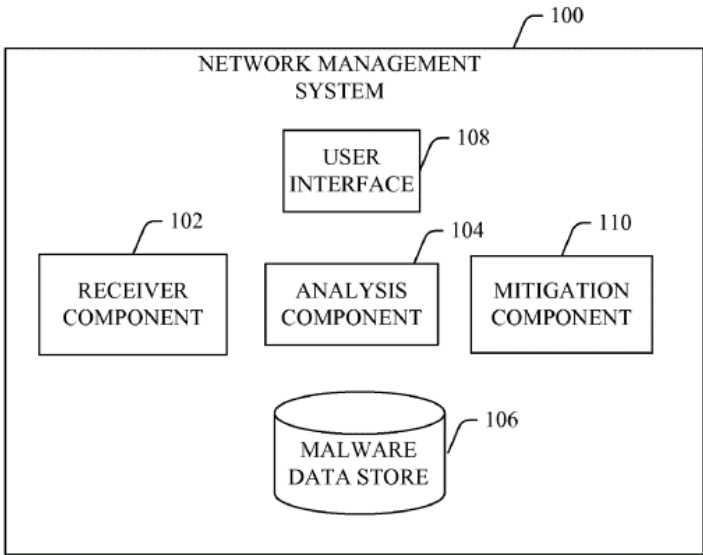
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 48: "In a further embodiment, the client malware scanner 310 generates an infection report 404 that contains information about the detected malware and transmits the infection report 404 to the client data server 206 of CoreStats 100. Report generation transmission is automatically triggered (pushed) upon detection of malware or based upon a periodic fixed time interval. Alternatively, infection reports 404 are maintained in the client device internal log file 402 until queried (pulled) by CoreStats 100. In yet another embodiment, infection reports 404 are delivered to CoreStats 100 using some combination of pulling and pushing. Infection reports 404 are transmitted, for example, using hypertext transfer protocol (http), file transfer protocol (ftp), or any packet data transmission method as would be generally known in the art."</p> <p>Tuvell at ¶ 49: "Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g.,</p>

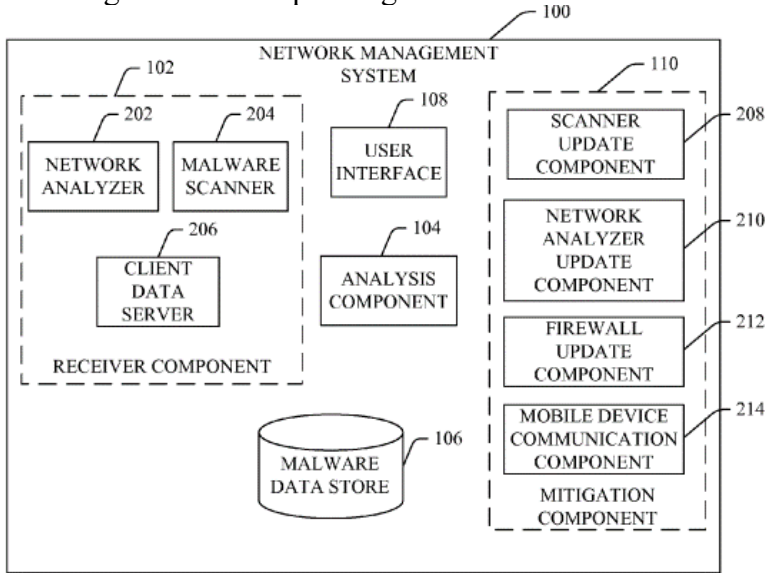
Claim	Exemplary Citation from Tuvell
	<p>phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

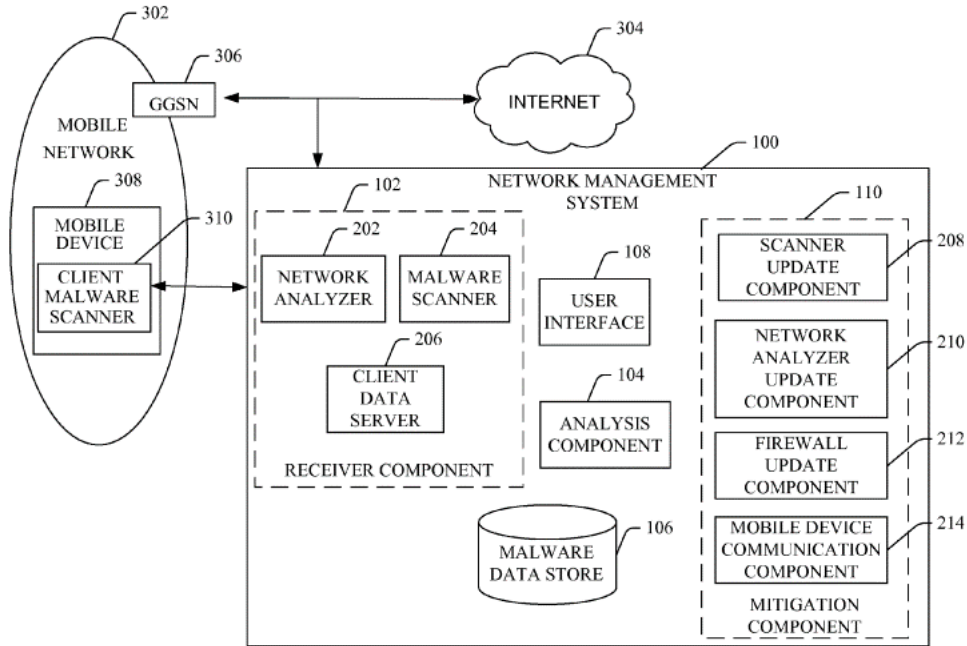
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 60: "At reference number 508 a determination is made as to whether to generate output, such as a report or alert. The determination can be based in whole or in part upon the malware data obtained from various sources. For example, if analysis indicates high levels of malware activity or significant impact on mobile network 302 performance, the determination is made to generate a report and alert or notify network administrators. Alternatively, reports are triggered periodically or upon operator request. In particular, operators can request particular reports via a user interface 108."</p> <p>Tuvell at ¶ 62: "In still other embodiments, a mitigation component 110 notifies a mobile device 308 user, force an update of mobile device 308 software, or even disable the mobile device's 308 data connections."</p> <p>Tuvell at ¶ 63: "Referring now to FIGS. 6-10, exemplary user interface displays are illustrated. As discussed above, CoreStats 100 also performs report generating functions. The analysis component 104 uses both stored and real-time information, including network traffic and individual user information, to generate statistics and dynamic graphs depicting malware activity and network statistics necessary to quantify relative levels of malware activity. For example, the analysis component 104 generates malware analyses, which can be presented by a user interface 108 as straightforward visual reports to alert managers and operators as to which platforms are infected with the most viruses, which viruses are spreading the fastest, the most recently infected mobile devices 308, and which infected mobile devices 308 are spreading the most viruses."</p>

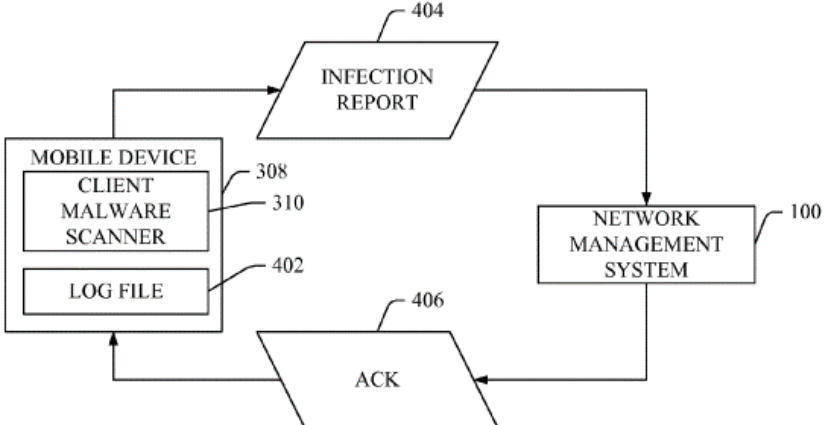
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[4B] identifying the OS associated with the computing device used by the subscriber to access the remediation portal;</p>	<p>Tuvell discloses and/or renders obvious identifying the OS associated with the computing device used by the subscriber to access the remediation portal.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

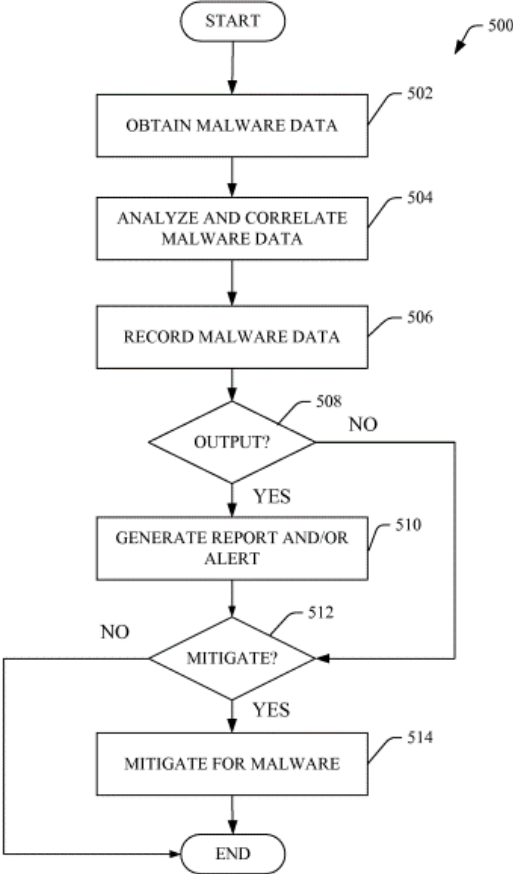
Claim	Exemplary Citation from Tuvell
	 <p>The diagram illustrates a Network Management System (100). It includes a User Interface (108) at the top, a Receiver Component (102) on the left, an Analysis Component (104) in the center, a Mitigation Component (110) on the right, and a Malware Data Store (106) at the bottom. The components are interconnected within the system boundary.</p> <p style="text-align: center;">FIG. 1</p>

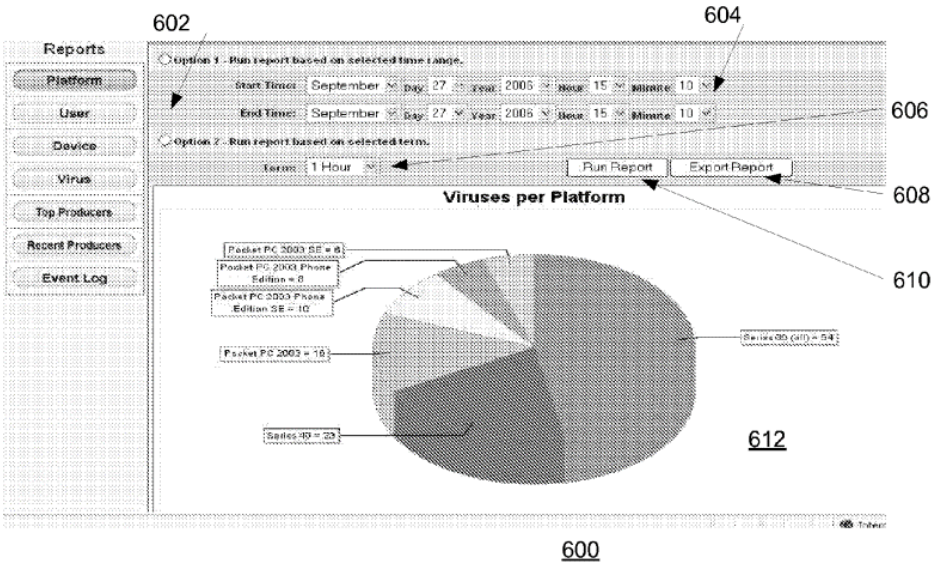
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) and an Analysis Component (104) are also shown. A Malware Data Store (106) is connected to the system. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

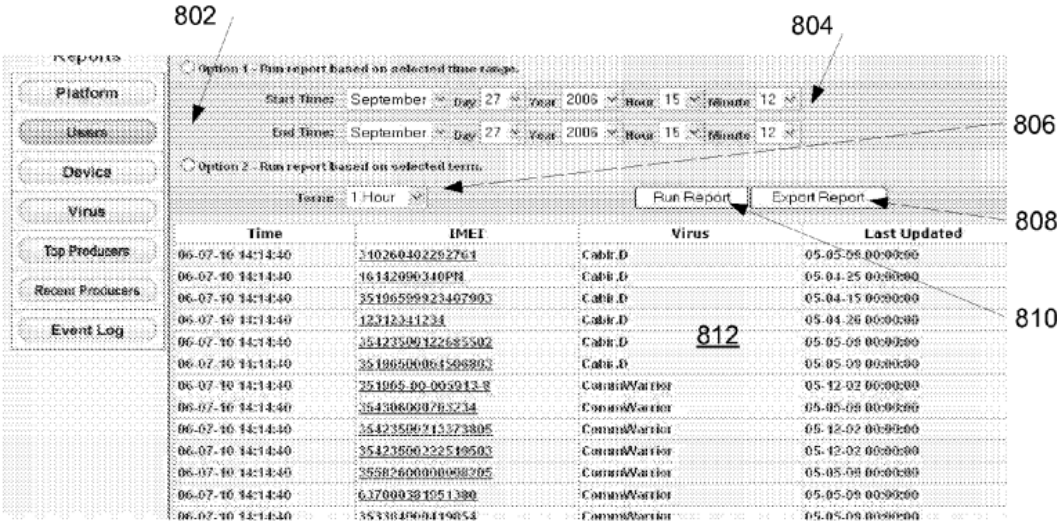
Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

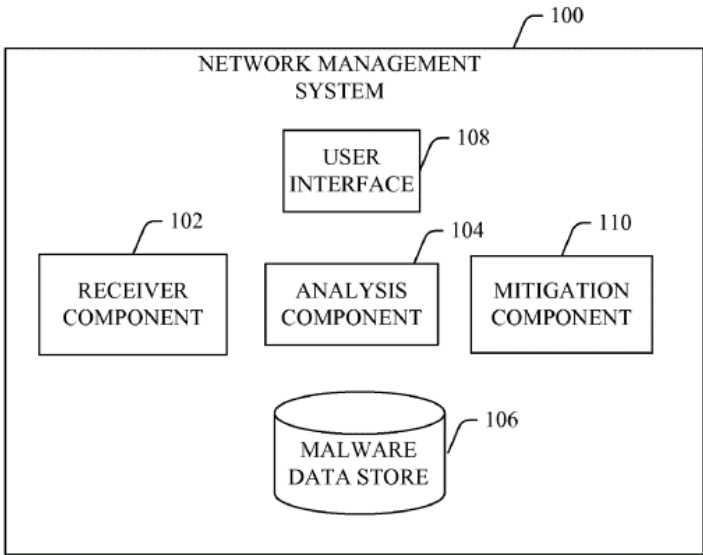
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

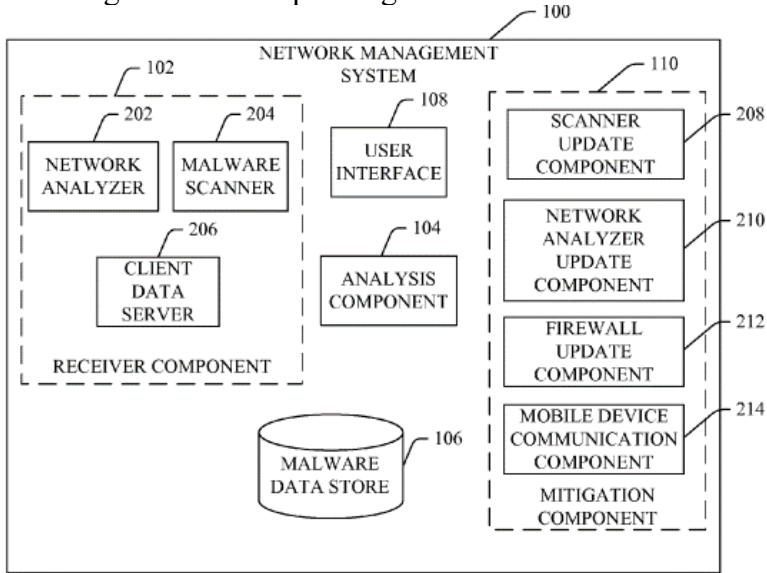
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

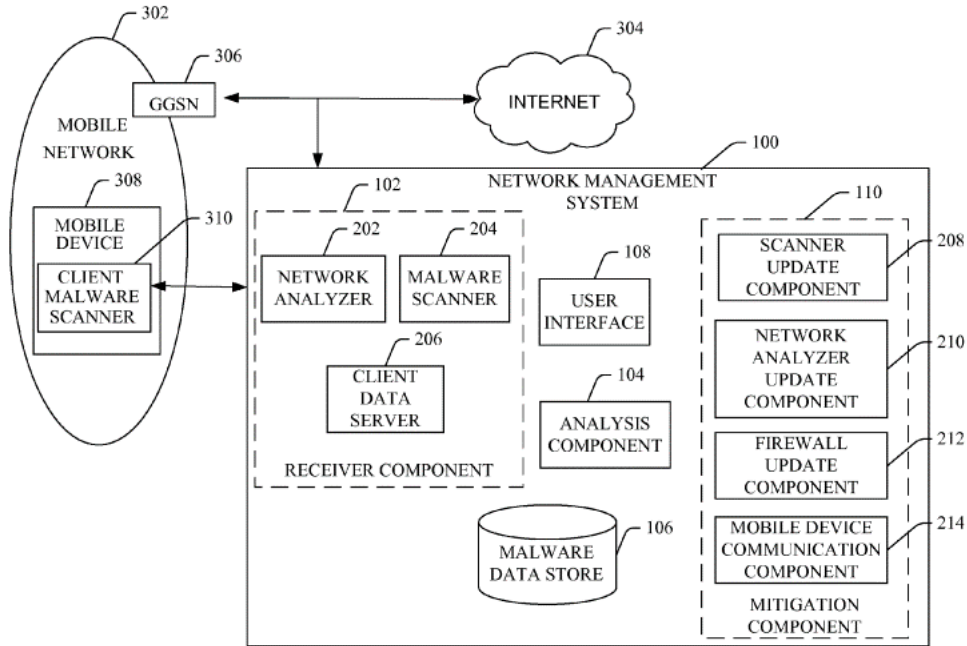
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

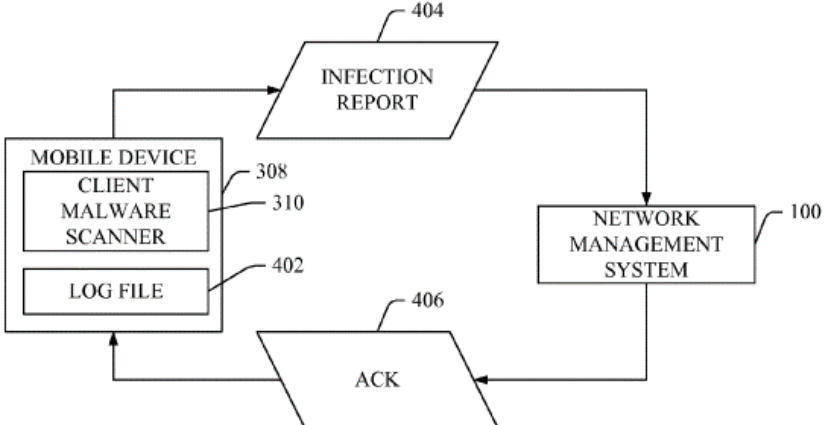
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[4C] comparing the OS associated with the computing device to the at least one of the first OS ID or second OS ID identified in the alert; and</p>	<p>Tuvell discloses and/or renders obvious comparing the OS associated with the computing device to the at least one of the first OS ID or second OS ID identified in the alert.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

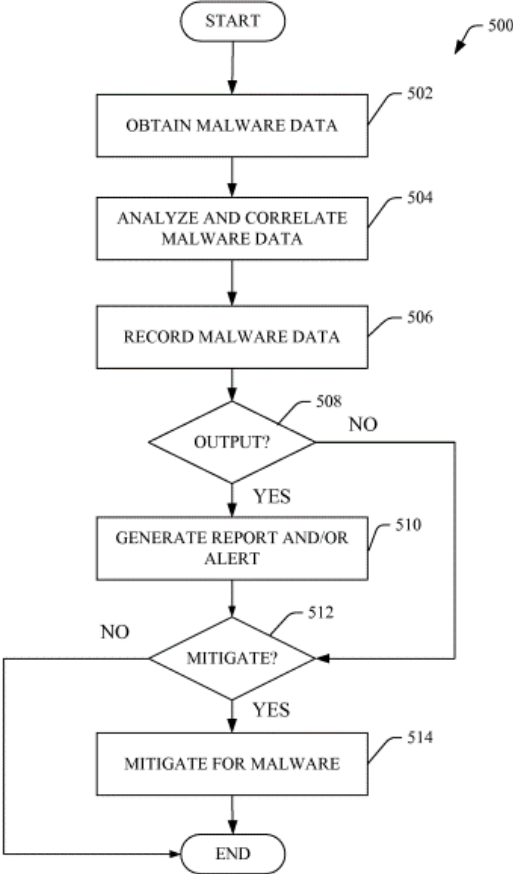
Claim	Exemplary Citation from Tuvell
	 <p>The diagram illustrates a Network Management System (100). It includes a User Interface (108) at the top, which is connected to three main processing components: a Receiver Component (102) on the left, an Analysis Component (104) in the center, and a Mitigation Component (110) on the right. Below these components is a Malware Data Store (106), represented by a cylinder. The entire system is enclosed in a rectangular box labeled 100.</p> <p style="text-align: center;">FIG. 1</p>

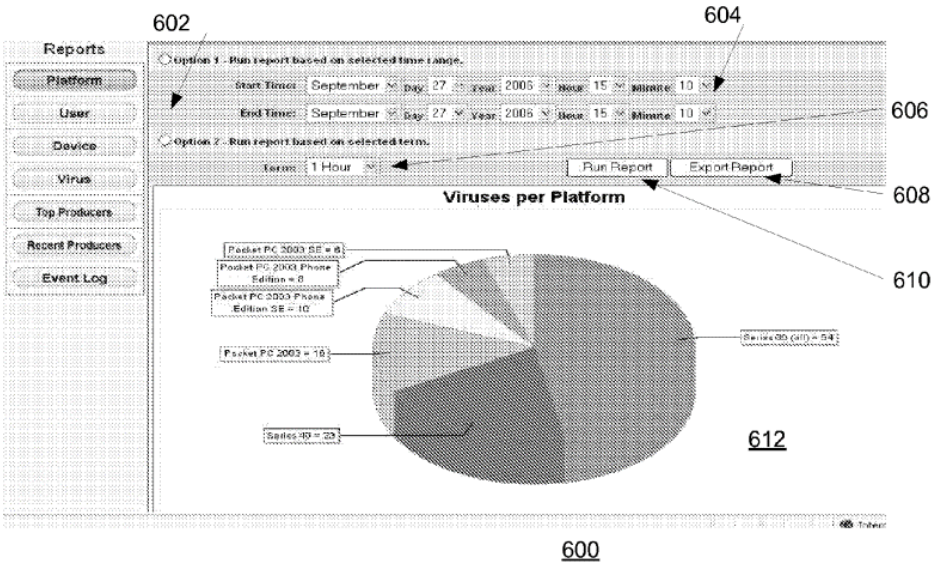
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) is connected to the system. An Analysis Component (104) is also present. A Malware Data Store (106) is shown as a database. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD MD[MOBILE DEVICE 308] --> IR[/INFECTION REPORT 404/] IR --> NMS[NETWORK MANAGEMENT SYSTEM 100] NMS --> ACK[/ACK 406/] ACK --> MD subgraph MD_Box [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

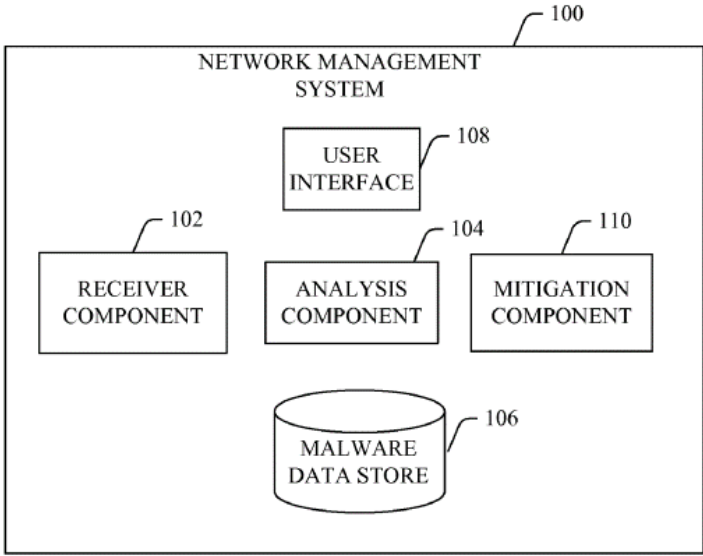
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

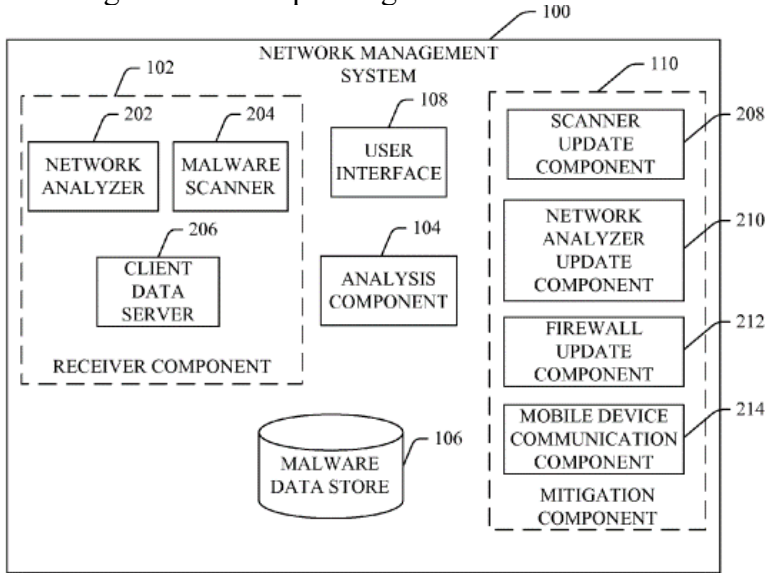
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p> <p>FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
[4D] providing malware remediation to the computing device when the at least one of the first OS ID or second OS ID	Tuvell discloses and/or renders obvious providing malware remediation to the computing device when the at least one of the first OS ID or second OS ID matches the OS associated with the computing device.

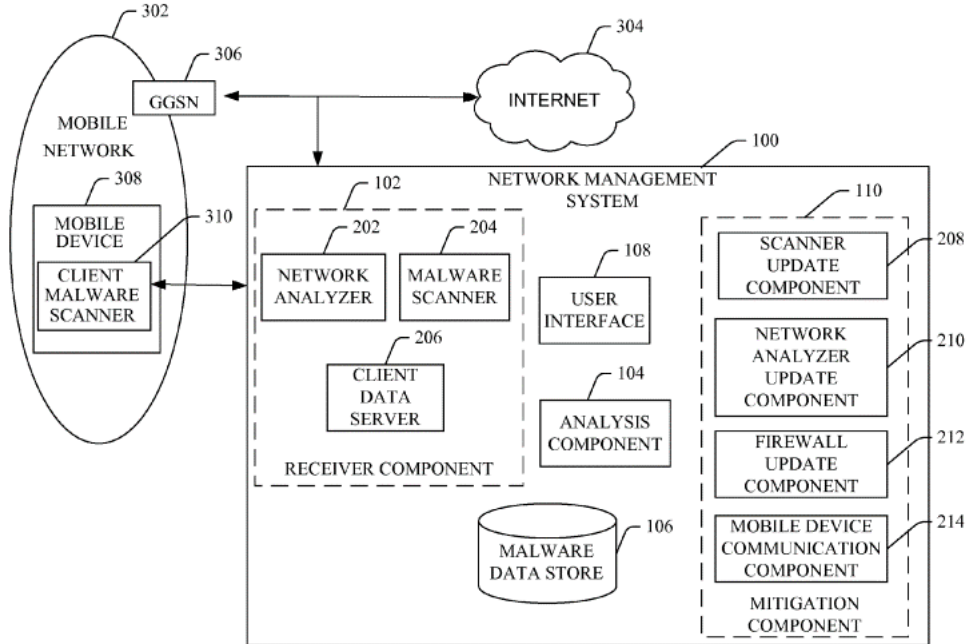
Claim	Exemplary Citation from Tuvell
<p>matches the OS associated with the computing device.</p>	<p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

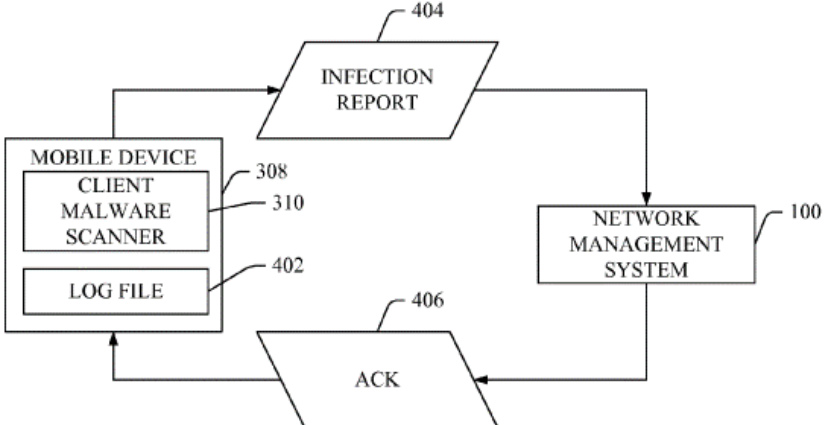
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104</p>

Claim	Exemplary Citation from Tuvell
	<p>generates a malware analysis or report that details and describes instances of malware in the mobile network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 31: "One way is to associate CoreStats 100 with the mobile network administrator's firewall so that the administrator can block identified malicious content. Another way is to report alarms upstream to operational support systems or OAM&P (Operations, Administration, Maintenance, and Provisioning) systems used by network service providers to manage mobile networks."</p> <p>Tuvell at ¶ 32: "In another embodiment, CoreStats 100 facilitates malware prevention by informing mobile device users and/or taking preventative steps at the mobile device. Once CoreStats 100 identifies an infected user, network administrators or operators send messages to a user to alert them to the problem, force an update of the user's mobile device's anti-virus software and definitions, or even disable the mobile device's data connections altogether."</p>

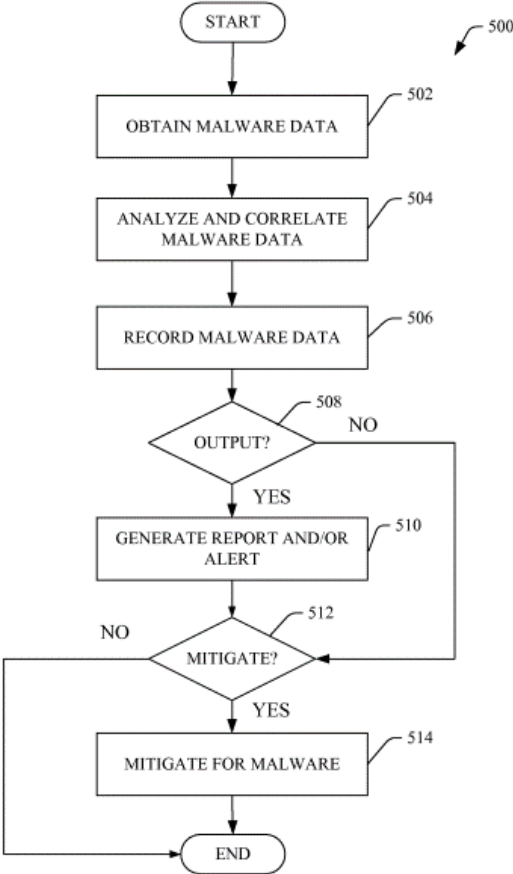
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) is connected to the system. An Analysis Component (104) is also present. A Malware Data Store (106) is shown as a database. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 35: "In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p>

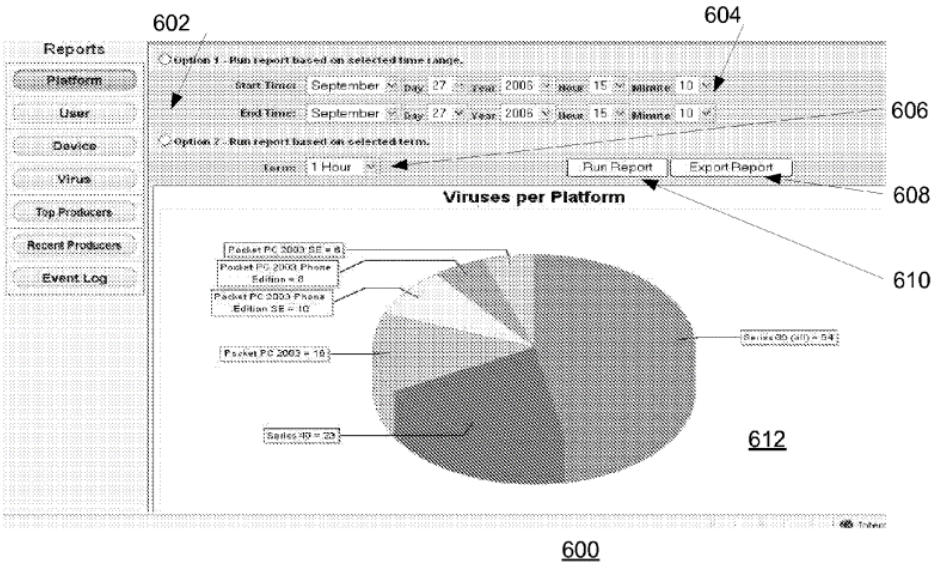
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

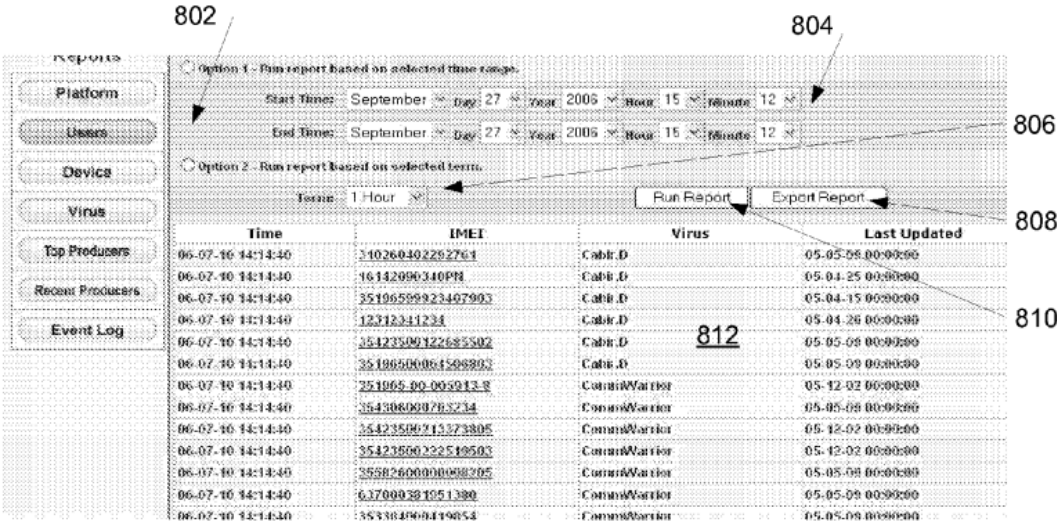
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph LR subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 45:</p> <p>"In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46:</p> <p>"Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 48: "In a further embodiment, the client malware scanner 310 generates an infection report 404 that contains information about the detected malware and transmits the infection report 404 to the client data server 206 of CoreStats 100. Report generation transmission is automatically triggered (pushed) upon detection of malware or based upon a periodic fixed time interval. Alternatively, infection reports 404 are maintained in the client device internal log file 402 until queried (pulled) by CoreStats 100. In yet another embodiment, infection reports 404 are delivered to CoreStats 100 using some combination of pulling and pushing. Infection reports 404 are transmitted, for example, using hypertext transfer protocol (http), file transfer protocol (ftp), or any packet data transmission method as would be generally known in the art."</p> <p>Tuvell at ¶ 49: "Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g.,</p>

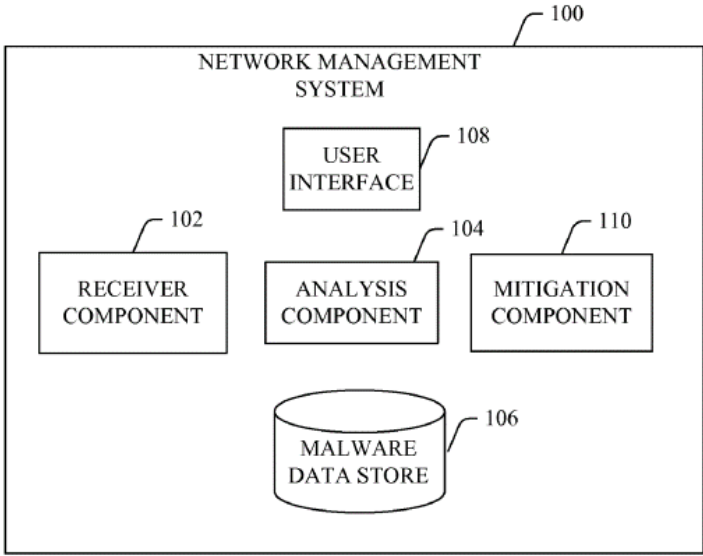
Claim	Exemplary Citation from Tuvell
	<p>phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 60: "At reference number 508 a determination is made as to whether to generate output, such as a report or alert. The determination can be based in whole or in part upon the malware data obtained from various sources. For example, if analysis indicates high levels of malware activity or significant impact on mobile network 302 performance, the determination is made to generate a report and alert or notify network administrators. Alternatively, reports are triggered periodically or upon operator request. In particular, operators can request particular reports via a user interface 108."</p> <p>Tuvell at ¶ 62: "In still other embodiments, a mitigation component 110 notifies a mobile device 308 user, force an update of mobile device 308 software, or even disable the mobile device's 308 data connections."</p> <p>Tuvell at ¶ 63: "Referring now to FIGS. 6-10, exemplary user interface displays are illustrated. As discussed above, CoreStats 100 also performs report generating functions. The analysis component 104 uses both stored and real-time information, including network traffic and individual user information, to generate statistics and dynamic graphs depicting malware activity and network statistics necessary to quantify relative levels of malware activity. For example, the analysis component 104 generates malware analyses, which can be presented by a user interface 108 as straightforward visual reports to alert managers and operators as to which platforms are infected with the most viruses, which viruses are spreading the fastest, the most recently infected mobile devices 308, and which infected mobile devices 308 are spreading the most viruses."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66:</p> <p>"Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[5] The method of claim 3 wherein if the OS of the computing device is not the at least one of the first OS ID or second OS ID identified in the alert an</p>	<p>Tuvell discloses and/or renders obvious that, if the OS of the computing device is not the at least one of the first OS ID or second OS ID identified in the alert an indication is provided to the subscriber to connect to the remediation portal with an other computing device connected to the access device.</p>

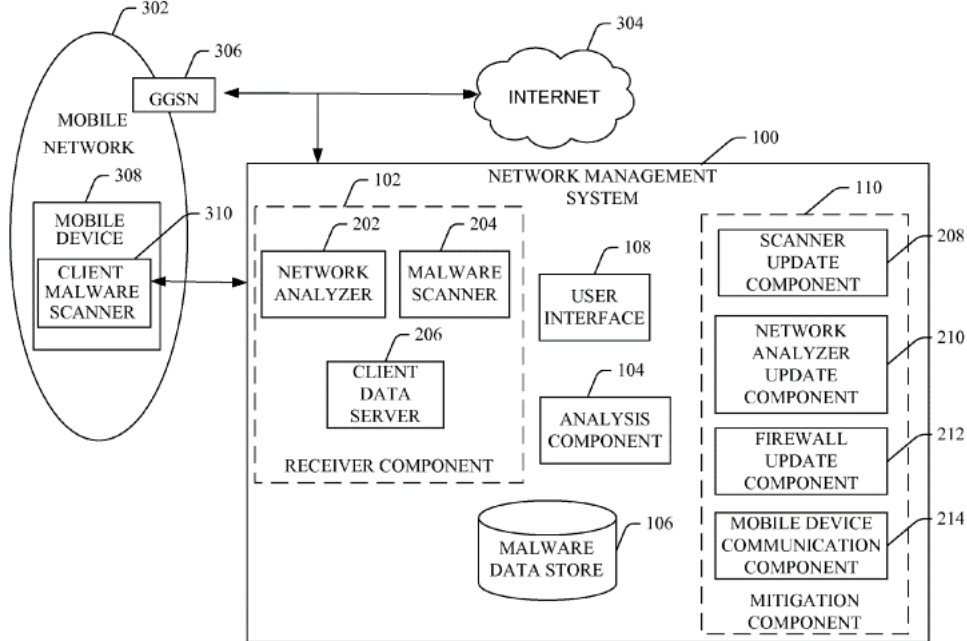
Claim	Exemplary Citation from Tuvell
<p>indication is provided to the subscriber to connect to the remediation portal with an other computing device connected to the access device.</p>	<p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

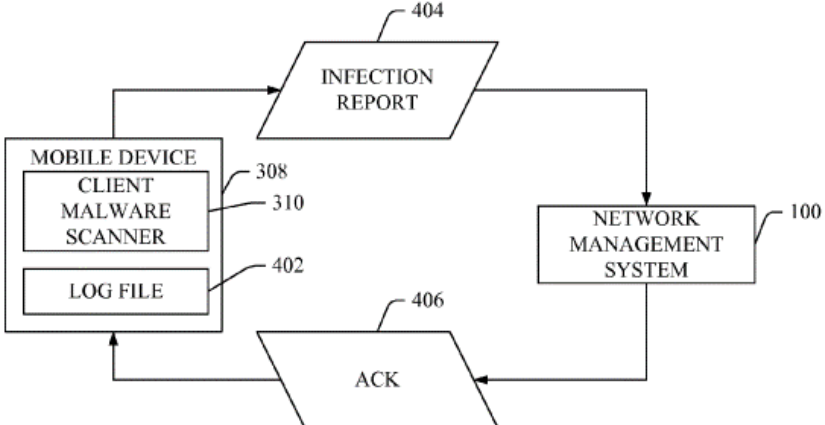
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104</p>

Claim	Exemplary Citation from Tuvell
	<p>generates a malware analysis or report that details and describes instances of malware in the mobile network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 31: "One way is to associate CoreStats 100 with the mobile network administrator's firewall so that the administrator can block identified malicious content. Another way is to report alarms upstream to operational support systems or OAM&P (Operations, Administration, Maintenance, and Provisioning) systems used by network service providers to manage mobile networks."</p> <p>Tuvell at ¶ 32: "In another embodiment, CoreStats 100 facilitates malware prevention by informing mobile device users and/or taking preventative steps at the mobile device. Once CoreStats 100 identifies an infected user, network administrators or operators send messages to a user to alert them to the problem, force an update of the user's mobile device's anti-virus software and definitions, or even disable the mobile device's data connections altogether."</p>

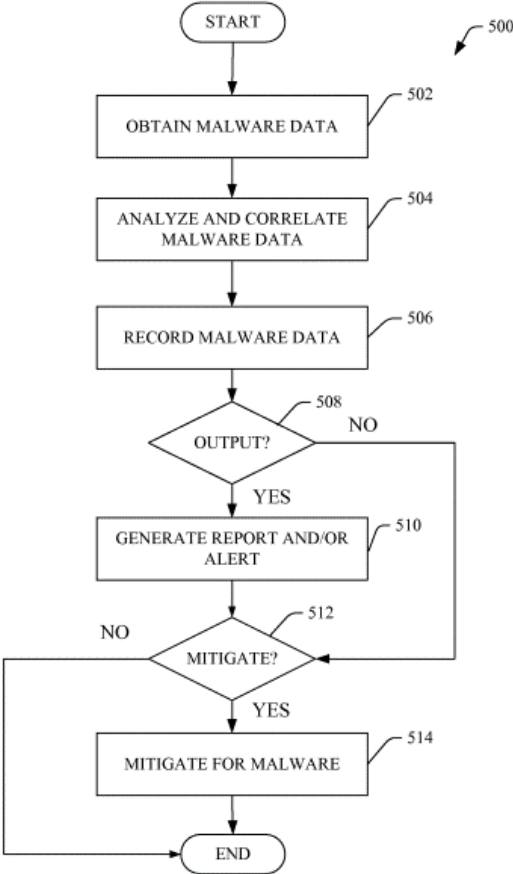
Claim	Exemplary Citation from Tuvell
	<p data-bbox="598 240 1102 267">Tuvell at Fig. 2 and corresponding text:</p> <div data-bbox="646 267 1407 836"> <pre> graph TD subgraph 100 [NETWORK MANAGEMENT SYSTEM] subgraph 102 [RECEIVER COMPONENT] 202[NETWORK ANALYZER] 204[MALWARE SCANNER] 206[CLIENT DATA SERVER] end 108[USER INTERFACE] 104[ANALYSIS COMPONENT] 106[(MALWARE DATA STORE)] subgraph 110 [MITIGATION COMPONENT] 208[SCANNER UPDATE COMPONENT] 210[NETWORK ANALYZER UPDATE COMPONENT] 212[FIREWALL UPDATE COMPONENT] 214[MOBILE DEVICE COMMUNICATION COMPONENT] end end </pre> </div> <p data-bbox="966 1071 1081 1112" style="text-align: center;">FIG. 2</p> <p data-bbox="598 1161 787 1193">Tuvell at ¶ 35:</p> <p data-bbox="598 1201 1890 1412">"In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p>

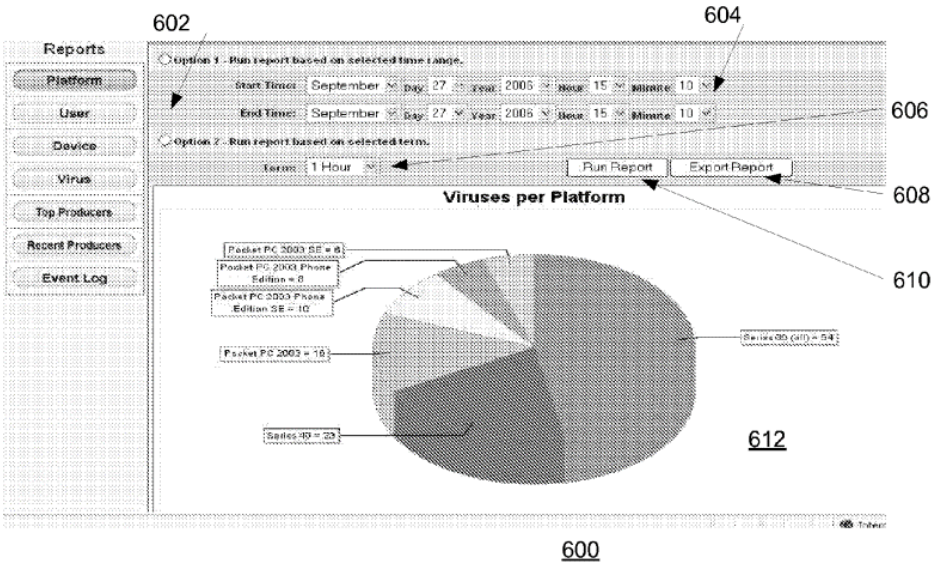
Claim	Exemplary Citation from Tuvell
	<p data-bbox="598 240 1102 272">Tuvell at Fig. 3 and corresponding text:</p>  <p data-bbox="1045 979 1155 1011">FIG. 3</p>

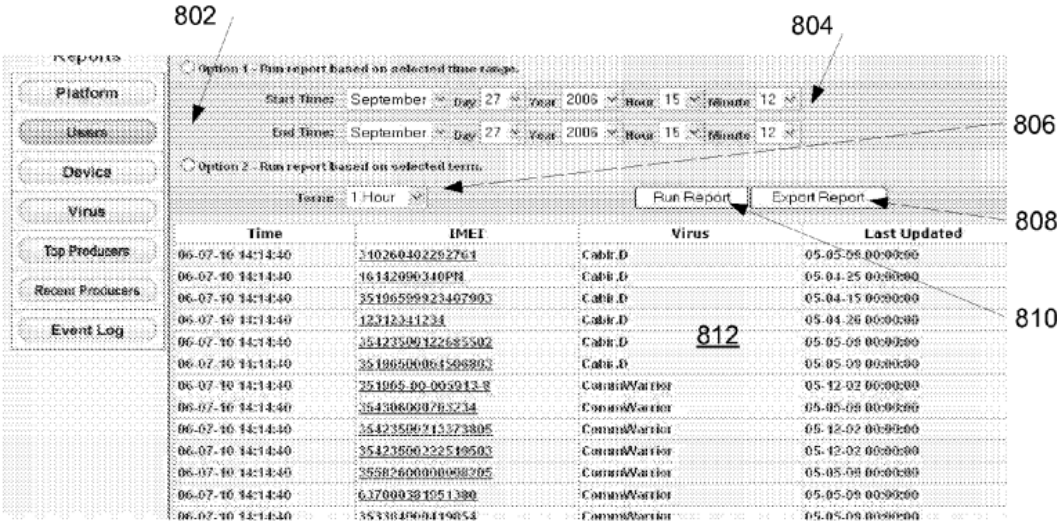
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph LR subgraph 308 [MOBILE DEVICE] 310[CLIENT MALWARE SCANNER] 402[LOG FILE] end 404[/INFECTION REPORT/] 100[NETWORK MANAGEMENT SYSTEM] 406[/ACK/] 310 --> 404 404 --> 100 100 --> 406 406 --> 308 </pre> <p>Tuvell at ¶ 45:</p> <p>"In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46:</p> <p>"Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 48: "In a further embodiment, the client malware scanner 310 generates an infection report 404 that contains information about the detected malware and transmits the infection report 404 to the client data server 206 of CoreStats 100. Report generation transmission is automatically triggered (pushed) upon detection of malware or based upon a periodic fixed time interval. Alternatively, infection reports 404 are maintained in the client device internal log file 402 until queried (pulled) by CoreStats 100. In yet another embodiment, infection reports 404 are delivered to CoreStats 100 using some combination of pulling and pushing. Infection reports 404 are transmitted, for example, using hypertext transfer protocol (http), file transfer protocol (ftp), or any packet data transmission method as would be generally known in the art."</p> <p>Tuvell at ¶ 49: "Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g.,</p>

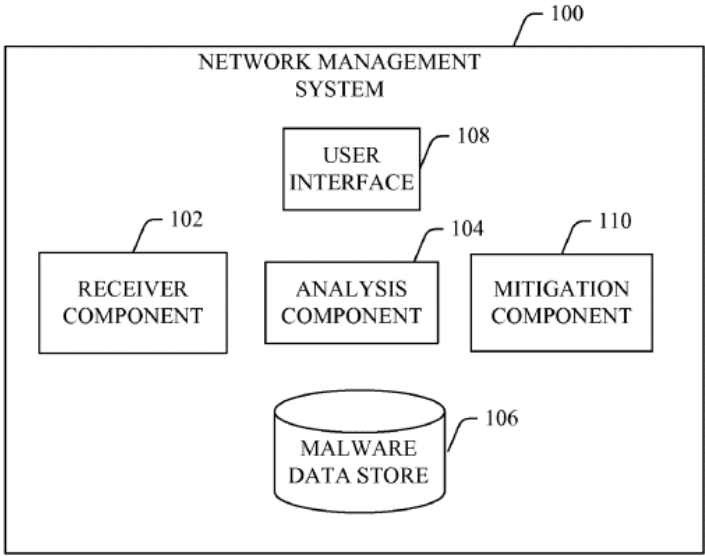
Claim	Exemplary Citation from Tuvell
	<p>phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 510 --> 512{MITIGATE?} 508 -- NO --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 514 --> END([END]) 512 -- NO --> END </pre> <p>FIG. 5</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 60: "At reference number 508 a determination is made as to whether to generate output, such as a report or alert. The determination can be based in whole or in part upon the malware data obtained from various sources. For example, if analysis indicates high levels of malware activity or significant impact on mobile network 302 performance, the determination is made to generate a report and alert or notify network administrators. Alternatively, reports are triggered periodically or upon operator request. In particular, operators can request particular reports via a user interface 108."</p> <p>Tuvell at ¶ 62: "In still other embodiments, a mitigation component 110 notifies a mobile device 308 user, force an update of mobile device 308 software, or even disable the mobile device's 308 data connections."</p> <p>Tuvell at ¶ 63: "Referring now to FIGS. 6-10, exemplary user interface displays are illustrated. As discussed above, CoreStats 100 also performs report generating functions. The analysis component 104 uses both stored and real-time information, including network traffic and individual user information, to generate statistics and dynamic graphs depicting malware activity and network statistics necessary to quantify relative levels of malware activity. For example, the analysis component 104 generates malware analyses, which can be presented by a user interface 108 as straightforward visual reports to alert managers and operators as to which platforms are infected with the most viruses, which viruses are spreading the fastest, the most recently infected mobile devices 308, and which infected mobile devices 308 are spreading the most viruses."</p>

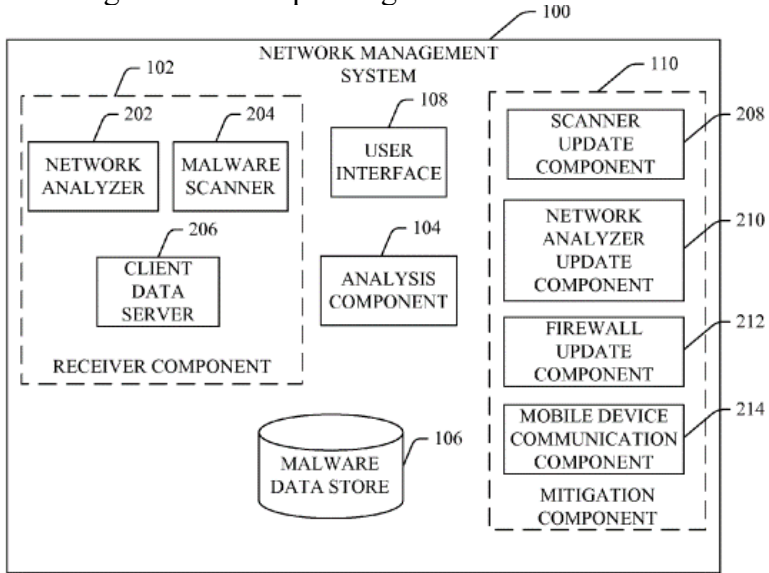
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[6] The method of claim 3 wherein the alert is cleared when all computing devices associated with the access device and the computing</p>	<p>Tuvell discloses and/or renders obvious that the alert is cleared when all computing devices associated with the access device and the computing device having the at least one of the first OS ID or second OS ID have accessed the remediation portal.</p> <p>Tuvell at Abstract:</p>

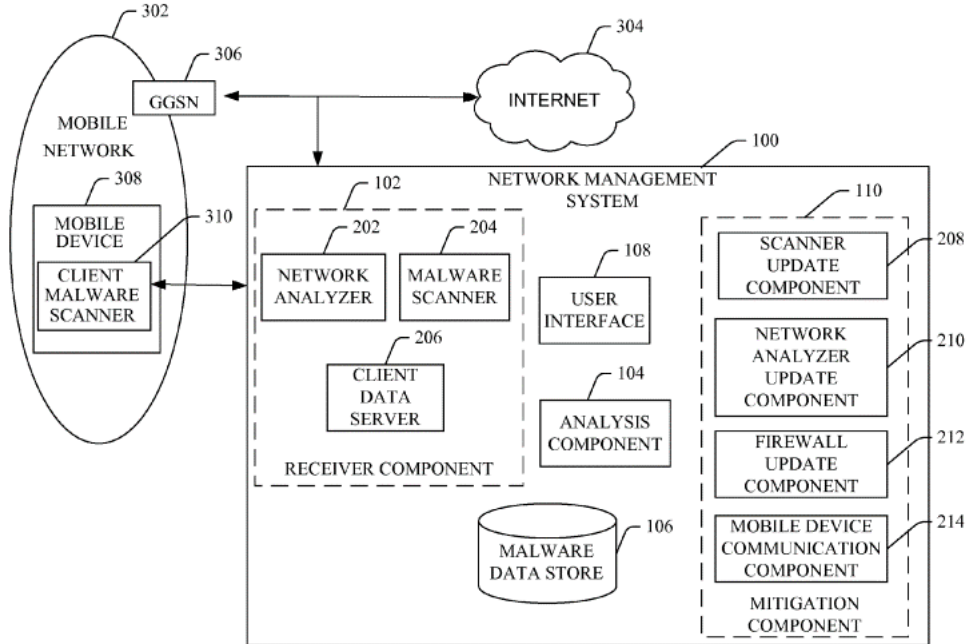
Claim	Exemplary Citation from Tuvell
<p>device having the at least one of the first OS ID or second OS ID have accessed the remediation portal.</p>	<p>"A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5:</p> <p>"Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9:</p> <p>"The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

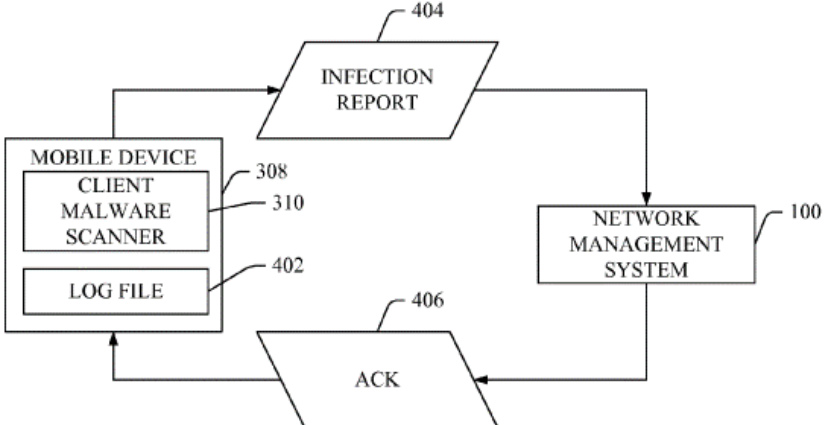
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104</p>

Claim	Exemplary Citation from Tuvell
	<p>generates a malware analysis or report that details and describes instances of malware in the mobile network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 31: "One way is to associate CoreStats 100 with the mobile network administrator's firewall so that the administrator can block identified malicious content. Another way is to report alarms upstream to operational support systems or OAM&P (Operations, Administration, Maintenance, and Provisioning) systems used by network service providers to manage mobile networks."</p> <p>Tuvell at ¶ 32: "In another embodiment, CoreStats 100 facilitates malware prevention by informing mobile device users and/or taking preventative steps at the mobile device. Once CoreStats 100 identifies an infected user, network administrators or operators send messages to a user to alert them to the problem, force an update of the user's mobile device's anti-virus software and definitions, or even disable the mobile device's data connections altogether."</p>

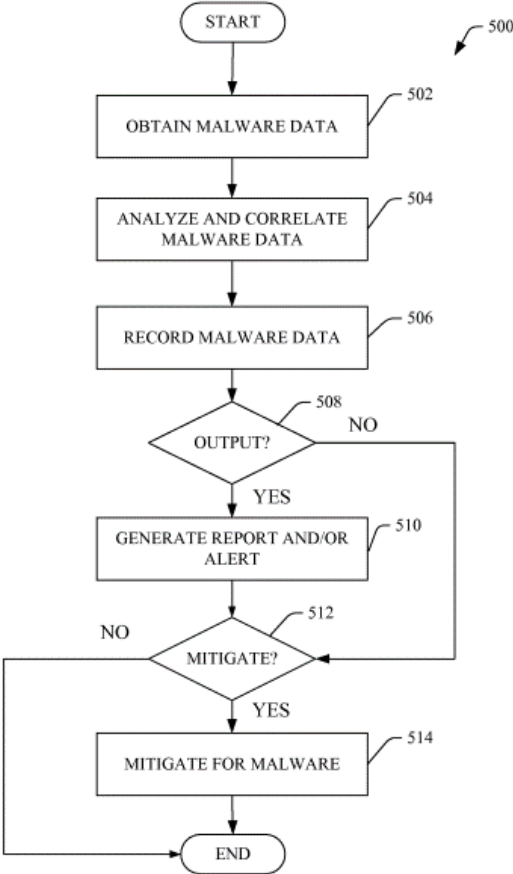
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100) with the following components and connections:</p> <ul style="list-style-type: none"> Receiver Component (102): A dashed box containing: <ul style="list-style-type: none"> Network Analyzer (202) and Malware Scanner (204) connected to a Client Data Server (206). User Interface (108) and Analysis Component (104) are connected to the Receiver Component (102). Malware Data Store (106) is connected to the Analysis Component (104). Update and Mitigation Components (110): A dashed box containing: <ul style="list-style-type: none"> Scanner Update Component (208), Network Analyzer Update Component (210), Firewall Update Component (212), Mobile Device Communication Component (214), and Mitigation Component. <p>FIG. 2</p> <p>Tuvell at ¶ 35:</p> <p>"In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p>

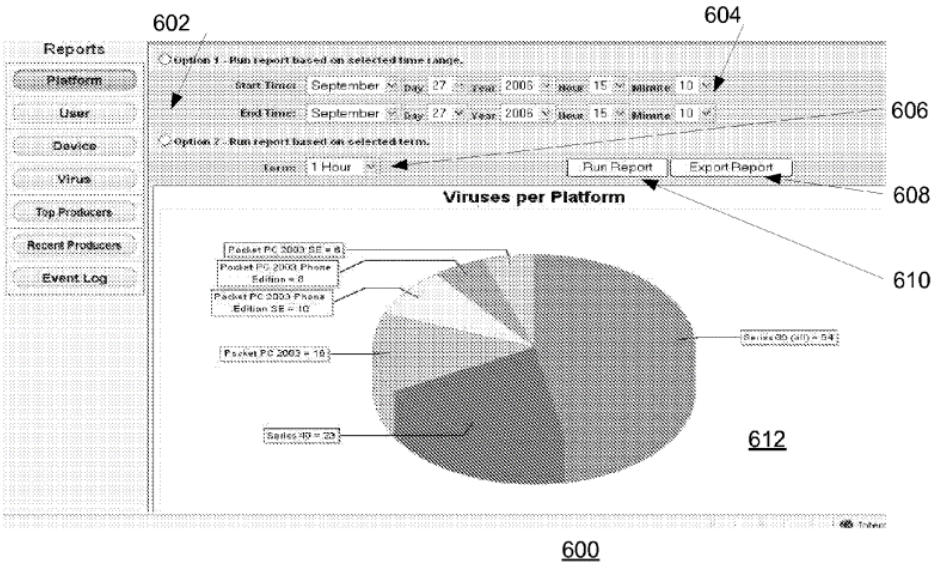
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

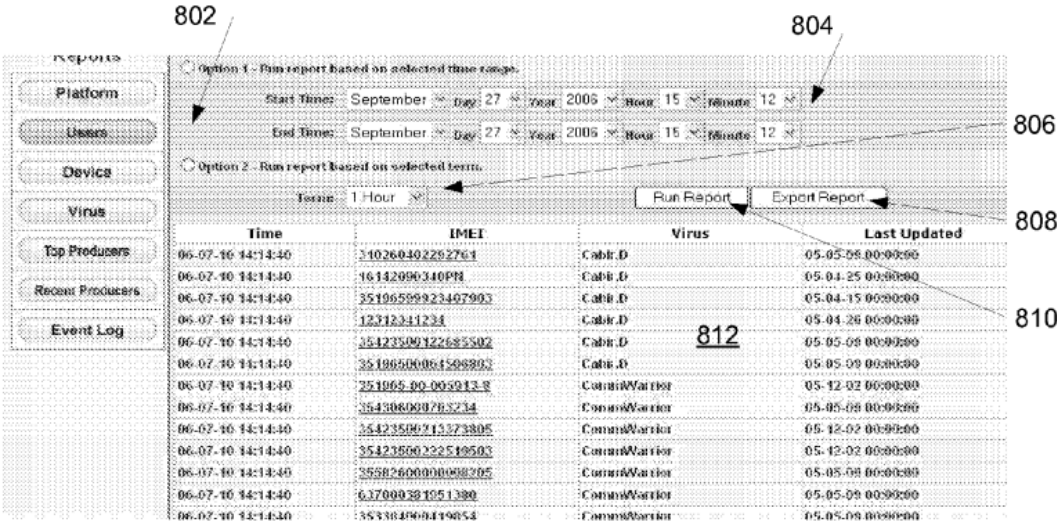
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph LR subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 45:</p> <p>"In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46:</p> <p>"Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 48: "In a further embodiment, the client malware scanner 310 generates an infection report 404 that contains information about the detected malware and transmits the infection report 404 to the client data server 206 of CoreStats 100. Report generation transmission is automatically triggered (pushed) upon detection of malware or based upon a periodic fixed time interval. Alternatively, infection reports 404 are maintained in the client device internal log file 402 until queried (pulled) by CoreStats 100. In yet another embodiment, infection reports 404 are delivered to CoreStats 100 using some combination of pulling and pushing. Infection reports 404 are transmitted, for example, using hypertext transfer protocol (http), file transfer protocol (ftp), or any packet data transmission method as would be generally known in the art."</p> <p>Tuvell at ¶ 49: "Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g.,</p>

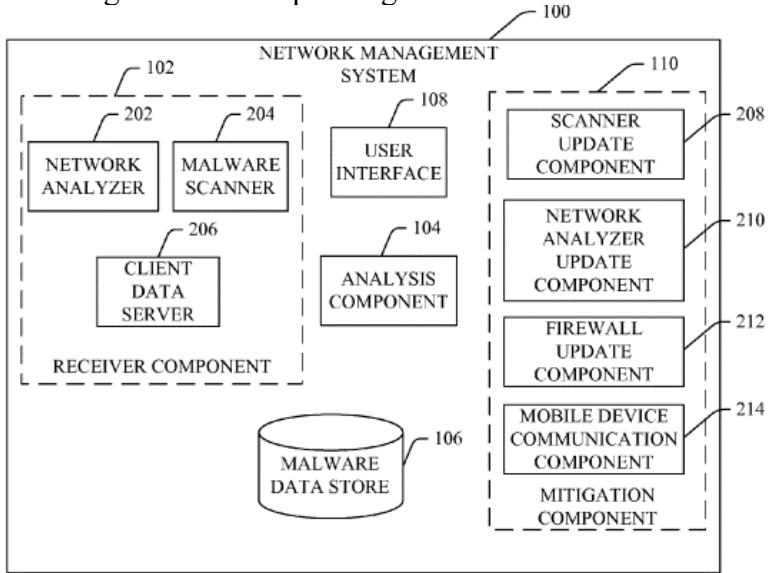
Claim	Exemplary Citation from Tuvell
	<p>phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

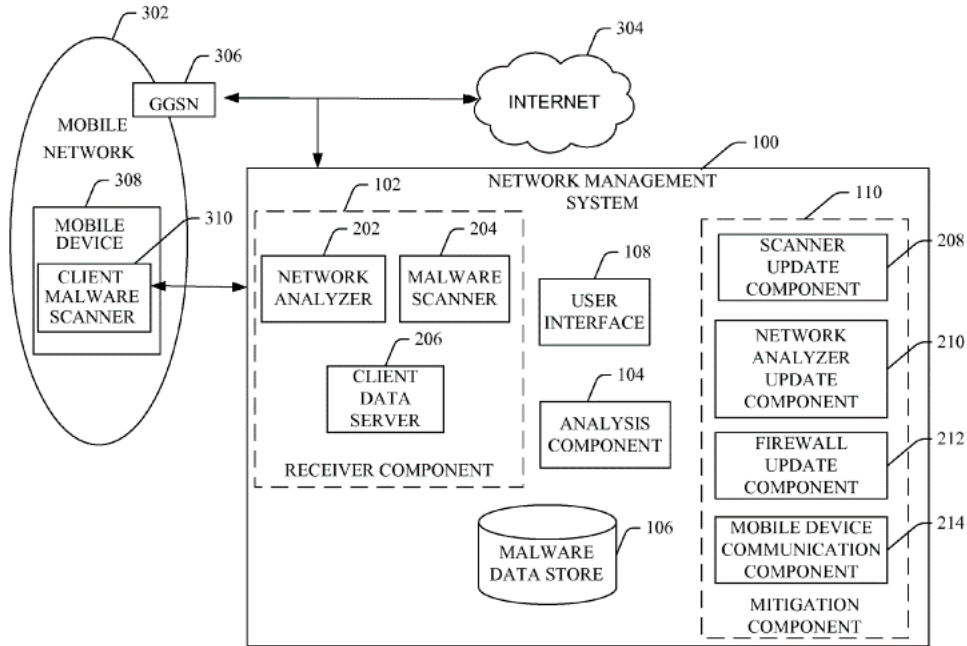
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 60: "At reference number 508 a determination is made as to whether to generate output, such as a report or alert. The determination can be based in whole or in part upon the malware data obtained from various sources. For example, if analysis indicates high levels of malware activity or significant impact on mobile network 302 performance, the determination is made to generate a report and alert or notify network administrators. Alternatively, reports are triggered periodically or upon operator request. In particular, operators can request particular reports via a user interface 108."</p> <p>Tuvell at ¶ 62: "In still other embodiments, a mitigation component 110 notifies a mobile device 308 user, force an update of mobile device 308 software, or even disable the mobile device's 308 data connections."</p> <p>Tuvell at ¶ 63: "Referring now to FIGS. 6-10, exemplary user interface displays are illustrated. As discussed above, CoreStats 100 also performs report generating functions. The analysis component 104 uses both stored and real-time information, including network traffic and individual user information, to generate statistics and dynamic graphs depicting malware activity and network statistics necessary to quantify relative levels of malware activity. For example, the analysis component 104 generates malware analyses, which can be presented by a user interface 108 as straightforward visual reports to alert managers and operators as to which platforms are infected with the most viruses, which viruses are spreading the fastest, the most recently infected mobile devices 308, and which infected mobile devices 308 are spreading the most viruses."</p>

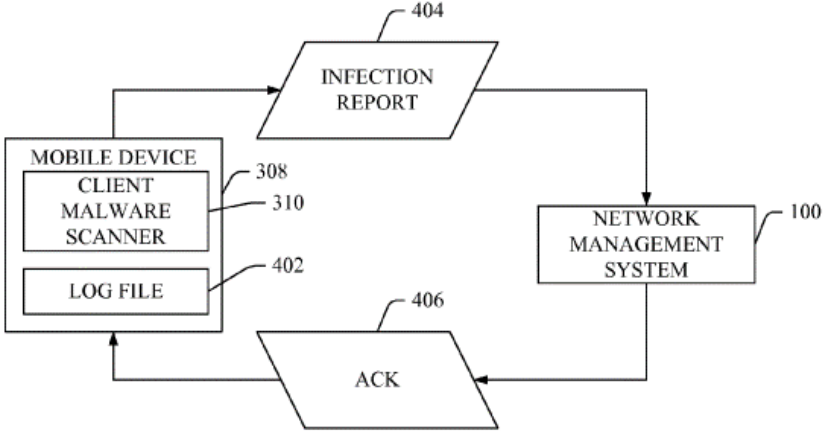
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66:</p> <p>"Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[7] The method of claim 2 wherein determining the first OS ID is performed by comparing one or more protocol parameters of the one or more TCP packets</p>	<p>Tuvell discloses and/or renders obvious that determining the first OS ID is performed by comparing one or more protocol parameters of the one or more TCP packets having a SYN flag associated with the session, wherein the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS ID associated with parameters.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

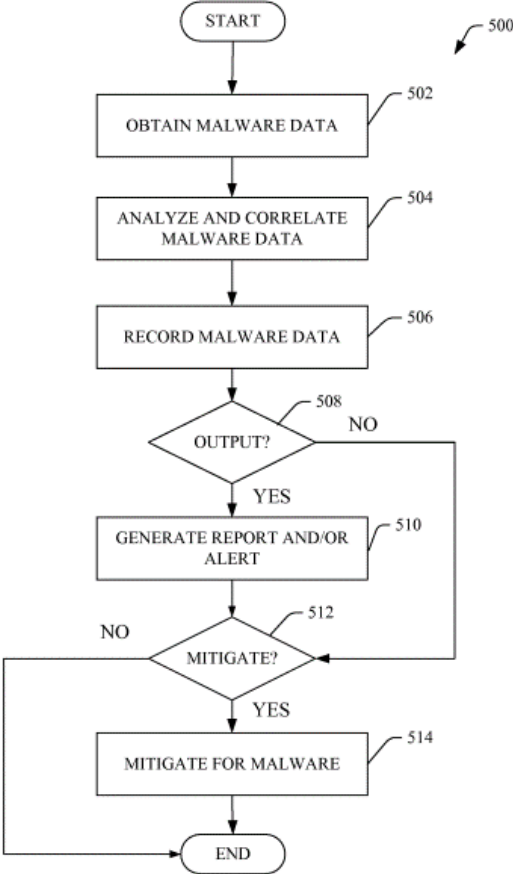
Claim	Exemplary Citation from Tuvell
<p>having a SYN flag associated with the session, wherein the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS ID associated with parameters.</p>	<div data-bbox="646 251 1344 803"> <pre> graph TD subgraph 100 [NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 108 --- 102 108 --- 104 108 --- 110 102 --- 104 104 --- 110 102 --- 106 104 --- 106 110 --- 106 end </pre> </div> <p style="text-align: center;">FIG. 1</p>

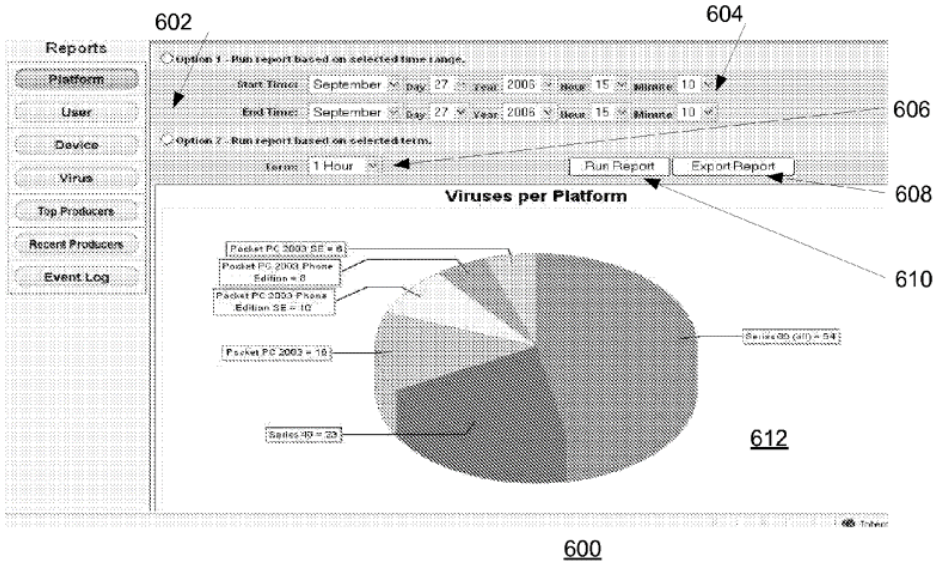
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) and an Analysis Component (104) are also shown. A Malware Data Store (106) is connected to the system. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD MD[MOBILE DEVICE 308] --> IR[/INFECTION REPORT 404/] IR --> NMS[NETWORK MANAGEMENT SYSTEM 100] NMS --> ACK[/ACK 406/] ACK --> MD subgraph MD_Box [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

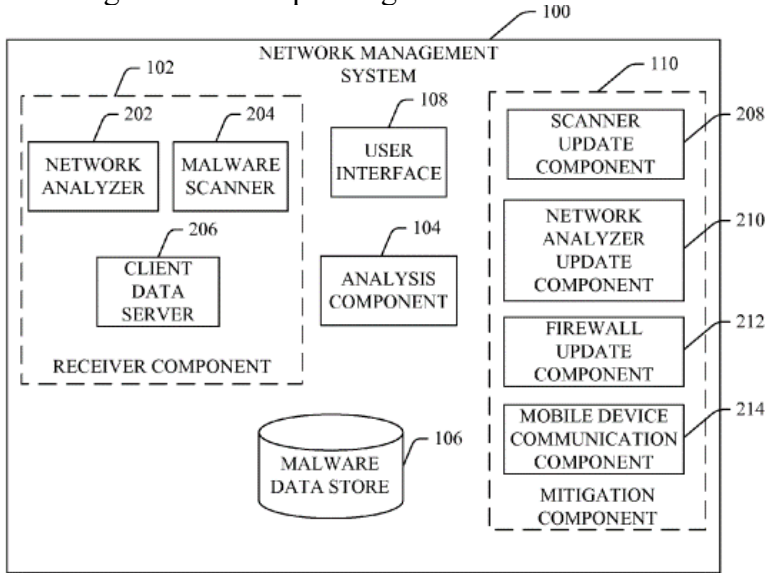
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

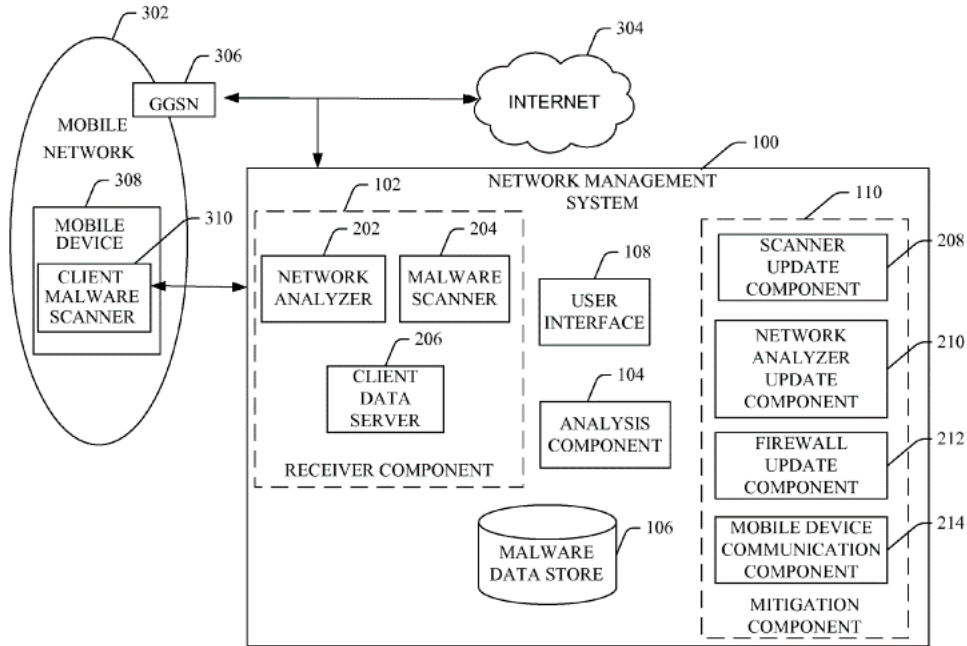
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 510 --> 512{MITIGATE?} 508 -- NO --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 514 --> END([END]) 512 -- NO --> END </pre> <p>FIG. 5</p>

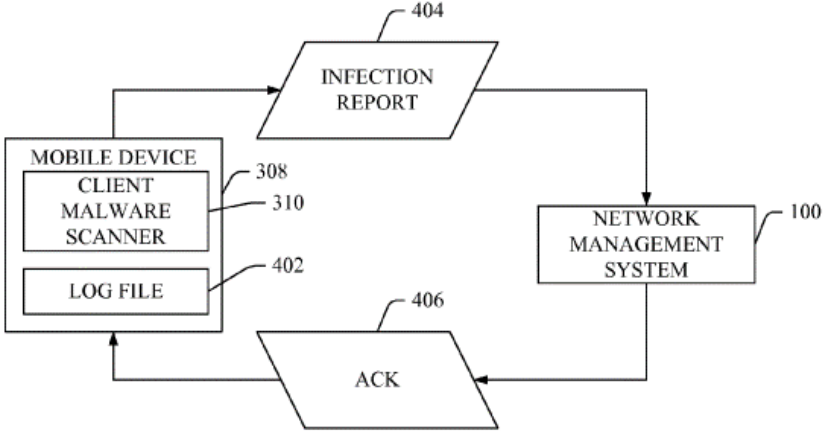
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p> <p>FIG. 8</p> <p>Tuvell at ¶ 66:</p> <p>"Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
[8] The method of claim 2 wherein determining the application parameter associated with the session and identifying the second OS ID from the application	<p>Tuvell discloses and/or renders obvious that determining the application parameter associated with the session and identifying the second OS ID from the application parameter by inspecting an HTTP user agent string to determine the second OS ID.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

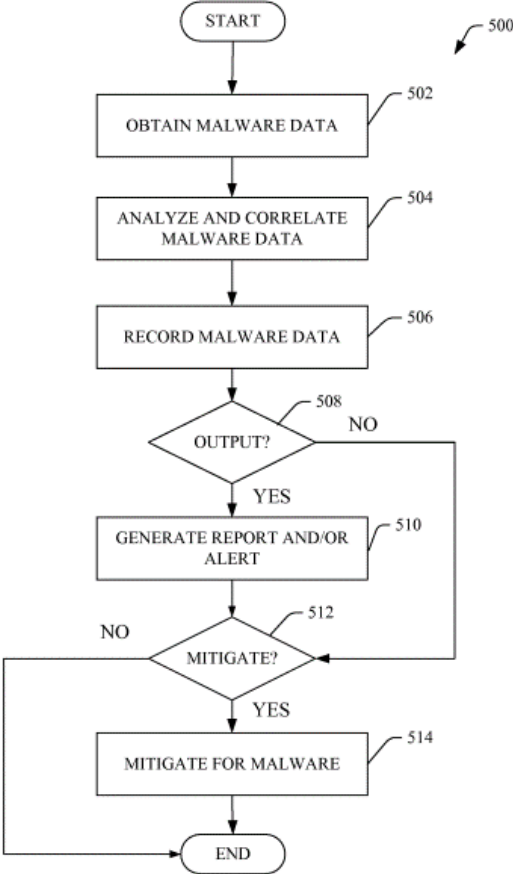
Claim	Exemplary Citation from Tuvell
<p>parameter by inspecting an HTTP user agent string to determine the second OS ID.</p>	<div data-bbox="646 251 1344 803"> <pre> graph TD 100[NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 108 --- 100 102 --- 100 104 --- 100 110 --- 100 106 --- 100 </pre> <p>The diagram illustrates a Network Management System (100). It consists of several interconnected components: a User Interface (108) at the top, a Receiver Component (102) on the left, an Analysis Component (104) in the center, a Mitigation Component (110) on the right, and a Malware Data Store (106) at the bottom. All components are part of the overall system 100.</p> </div> <p style="text-align: center;">FIG. 1</p>

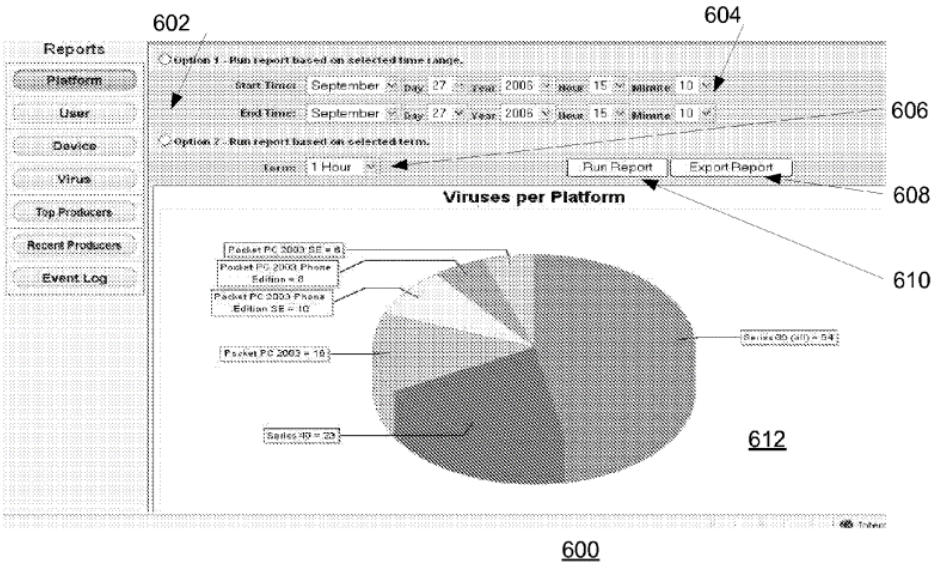
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) and an Analysis Component (104) are also shown. A Malware Data Store (106) is connected to the system. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD MD[MOBILE DEVICE 308] --> IR[/INFECTION REPORT 404/] IR --> NMS[NETWORK MANAGEMENT SYSTEM 100] NMS --> ACK[/ACK 406/] ACK --> MD subgraph MD_Box [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

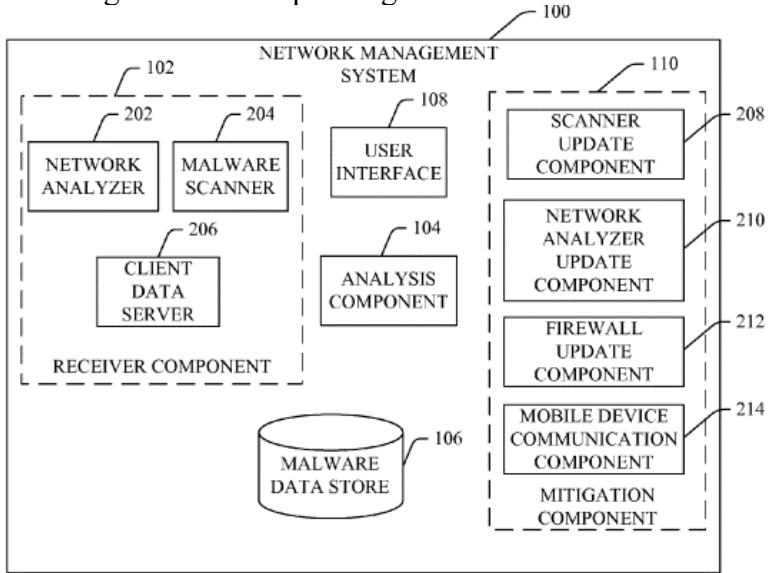
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

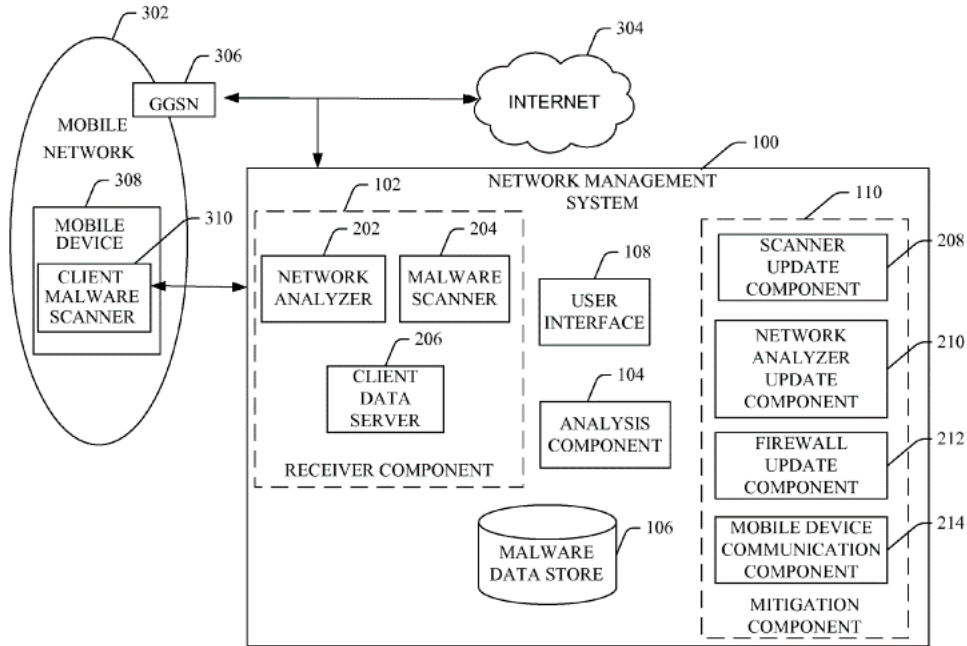
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

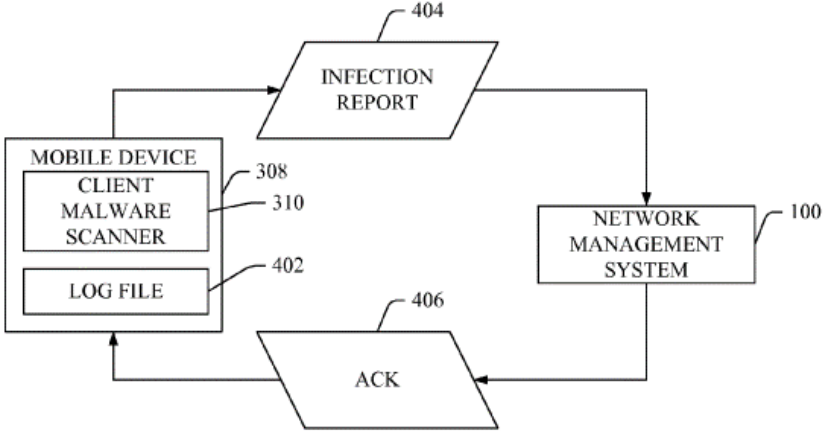
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p> <p>FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
[9] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the	<p>Tuvell discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the session is identified by the OS ID identified as a more reliable OS ID.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

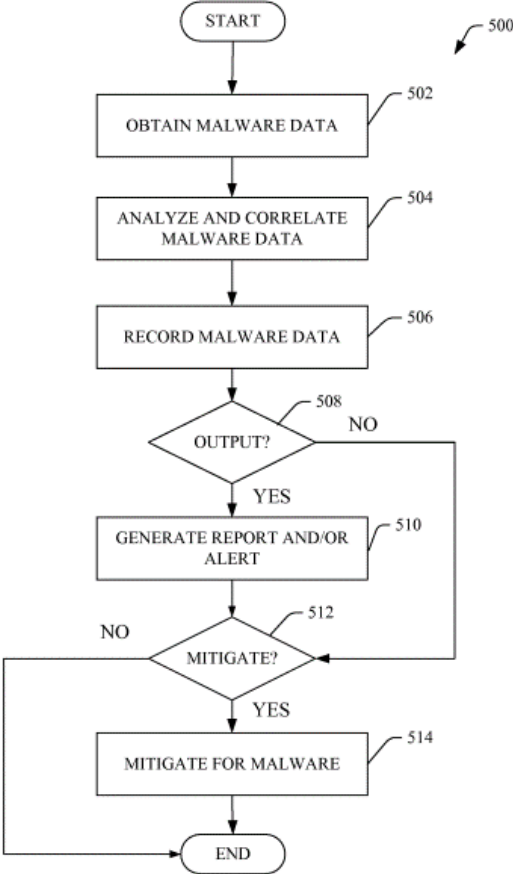
Claim	Exemplary Citation from Tuvell
<p>session is identified by the OS ID identified as a more reliable OS ID.</p>	<div data-bbox="646 251 1344 803"> <pre> graph TD 100[NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 108 --- 100 102 --- 100 104 --- 100 110 --- 100 106 --- 100 </pre> <p>The diagram illustrates a Network Management System (100). It includes a User Interface (108) at the top, a Receiver Component (102) on the left, an Analysis Component (104) in the center, a Mitigation Component (110) on the right, and a Malware Data Store (106) at the bottom. All components are interconnected within the system boundary.</p> </div> <p style="text-align: center;">FIG. 1</p>

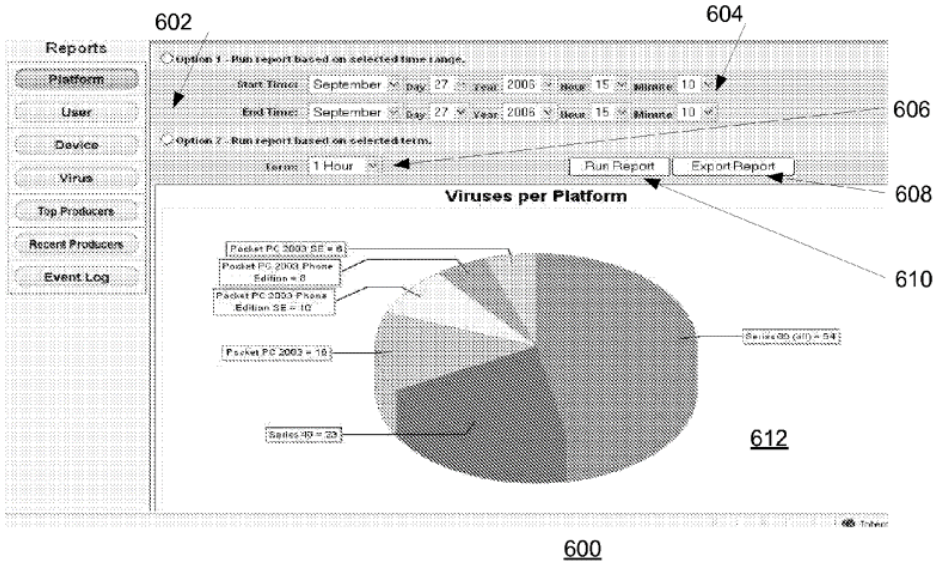
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) and an Analysis Component (104) are also shown. A Malware Data Store (106) is connected to the system. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

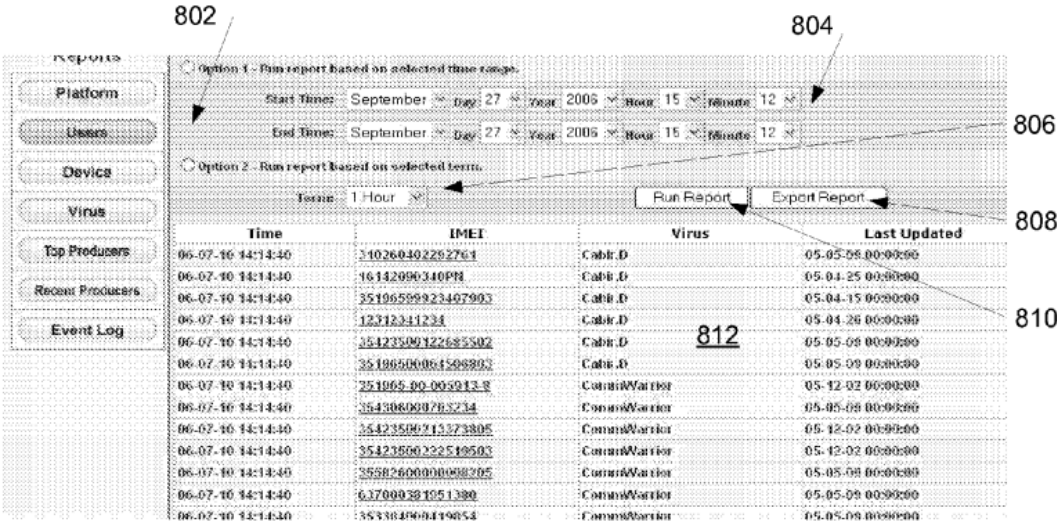
Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

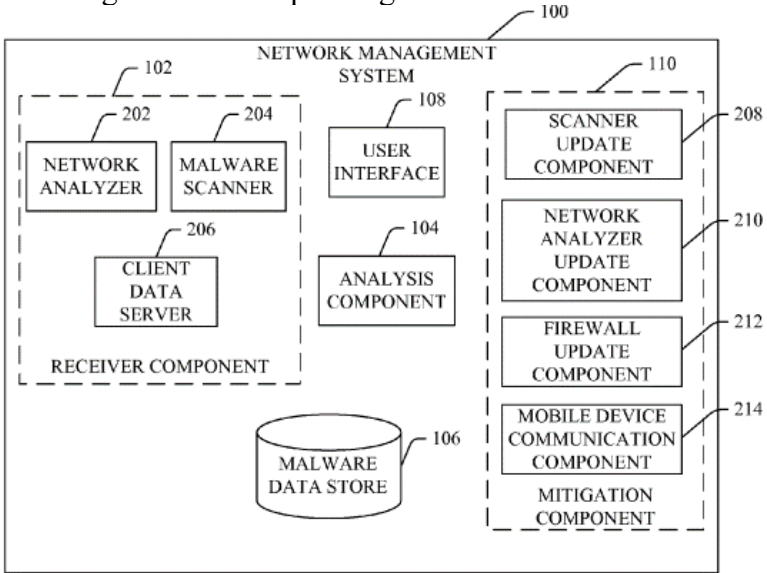
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

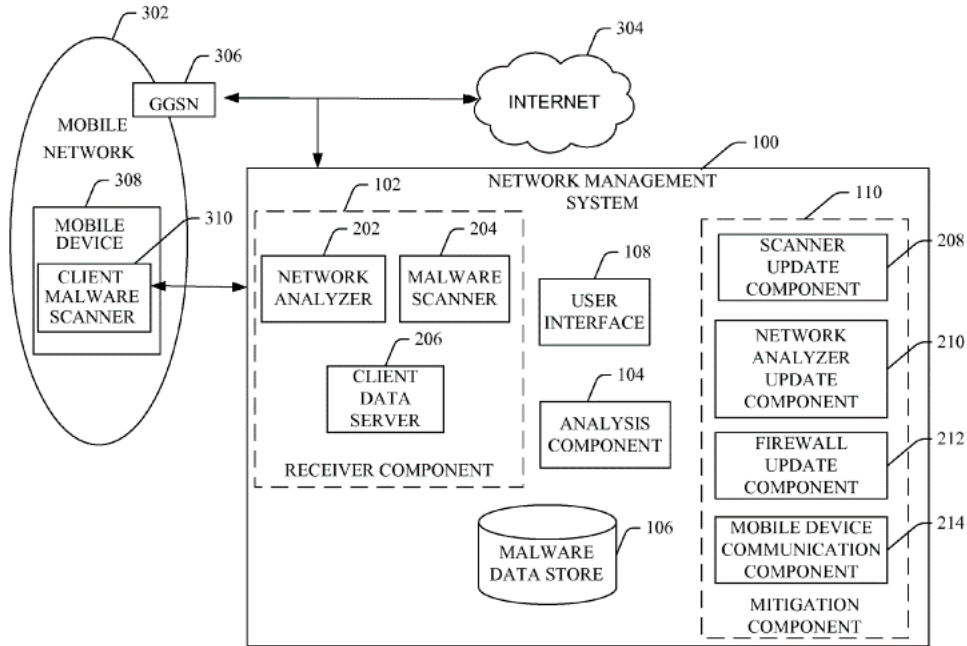
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

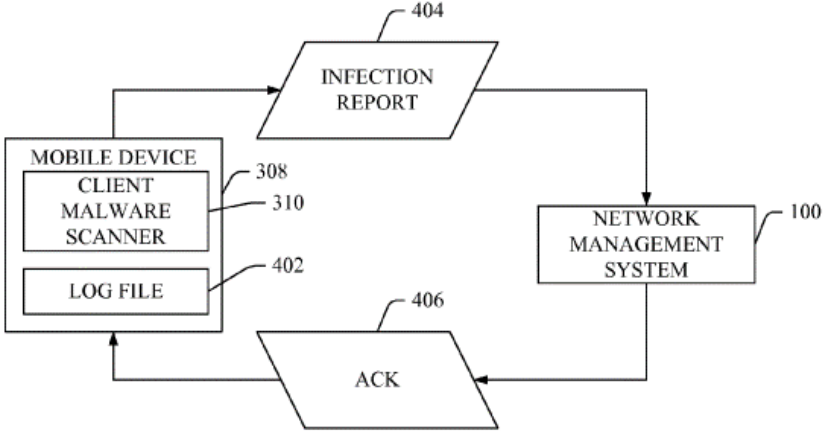
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[10] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID matches</p>	<p>Tuvell discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID either OS ID is used.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>

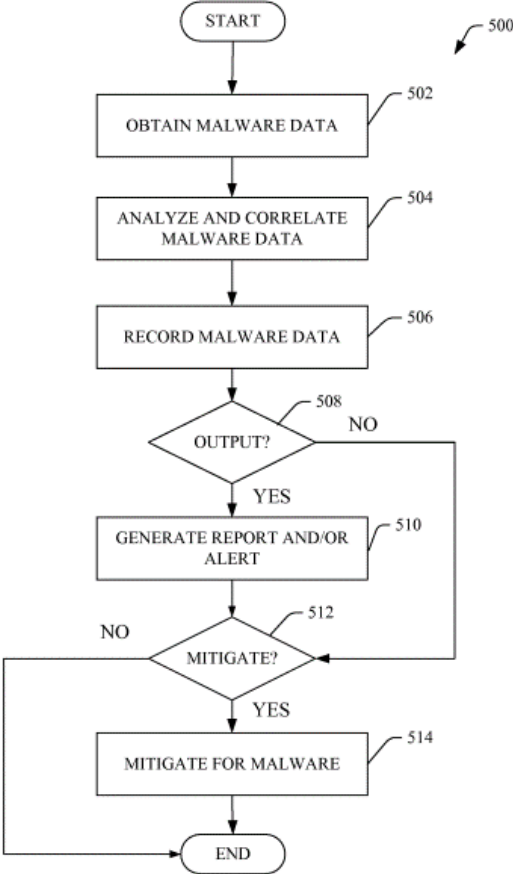
Claim	Exemplary Citation from Tuvell
<p>the first OS ID either OS ID is used.</p>	<div data-bbox="646 251 1346 803"> <pre> graph TD 100[NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 108 --- 100 102 --- 100 104 --- 100 110 --- 100 106 --- 100 </pre> <p>The diagram illustrates a Network Management System (100) enclosed in a rectangular box. At the top center of the box is the label "NETWORK MANAGEMENT SYSTEM". Below this label, there are five components: a "USER INTERFACE" (108) at the top center, a "RECEIVER COMPONENT" (102) on the left, an "ANALYSIS COMPONENT" (104) in the center, a "MITIGATION COMPONENT" (110) on the right, and a "MALWARE DATA STORE" (106) represented by a cylinder at the bottom center. Each component is connected to the system label by a thin line.</p> </div> <p style="text-align: center;">FIG. 1</p>

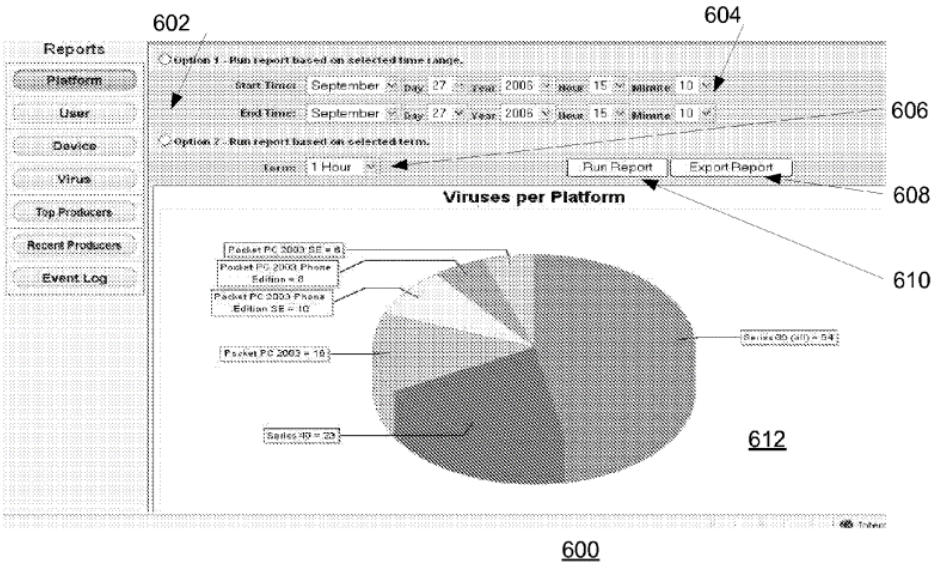
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) and an Analysis Component (104) are also shown. A Malware Data Store (106) is connected to the system. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

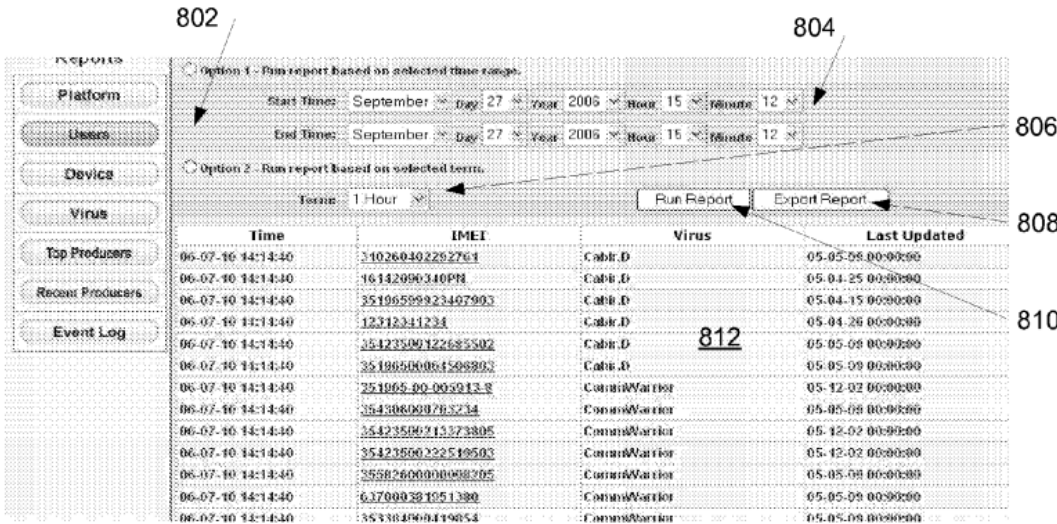
Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end MD --> IR[/INFECTION REPORT 404/] IR --> NMS[NETWORK MANAGEMENT SYSTEM 100] NMS --> ACK[/ACK 406/] ACK --> MD </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

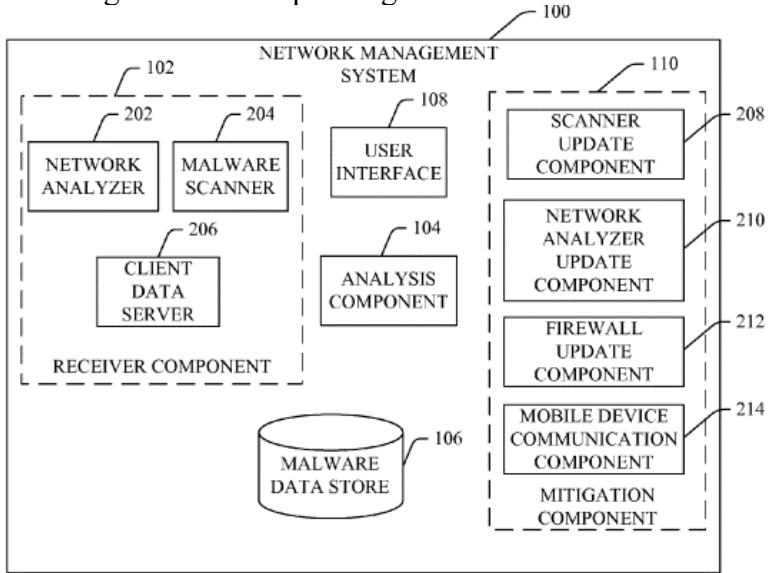
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

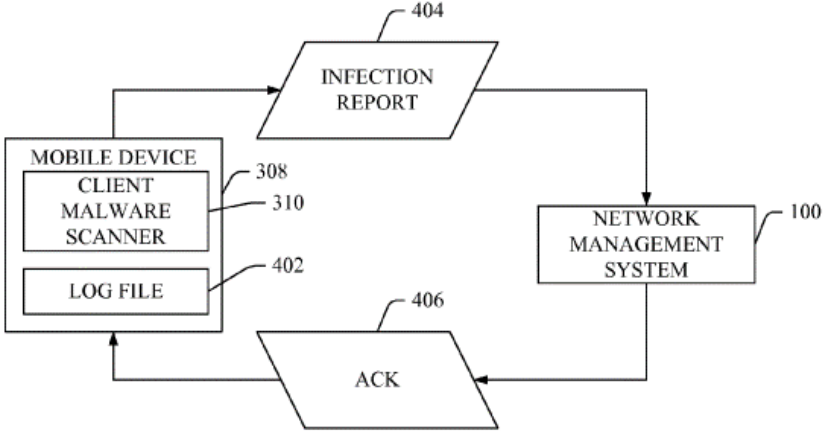
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell																																																								
	<p>Tuvell at Fig. 8 and corresponding text:</p> <div><p>The screenshot shows a web-based interface for generating a user infection report. On the left is a sidebar with filters: Platform, Users, Device, Virus, Top Producers, Recent Producers, and Event Log. The main area has two radio button options: 'Option 1 - Run report based on selected time range' and 'Option 2 - Run report based on selected term'. Under Option 1, there are date pickers for Start Time and End Time, both set to September 27, 2005, at 15:12. Under Option 2, there is a 'Term' dropdown set to '1 Hour'. To the right are 'Run Report' and 'Export Report' buttons. Below these is a table with the following data:</p><table><thead><tr><th>Time</th><th>IMEI</th><th>Virus</th><th>Last Updated</th></tr></thead><tbody><tr><td>06-07-16 14:14:40</td><td>340260402262764</td><td>Cable.D</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>16112090340PM</td><td>Cable.D</td><td>05-04-25 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35196559923407993</td><td>Cable.D</td><td>05-04-15 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>12312341234</td><td>Cable.D</td><td>05-04-26 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35423500122685502</td><td>Cable.D</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35186500064596803</td><td>Cable.D</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>251065 00-00-0013 8</td><td>CommWarrior</td><td>05-12-02 00:00:00</td></tr><tr><td>06-07-16 14:14:40</td><td>354300009765234</td><td>CommWarrior</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35423500215173805</td><td>CommWarrior</td><td>05-12-02 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>3542350022519503</td><td>CommWarrior</td><td>05-12-02 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35582600000008205</td><td>CommWarrior</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>637000381951300</td><td>CommWarrior</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>353384960419054</td><td>CommWarrior</td><td>05-05-09 00:00:00</td></tr></tbody></table></div> <p>FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>	Time	IMEI	Virus	Last Updated	06-07-16 14:14:40	340260402262764	Cable.D	05-05-09 00:00:00	06-07-10 14:14:40	16112090340PM	Cable.D	05-04-25 00:00:00	06-07-10 14:14:40	35196559923407993	Cable.D	05-04-15 00:00:00	06-07-10 14:14:40	12312341234	Cable.D	05-04-26 00:00:00	06-07-10 14:14:40	35423500122685502	Cable.D	05-05-09 00:00:00	06-07-10 14:14:40	35186500064596803	Cable.D	05-05-09 00:00:00	06-07-10 14:14:40	251065 00-00-0013 8	CommWarrior	05-12-02 00:00:00	06-07-16 14:14:40	354300009765234	CommWarrior	05-05-09 00:00:00	06-07-10 14:14:40	35423500215173805	CommWarrior	05-12-02 00:00:00	06-07-10 14:14:40	3542350022519503	CommWarrior	05-12-02 00:00:00	06-07-10 14:14:40	35582600000008205	CommWarrior	05-05-09 00:00:00	06-07-10 14:14:40	637000381951300	CommWarrior	05-05-09 00:00:00	06-07-10 14:14:40	353384960419054	CommWarrior	05-05-09 00:00:00
Time	IMEI	Virus	Last Updated																																																						
06-07-16 14:14:40	340260402262764	Cable.D	05-05-09 00:00:00																																																						
06-07-10 14:14:40	16112090340PM	Cable.D	05-04-25 00:00:00																																																						
06-07-10 14:14:40	35196559923407993	Cable.D	05-04-15 00:00:00																																																						
06-07-10 14:14:40	12312341234	Cable.D	05-04-26 00:00:00																																																						
06-07-10 14:14:40	35423500122685502	Cable.D	05-05-09 00:00:00																																																						
06-07-10 14:14:40	35186500064596803	Cable.D	05-05-09 00:00:00																																																						
06-07-10 14:14:40	251065 00-00-0013 8	CommWarrior	05-12-02 00:00:00																																																						
06-07-16 14:14:40	354300009765234	CommWarrior	05-05-09 00:00:00																																																						
06-07-10 14:14:40	35423500215173805	CommWarrior	05-12-02 00:00:00																																																						
06-07-10 14:14:40	3542350022519503	CommWarrior	05-12-02 00:00:00																																																						
06-07-10 14:14:40	35582600000008205	CommWarrior	05-05-09 00:00:00																																																						
06-07-10 14:14:40	637000381951300	CommWarrior	05-05-09 00:00:00																																																						
06-07-10 14:14:40	353384960419054	CommWarrior	05-05-09 00:00:00																																																						
[11] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID but the	<p>Tuvell discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID but the second OS ID provides additional information, the second OS ID is used.</p> <p>Tuvell at Fig. 1 and corresponding text:</p>																																																								

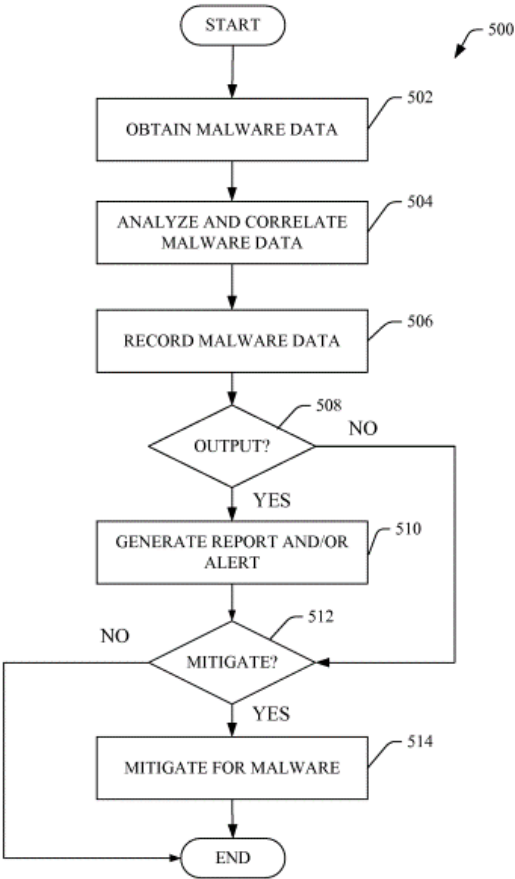
Claim	Exemplary Citation from Tuvell
<p>second OS ID provides additional information, the second OS ID is used.</p>	<div data-bbox="646 251 1344 803"> <pre> graph TD 100[NETWORK MANAGEMENT SYSTEM] 108[USER INTERFACE] 102[RECEIVER COMPONENT] 104[ANALYSIS COMPONENT] 110[MITIGATION COMPONENT] 106[(MALWARE DATA STORE)] 108 --- 100 102 --- 100 104 --- 100 110 --- 100 106 --- 100 </pre> <p>The diagram illustrates a Network Management System (100). It includes a User Interface (108) at the top, a Receiver Component (102) on the left, an Analysis Component (104) in the center, a Mitigation Component (110) on the right, and a Malware Data Store (106) at the bottom. All components are interconnected within the system boundary.</p> </div> <p style="text-align: center;">FIG. 1</p>

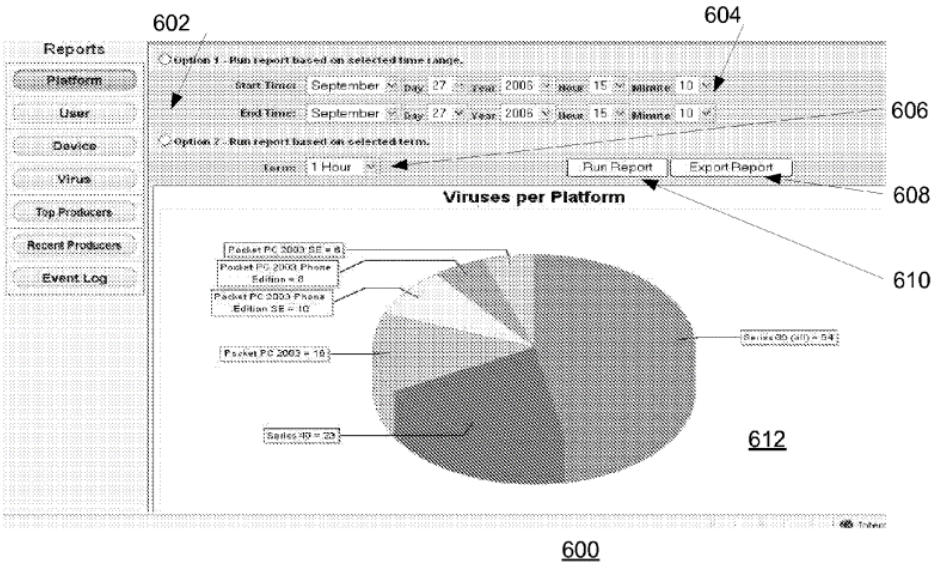
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100) with the following components and connections:</p> <ul style="list-style-type: none"> Receiver Component (102): A dashed box containing: <ul style="list-style-type: none"> Network Analyzer (202): Connected to the Client Data Server (206) and the Malware Scanner (204). Malware Scanner (204): Connected to the Network Analyzer (202) and the Client Data Server (206). Client Data Server (206): Connected to the Network Analyzer (202) and the Malware Scanner (204). User Interface (108): Connected to the Analysis Component (104). Analysis Component (104): Connected to the User Interface (108) and the Mitigation Component (110). Malware Data Store (106): A database connected to the Network Analyzer (202) and the Client Data Server (206). Mitigation Component (110): A dashed box containing: <ul style="list-style-type: none"> Scanner Update Component (208): Connected to the Network Analyzer Update Component (210). Network Analyzer Update Component (210): Connected to the Scanner Update Component (208) and the Firewall Update Component (212). Firewall Update Component (212): Connected to the Network Analyzer Update Component (210) and the Mobile Device Communication Component (214). Mobile Device Communication Component (214): Connected to the Firewall Update Component (212). <p>FIG. 2</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected</p>

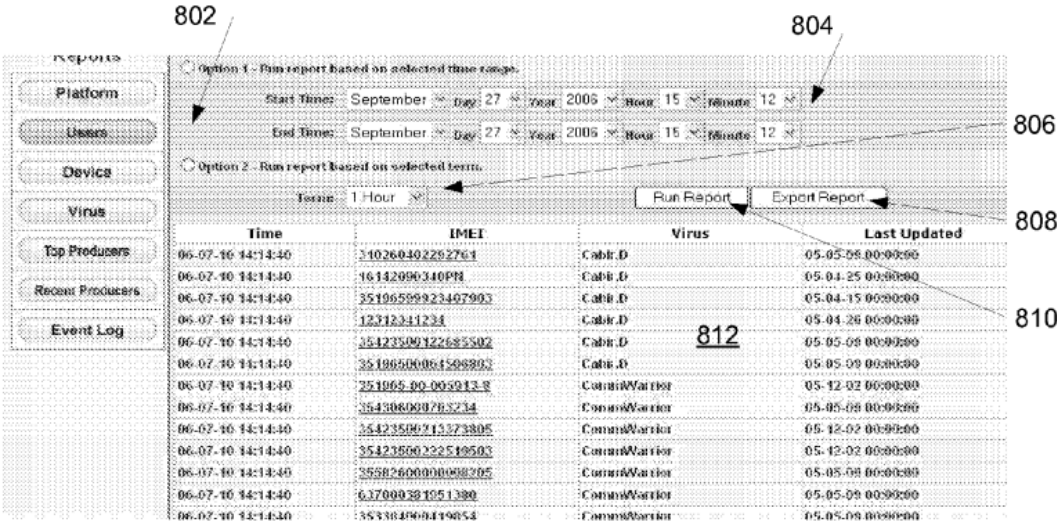
Claim	Exemplary Citation from Tuvell
	<p>mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p> <p>Tuvell at Fig. 3 and corresponding text:</p> <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD MD[MOBILE DEVICE 308] --> IR[/INFECTION REPORT 404/] subgraph MD_Box [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR --> NMS[NETWORK MANAGEMENT SYSTEM 100] NMS --> ACK[/ACK 406/] ACK --> MD_Box </pre> <p>Tuvell at ¶ 49:</p> <p>"Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p>

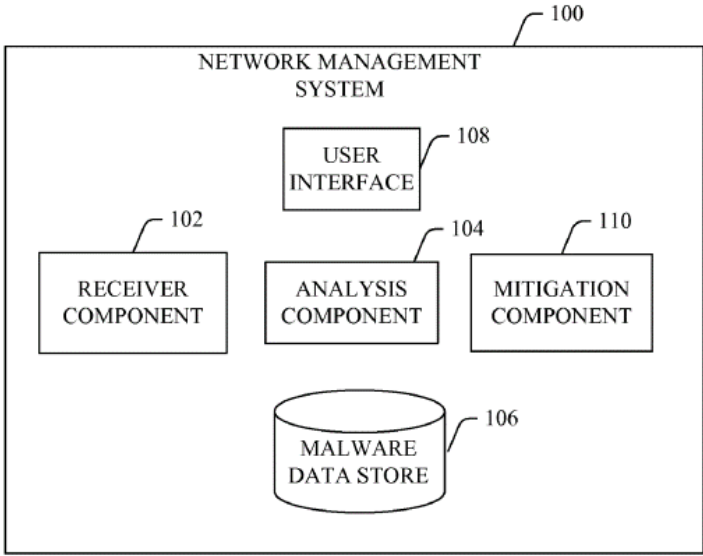
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g., phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at ¶ 54: "In certain embodiments, the analysis component 104 analyzes and correlates malware data obtained by the receiver component 102 and/or maintained by the malware data store 106. In particular, the analysis component 104 correlates data obtained from a variety of Sources (e.g., network traffic, data network traffic and individual mobile devices 308). One function of CoreStats 100 is to assist mobile network administrators and operators to monitor threats to the mobile network 302 thereby identifying the mobile network's 302 Vulnerability to malware. Early detection of the Vulnerability helps them take better preventative measures. CoreStats 100 reports the spreading pattern of malware using collected information from individual mobile devices 308 as well as the network traffic. On the mobile network 302, malware can spread over using short range transmission protocols (e.g., Bluetooth, Infrared), long range or standard network protocols (e.g., TCP/IP. Messaging) or a combination of short and long range protocols. Hence, in order to facilitate reports of infections and spreading patterns of malware across the mobile network 302, CoreStats 100 uses information regarding the infections found in mobile devices 308 as well as those malware found in the network traffic by the network analyzer 202 and malware scanner 204. In particular, CoreStats 100 can generate spreading statistics of long range malware, Such as malware that spreads using the mobile network 302 via TCP/IP. Messaging, and/or other protocols."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

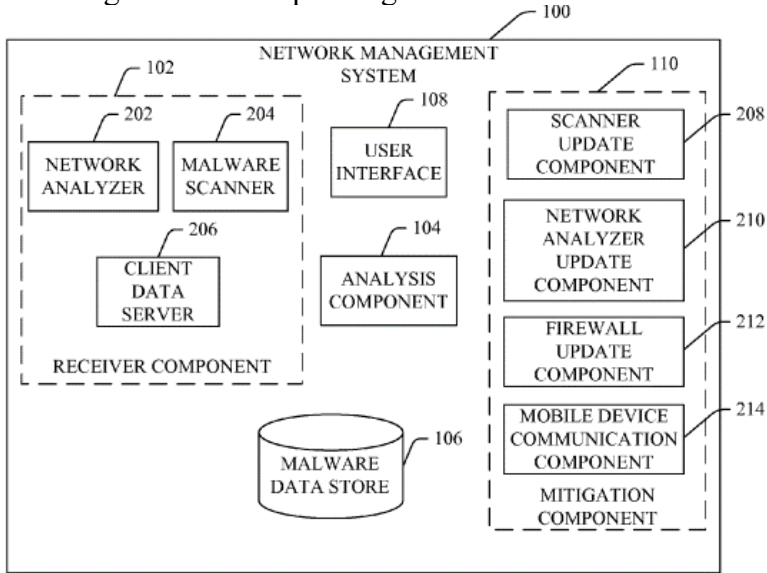
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p>  <p style="text-align: center;">FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
<p>[12] The method of claim 3 wherein sending the notification to the subscriber associated with the access device further comprises replacing the</p>	<p>Tuvell discloses and/or renders obvious that sending the notification to the subscriber associated with the access device further comprises replacing the network address of the access device with a subscriber ID.</p>

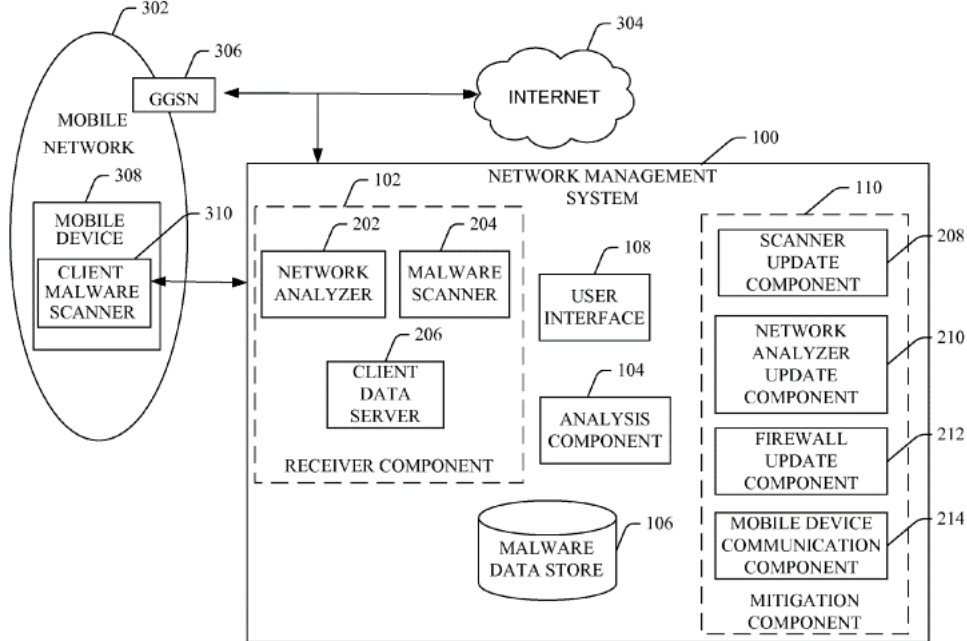
Claim	Exemplary Citation from Tuvell
<p>network address of the access device with a subscriber ID.</p>	<p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

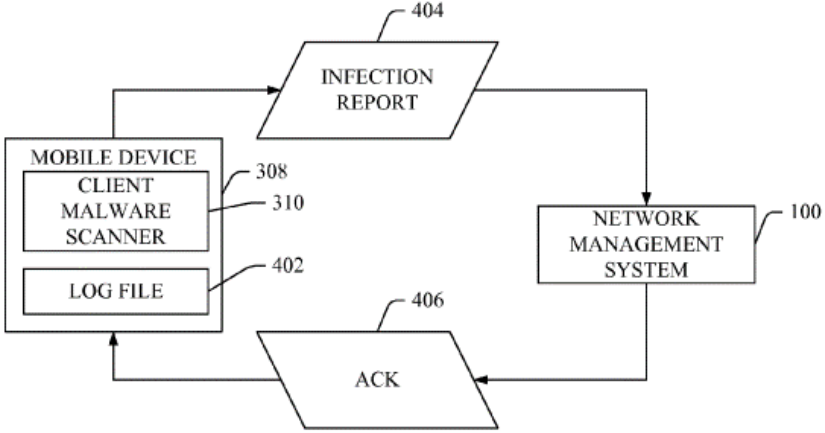
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104</p>

Claim	Exemplary Citation from Tuvell
	<p>generates a malware analysis or report that details and describes instances of malware in the mobile network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 31: "One way is to associate CoreStats 100 with the mobile network administrator's firewall so that the administrator can block identified malicious content. Another way is to report alarms upstream to operational support systems or OAM&P (Operations, Administration, Maintenance, and Provisioning) systems used by network service providers to manage mobile networks."</p> <p>Tuvell at ¶ 32: "In another embodiment, CoreStats 100 facilitates malware prevention by informing mobile device users and/or taking preventative steps at the mobile device. Once CoreStats 100 identifies an infected user, network administrators or operators send messages to a user to alert them to the problem, force an update of the user's mobile device's anti-virus software and definitions, or even disable the mobile device's data connections altogether."</p>

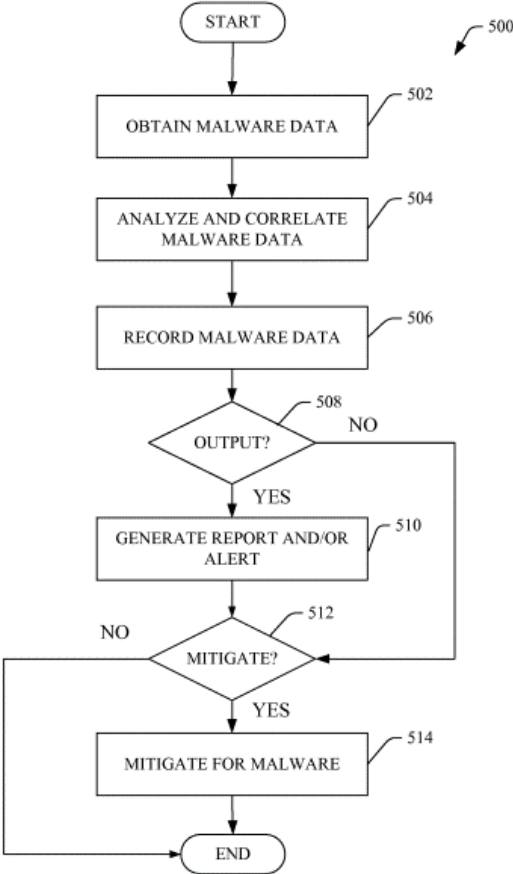
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100) with the following components and connections:</p> <ul style="list-style-type: none"> Receiver Component (102): A dashed box containing: <ul style="list-style-type: none"> Network Analyzer (202) and Malware Scanner (204) connected to a Client Data Server (206). User Interface (108) and Analysis Component (104) are connected to the Receiver Component (102). Malware Data Store (106) is connected to the Analysis Component (104). Scanner Update Component (208), Network Analyzer Update Component (210), Firewall Update Component (212), and Mobile Device Communication Component (214) are connected to the Analysis Component (104). Mitigation Component is connected to the Mobile Device Communication Component (214). <p>FIG. 2</p> <p>Tuvell at ¶ 35:</p> <p>"In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p>

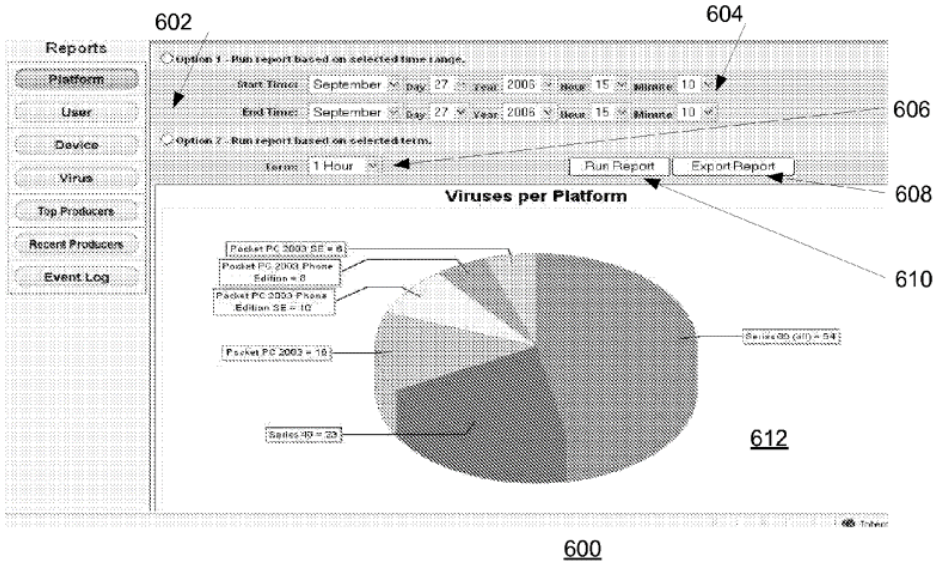
Claim	Exemplary Citation from Tuvell
	<p data-bbox="598 240 1102 272">Tuvell at Fig. 3 and corresponding text:</p>  <p data-bbox="1045 979 1155 1011">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph LR subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre> <p>Tuvell at ¶ 45:</p> <p>"In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46:</p> <p>"Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 48: "In a further embodiment, the client malware scanner 310 generates an infection report 404 that contains information about the detected malware and transmits the infection report 404 to the client data server 206 of CoreStats 100. Report generation transmission is automatically triggered (pushed) upon detection of malware or based upon a periodic fixed time interval. Alternatively, infection reports 404 are maintained in the client device internal log file 402 until queried (pulled) by CoreStats 100. In yet another embodiment, infection reports 404 are delivered to CoreStats 100 using some combination of pulling and pushing. Infection reports 404 are transmitted, for example, using hypertext transfer protocol (http), file transfer protocol (ftp), or any packet data transmission method as would be generally known in the art."</p> <p>Tuvell at ¶ 49: "Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g.,</p>

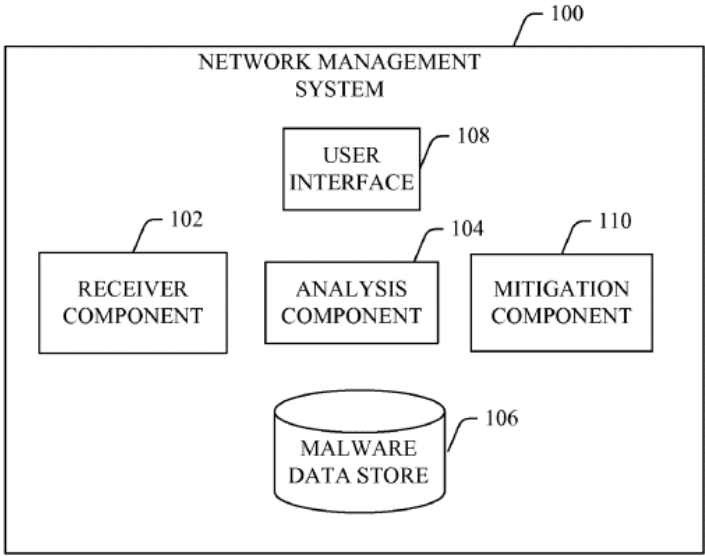
Claim	Exemplary Citation from Tuvell
	<p>phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 60: "At reference number 508 a determination is made as to whether to generate output, such as a report or alert. The determination can be based in whole or in part upon the malware data obtained from various sources. For example, if analysis indicates high levels of malware activity or significant impact on mobile network 302 performance, the determination is made to generate a report and alert or notify network administrators. Alternatively, reports are triggered periodically or upon operator request. In particular, operators can request particular reports via a user interface 108."</p> <p>Tuvell at ¶ 62: "In still other embodiments, a mitigation component 110 notifies a mobile device 308 user, force an update of mobile device 308 software, or even disable the mobile device's 308 data connections."</p> <p>Tuvell at ¶ 63: "Referring now to FIGS. 6-10, exemplary user interface displays are illustrated. As discussed above, CoreStats 100 also performs report generating functions. The analysis component 104 uses both stored and real-time information, including network traffic and individual user information, to generate statistics and dynamic graphs depicting malware activity and network statistics necessary to quantify relative levels of malware activity. For example, the analysis component 104 generates malware analyses, which can be presented by a user interface 108 as straightforward visual reports to alert managers and operators as to which platforms are infected with the most viruses, which viruses are spreading the fastest, the most recently infected mobile devices 308, and which infected mobile devices 308 are spreading the most viruses."</p>

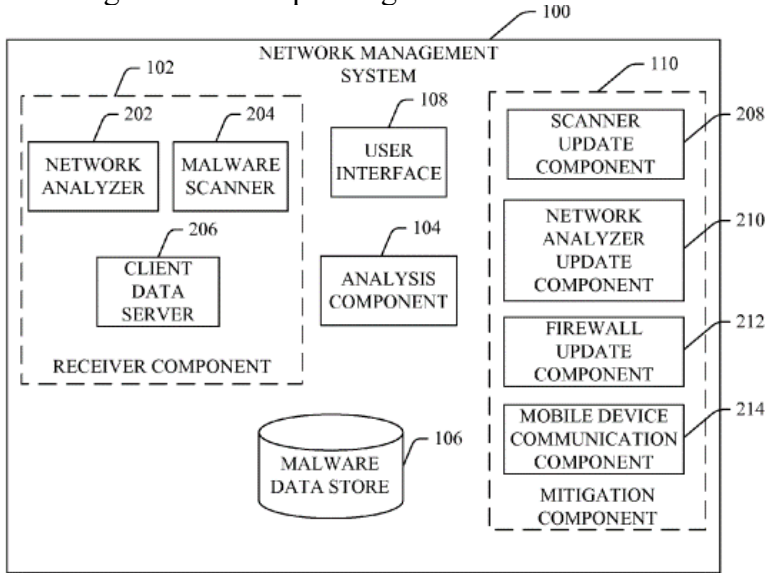
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

Claim	Exemplary Citation from Tuvell																																																								
	<p>Tuvell at Fig. 8 and corresponding text:</p> <div><div><div>Reports</div><div>Platform</div><div>Users</div><div>Device</div><div>Virus</div><div>Top Producers</div><div>Recent Producers</div><div>Event Log</div></div><div><div>Option 1 - Run report based on selected time range.</div><div>Start Time: September 27 2005 15:12</div><div>End Time: September 27 2005 15:12</div><div>Option 2 - Run report based on selected term.</div><div>Term: 1 Hour</div><div>Run Report</div><div>Export Report</div></div><table><thead><tr><th>Time</th><th>IMEI</th><th>Virus</th><th>Last Updated</th></tr></thead><tbody><tr><td>06-07-16 14:14:40</td><td>340260402262764</td><td>Cable.D</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>16112090340PM</td><td>Cable.D</td><td>05-04-25 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35196559923407993</td><td>Cable.D</td><td>05-04-15 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>12312341234</td><td>Cable.D</td><td>05-04-26 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35423500122685502</td><td>Cable.D</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35186500064598803</td><td>Cable.D</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>251065 00-00-0013 8</td><td>CommWarrior</td><td>05-12-02 00:00:00</td></tr><tr><td>06-07-16 14:14:40</td><td>354300009765234</td><td>CommWarrior</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35423500215173805</td><td>CommWarrior</td><td>05-12-02 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>3542350022519503</td><td>CommWarrior</td><td>05-12-02 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>35582600000008205</td><td>CommWarrior</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>637000391951380</td><td>CommWarrior</td><td>05-05-09 00:00:00</td></tr><tr><td>06-07-10 14:14:40</td><td>353384909419054</td><td>CommWarrior</td><td>05-05-09 00:00:00</td></tr></tbody></table></div> <p>FIG. 8</p>	Time	IMEI	Virus	Last Updated	06-07-16 14:14:40	340260402262764	Cable.D	05-05-09 00:00:00	06-07-10 14:14:40	16112090340PM	Cable.D	05-04-25 00:00:00	06-07-10 14:14:40	35196559923407993	Cable.D	05-04-15 00:00:00	06-07-10 14:14:40	12312341234	Cable.D	05-04-26 00:00:00	06-07-10 14:14:40	35423500122685502	Cable.D	05-05-09 00:00:00	06-07-10 14:14:40	35186500064598803	Cable.D	05-05-09 00:00:00	06-07-10 14:14:40	251065 00-00-0013 8	CommWarrior	05-12-02 00:00:00	06-07-16 14:14:40	354300009765234	CommWarrior	05-05-09 00:00:00	06-07-10 14:14:40	35423500215173805	CommWarrior	05-12-02 00:00:00	06-07-10 14:14:40	3542350022519503	CommWarrior	05-12-02 00:00:00	06-07-10 14:14:40	35582600000008205	CommWarrior	05-05-09 00:00:00	06-07-10 14:14:40	637000391951380	CommWarrior	05-05-09 00:00:00	06-07-10 14:14:40	353384909419054	CommWarrior	05-05-09 00:00:00
Time	IMEI	Virus	Last Updated																																																						
06-07-16 14:14:40	340260402262764	Cable.D	05-05-09 00:00:00																																																						
06-07-10 14:14:40	16112090340PM	Cable.D	05-04-25 00:00:00																																																						
06-07-10 14:14:40	35196559923407993	Cable.D	05-04-15 00:00:00																																																						
06-07-10 14:14:40	12312341234	Cable.D	05-04-26 00:00:00																																																						
06-07-10 14:14:40	35423500122685502	Cable.D	05-05-09 00:00:00																																																						
06-07-10 14:14:40	35186500064598803	Cable.D	05-05-09 00:00:00																																																						
06-07-10 14:14:40	251065 00-00-0013 8	CommWarrior	05-12-02 00:00:00																																																						
06-07-16 14:14:40	354300009765234	CommWarrior	05-05-09 00:00:00																																																						
06-07-10 14:14:40	35423500215173805	CommWarrior	05-12-02 00:00:00																																																						
06-07-10 14:14:40	3542350022519503	CommWarrior	05-12-02 00:00:00																																																						
06-07-10 14:14:40	35582600000008205	CommWarrior	05-05-09 00:00:00																																																						
06-07-10 14:14:40	637000391951380	CommWarrior	05-05-09 00:00:00																																																						
06-07-10 14:14:40	353384909419054	CommWarrior	05-05-09 00:00:00																																																						
[13] The method of claim 12 wherein sending the notification to the subscriber associated with the access device further comprises aggregating one	<p>Tuvell at ¶ 66:</p> <p>"Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>																																																								
	<p>Tuvell discloses and/or renders obvious that sending the notification to the subscriber associated with the access device further comprises aggregating one or more detailed alerts having the subscriber ID into a single alert summary associated with the subscriber ID.</p>																																																								

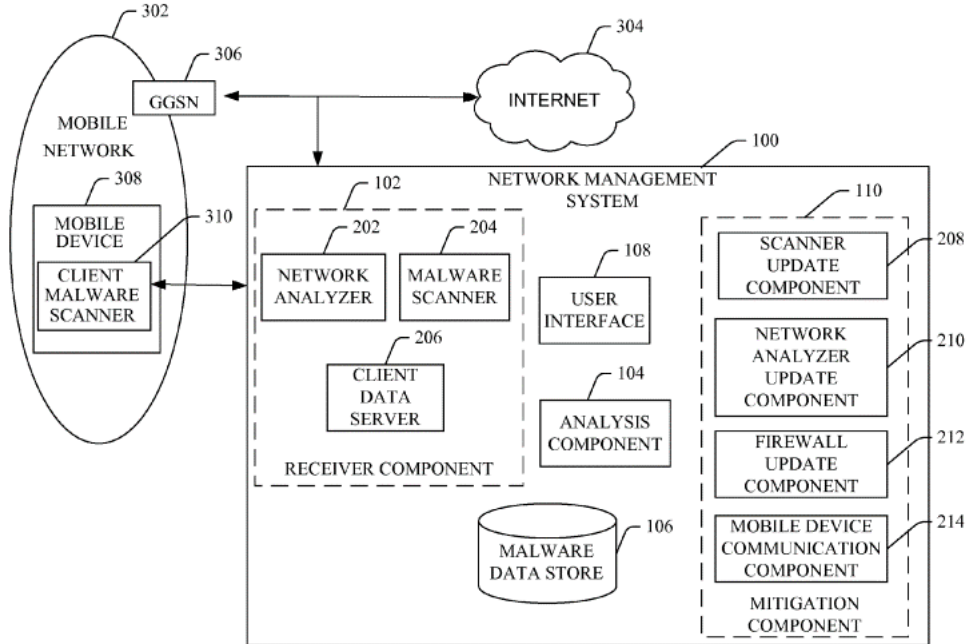
Claim	Exemplary Citation from Tuvell
<p>or more detailed alerts having the subscriber ID into a single alert summary associated with the subscriber ID.</p>	<p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

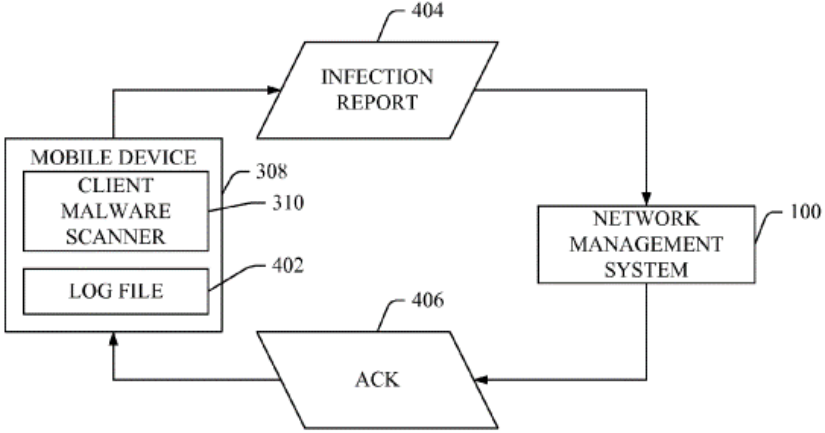
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 27-28: "An analysis component 104 receives information from the receiver component 102 regarding the presence, effects and types of malware impacting a mobile network. In an embodiment, the analysis component 104 is able to synthesize malware data received from the plurality of sources to better analyze the nature and effect of malware. In another embodiment, the analysis component 104</p>

Claim	Exemplary Citation from Tuvell
	<p>generates a malware analysis or report that details and describes instances of malware in the mobile network and the particular mobile devices affected by malware. A malware data store 106 records malware related information including, but not limited to, malware analysis, reports or processed malware data generated by the analysis component 104. In another embodiment, the malware data store 106 stores raw information gathered by the receiver component 102. As used herein, the term "data store" refers to a collection of data (e.g., database, file, cache). In an embodiment, any user specific information is stored in a secure data store to maintain customer privacy."</p> <p>Tuvell at ¶ 31: "One way is to associate CoreStats 100 with the mobile network administrator's firewall so that the administrator can block identified malicious content. Another way is to report alarms upstream to operational support systems or OAM&P (Operations, Administration, Maintenance, and Provisioning) systems used by network service providers to manage mobile networks."</p> <p>Tuvell at ¶ 32: "In another embodiment, CoreStats 100 facilitates malware prevention by informing mobile device users and/or taking preventative steps at the mobile device. Once CoreStats 100 identifies an infected user, network administrators or operators send messages to a user to alert them to the problem, force an update of the user's mobile device's anti-virus software and definitions, or even disable the mobile device's data connections altogether."</p>

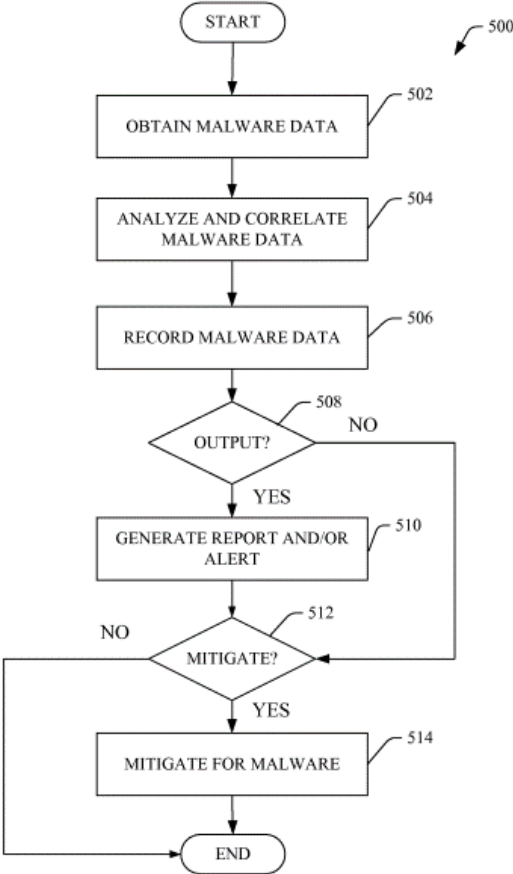
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100). It includes a Receiver Component (102) containing a Network Analyzer (202), a Malware Scanner (204), and a Client Data Server (206). A User Interface (108) and an Analysis Component (104) are also shown. A Malware Data Store (106) is connected to the system. A Mitigation Component (110) includes a Scanner Update Component (208), a Network Analyzer Update Component (210), a Firewall Update Component (212), and a Mobile Device Communication Component (214).</p> <p style="text-align: center;">FIG. 2</p> <p>Tuvell at ¶ 35: "In a further embodiment, the receiver component 102 includes a client data server 206 that receives reports of viruses or other malware from one or more client mobile devices. Individual mobile devices utilize scanning software to determine when malware is present and transmit malware or infection reports to the mobile network. Some representative malware scanning algorithms for mobile devices include, but are not limited to, malware signature searches; hash signature searches as described in U.S. patent application Ser. No. 11/697,647... The malware reports include malware</p>

Claim	Exemplary Citation from Tuvell
	<p>data, such as information regarding infected files, type, or name of infection. In an embodiment, the malware reports include device specific information such as current device hardware, software and/or an identifier for the infected mobile device (e.g., telephone number)."</p> <p>Tuvell at ¶ 36: "In certain embodiments, malware data obtained by the network analyzer 202 and client data server 206 includes device-specific information. In particular, reports received by the client data server 206 include data identifying the particular device that detected the malware. The malware data received by the network analyzer 202 is correlated with data from the mobile network to identify the mobile device that transmitted the infected packet or packets. In both cases, the identity of the affected mobile device is determined. Such device specific information is critical in analysis and reaction to the presence of malware within a network."</p>

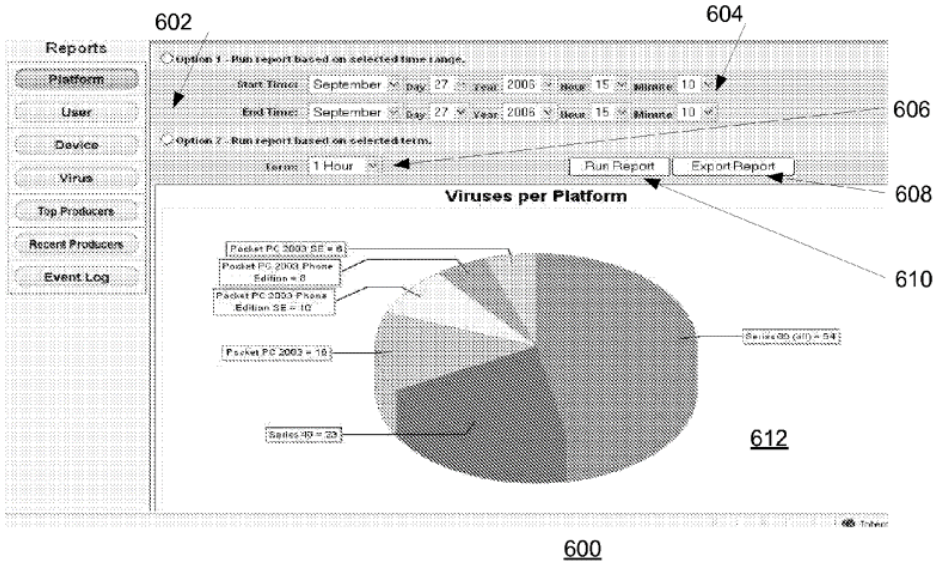
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph LR subgraph 308 [MOBILE DEVICE] 310[CLIENT MALWARE SCANNER] 402[LOG FILE] end 404[/INFECTION REPORT/] 100[NETWORK MANAGEMENT SYSTEM] 406[/ACK/] 310 --> 404 404 --> 100 100 --> 406 406 --> 308 </pre> <p>Tuvell at ¶ 45:</p> <p>"In another embodiment, CoreStats 100 receives communications from mobile client devices (also referred to as mobile devices or client devices) 308. In certain embodiments, mobile client devices 308 include a client malware scanner 310 capable of detecting malware on mobile client devices 308. Once malware is detected, the client malware scanner 310 generates an infection report 404 that provides malware data to the receiver component 102 of the CoreStats system 100. The malware data can be used to reconfigure the malware detection algorithms for malware in the network malware scanner 204 and client malware scanners 310."</p> <p>Tuvell at ¶ 46:</p> <p>"Turning now to FIG. 4, a block diagram depicting communication between CoreStats 100 and a mobile device 308 is illustrated. In one embodiment, upon detecting malware, a mobile device 308 generates or updates an internal log file (or log file) 402, recording malware information. The internal log file 402 can be plain text containing the name of the infected file and the name of the malware that infected the file as a semi-colon delimited text file."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 48: "In a further embodiment, the client malware scanner 310 generates an infection report 404 that contains information about the detected malware and transmits the infection report 404 to the client data server 206 of CoreStats 100. Report generation transmission is automatically triggered (pushed) upon detection of malware or based upon a periodic fixed time interval. Alternatively, infection reports 404 are maintained in the client device internal log file 402 until queried (pulled) by CoreStats 100. In yet another embodiment, infection reports 404 are delivered to CoreStats 100 using some combination of pulling and pushing. Infection reports 404 are transmitted, for example, using hypertext transfer protocol (http), file transfer protocol (ftp), or any packet data transmission method as would be generally known in the art."</p> <p>Tuvell at ¶ 49: "Infection reports 404 typically comprise information such as, but not limited to, detailed virus/threat vector information and mobile device related information, including type of mobile device 308, operating system, software and versions, and user information and mobile device 308 identifier. In an exemplary embodiment, the infection report 404 contains product identification that identifies the client malware scanner 310 software. For example, product identification includes, but is not limited to, a product identifier, major version identifier, minor version identifier and also a patch version as follows: "productid+majorversion+minorversion+patchversion." The infection report 404 can also include the infected filename and a unique identifier for the infected application, the name of the malware infection and the date and time of the infection. In addition, the infection report 404 can include mobile device 308 information, such as the identification of the mobile phone (e.g., phone number), firmware of the particular mobile device 308 (e.g., operating system information) and the software version of the mobile device 308."</p> <p>Tuvell at ¶ 52: "In an exemplary embodiment, the malware data store 106 maintains information obtained based upon network traffic analysis, including, but not limited to, Internet protocol (IP) address of the network level packet analyzer and the time at which the packet was detected. The malware data store 106 maintains records regarding the infected data, such as virus name, infected file name, infected file size, infected packet size and infected packet number. The malware data store 106 also maintains packet source related information, such as the source IP, source port and even source identifier (e.g.,</p>

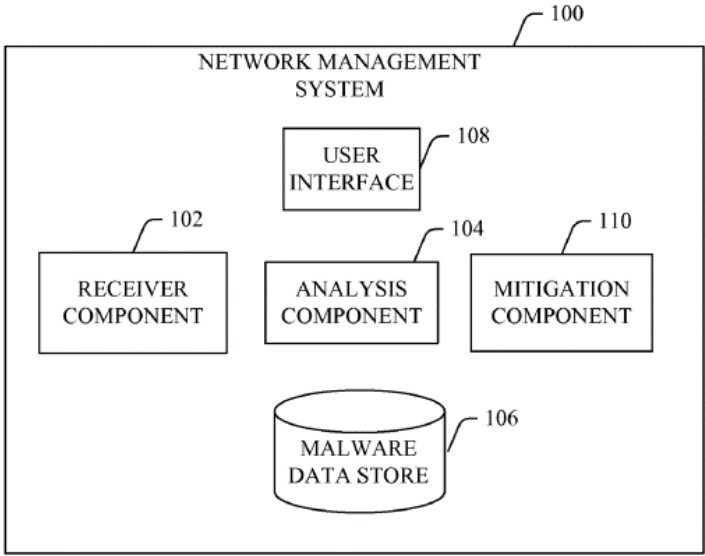
Claim	Exemplary Citation from Tuvell
	<p>phone number). Moreover, destination information such as destination IP address, destination port and destination phone number can be recorded for analysis and reporting. The malware data store 106 can also maintain a record of the particular protocol name used for transmission of the packet."</p> <p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 508 -- NO --> 512{MITIGATE?} 510 --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 512 -- NO --> END([END]) 514 --> END </pre> <p>FIG. 5</p>

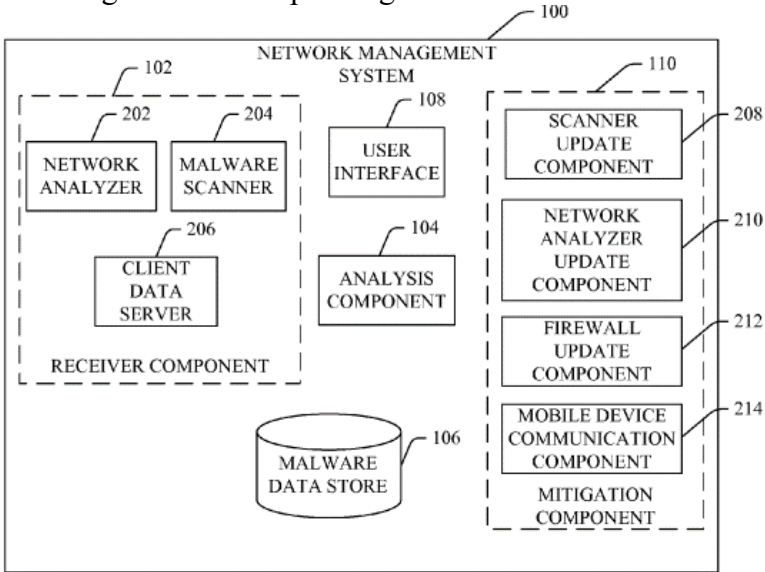
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 58: "Turning now to FIG. 5, a flowchart depicting an exemplary methodology 500 for mobile network management is illustrated. At reference number 502 malware data is obtained. In an embodiment, malware data is obtained from a plurality of sources, such as individual mobile devices 308, mobile network traffic and a computer network. In another embodiment, malware data includes information that specifies a particular mobile device or devices 308 affected by malware. For example, the malware data can include an identifier for the mobile device 308 reporting the malware or an identifier for the mobile device 308 sending and/or receiving a data packet containing malware."</p> <p>Tuvell at ¶ 60: "At reference number 508 a determination is made as to whether to generate output, such as a report or alert. The determination can be based in whole or in part upon the malware data obtained from various sources. For example, if analysis indicates high levels of malware activity or significant impact on mobile network 302 performance, the determination is made to generate a report and alert or notify network administrators. Alternatively, reports are triggered periodically or upon operator request. In particular, operators can request particular reports via a user interface 108."</p> <p>Tuvell at ¶ 62: "In still other embodiments, a mitigation component 110 notifies a mobile device 308 user, force an update of mobile device 308 software, or even disable the mobile device's 308 data connections."</p> <p>Tuvell at ¶ 63: "Referring now to FIGS. 6-10, exemplary user interface displays are illustrated. As discussed above, CoreStats 100 also performs report generating functions. The analysis component 104 uses both stored and real-time information, including network traffic and individual user information, to generate statistics and dynamic graphs depicting malware activity and network statistics necessary to quantify relative levels of malware activity. For example, the analysis component 104 generates malware analyses, which can be presented by a user interface 108 as straightforward visual reports to alert managers and operators as to which platforms are infected with the most viruses, which viruses are spreading the fastest, the most recently infected mobile devices 308, and which infected mobile devices 308 are spreading the most viruses."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 6 and corresponding text:</p>  <p style="text-align: center;">FIG. 6</p> <p>Tuvell at ¶ 64:</p> <p>"Referring to now to FIG. 6, a sample malware per platform report 600 is illustrated. The malware per platform report 600 illustrates which platforms are infected with the most malware. The sample malware per platform report 600 comprises option selections 602 for generating a report regarding a selectable interval of time in the past 604 or the most current period of time 606. The report 600 is presented on a display screen 610, as shown. Alternatively, reports 600 are exported 608 to a data structure. For example, reports 600 are output to semi-colon delimited text files. When presented on a display screen 610, the data is presented any number of ways including, for example, a graphical representation 612 of the number of viruses per platform."</p>

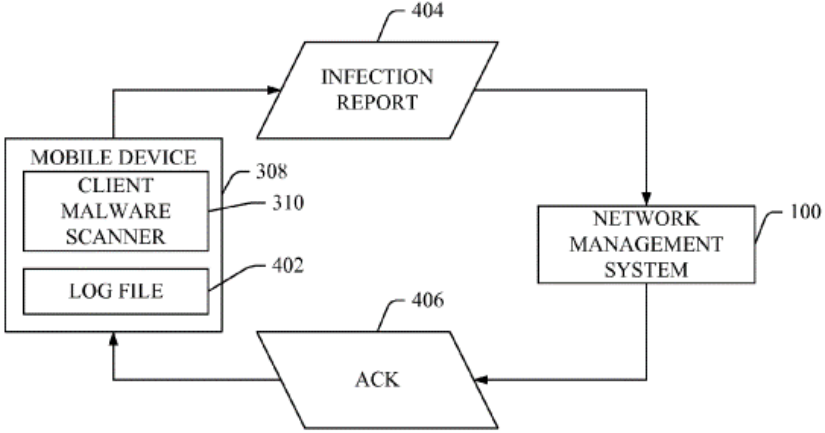
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 8 and corresponding text:</p> <p>FIG. 8</p> <p>Tuvell at ¶ 66: "Referring now to FIG. 8, a sample user infection report 800 is illustrated. The sample user infection report 800 shows recently infected users. In an embodiment, the sample user infection report 800 comprises option selections 802 for generating a report 800 regarding a selectable interval of time in the past 804 or the most current period of time 806. The report 800 is presented on a display screen 810 or is exported 808 to a data structure. For example, the report 800 is exported to a semi-colon delimited text file. When presented on a display screen 810, the data is presented any number of ways including, for example, a text list 812 of which platforms are infected by which viruses."</p>
[14] The method of claim 12 wherein the one or more computing devices are coupled to the access device through a local area network (LAN), the access device	<p>Tuvell discloses and/or renders obvious that the one or more computing devices are coupled to the access device through a local area network (LAN), the access device providing network address translation (NAT) or is coupled to NAT device to share the network address of the access device.</p> <p>Tuvell at Abstract:</p>

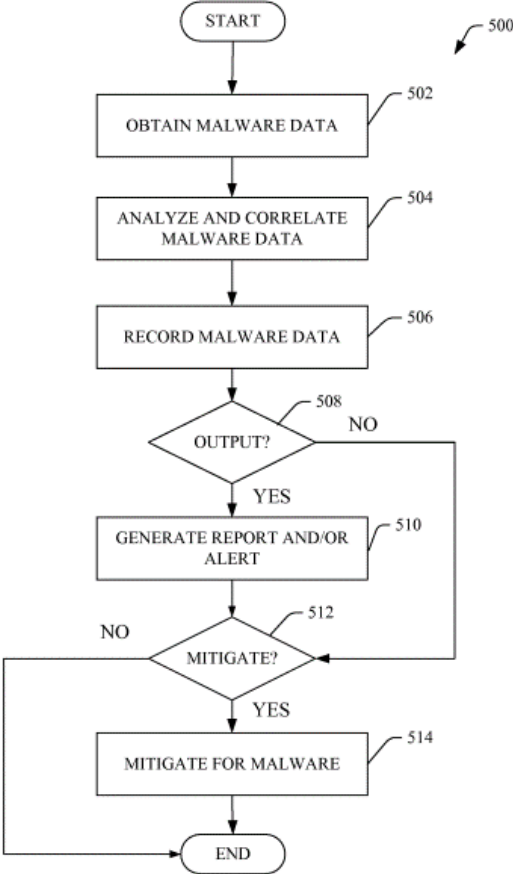
Claim	Exemplary Citation from Tuvell
<p>providing network address translation (NAT) or is coupled to NAT device to share the network address of the access device.</p>	<p>"A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 2: "The present invention relates generally to systems, devices, and methods for detecting malware in mobile networks and mobile devices."</p> <p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 25: "FIG. 1 depicts an exemplary network management system 100, also referred to herein as CoreStats, that provides for reporting and visualizing viruses on mobile networks. As used herein, the term "exemplary" indicates a sample or example. It is not indicative of preference over other aspects or embodiments. The network management system 100 includes a receiver component 102 that obtains</p>

Claim	Exemplary Citation from Tuvell
	<p>or receives malware data for a mobile network (not shown). As used herein, the term "component" refers to hardware, software, firmware or any combination thereof."</p> <p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100) enclosed in a dashed box. It includes several sub-components and a data store:</p> <ul style="list-style-type: none"> Receiver Component (102): A dashed box containing a Network Analyzer (202) and a Malware Scanner (204). Client Data Server (206): A box connected to the Receiver Component. User Interface (108): A box connected to the central system. Analysis Component (104): A box connected to the central system. Malware Data Store (106): A cylinder representing a database, connected to the central system. Mitigation Component (110): A dashed box containing four sub-components: <ul style="list-style-type: none"> Scanner Update Component (208) Network Analyzer Update Component (210) Firewall Update Component (212) Mobile Device Communication Component (214) <p>FIG. 2</p>

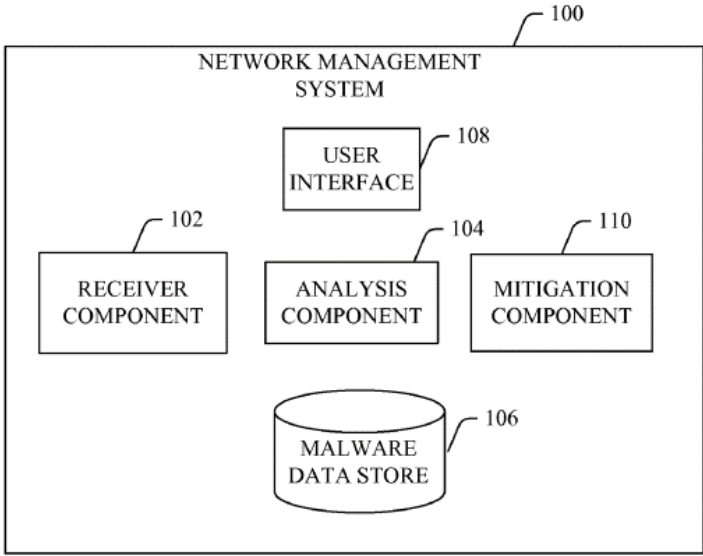
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p> <p>FIG. 3</p>

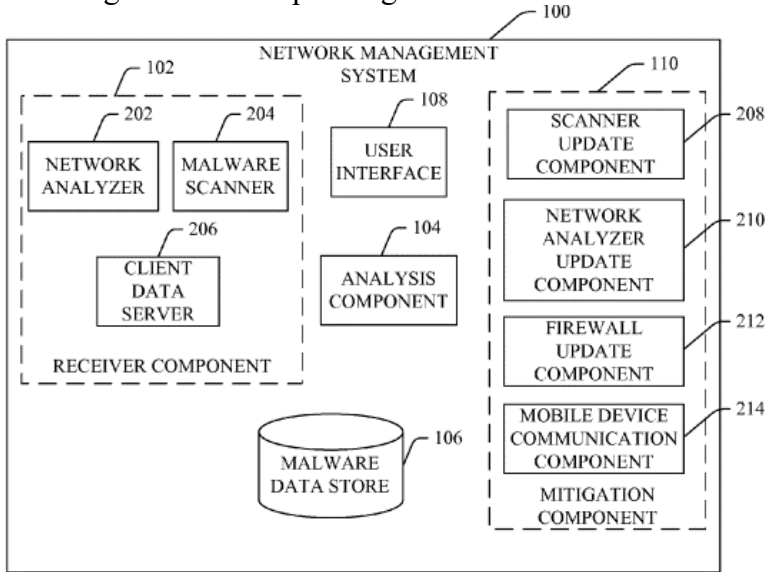
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre>

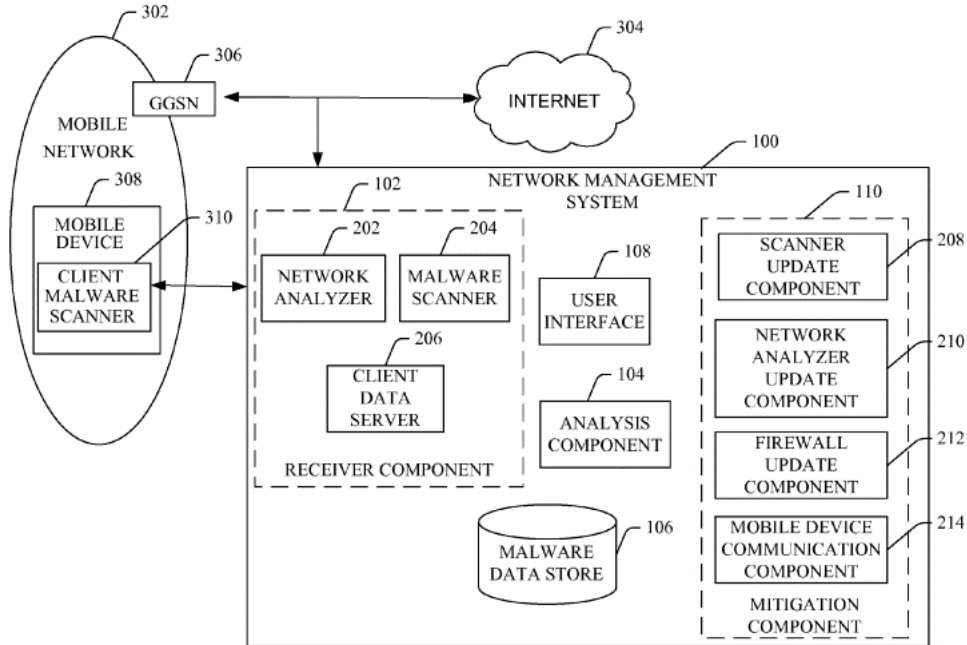
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 510 --> 512{MITIGATE?} 508 -- NO --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 514 --> END([END]) 512 -- NO --> END </pre> <p>FIG. 5</p> <p>Tuvell at ¶ 43-44: "Referring now to FIG. 3, an exemplary deployment of a sample CoreStats system 100 in a network environment is depicted. FIG. 3 illustrates a deployment of CoreStats 100 between the edge of a mobile network 302 and the Internet 304, although it can also be deployed effectively at various other points in the mobile network 302 depending upon the network topology and desired coverage.</p>

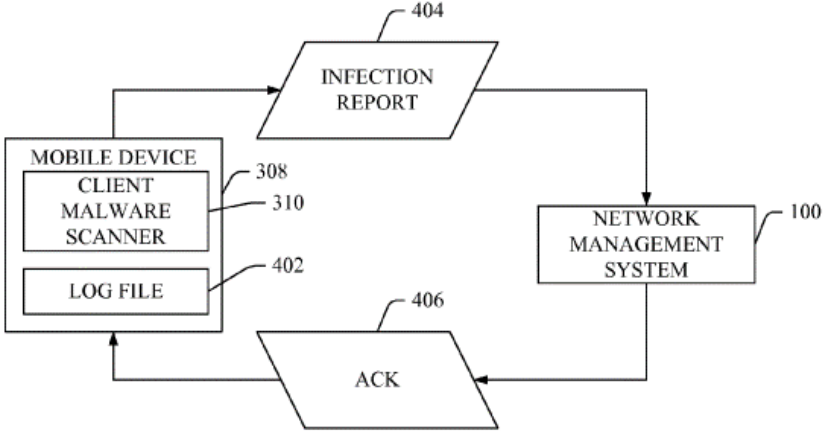
Claim	Exemplary Citation from Tuvell
	<p>The network analyzer 202 monitors and evaluates all traffic going from the mobile network 302 to the public data networks (e.g., the Internet 304) and vice-versa. The network analyzer 202 can intercept packets on either side of a firewall (not shown). In an embodiment, CoreStats 100 monitors a mobile network (or operator's network) 302 by monitoring or packet sniffing IP packets passing from the Gateway General Packet Radio Service ("GPRS") Support Node or Gateway GPRS Support Node ("GGSN") 306 and the Internet 304. In an embodiment, CoreStats 100 is deployed between the edge of the mobile network 302 and the Internet 304. The GGSN 306 links the access dependent Radio Access Network (RAN), shown on the figure as the mobile network 302, to the access independent Internet 304. RAN comprises the entire radio/wireless network with a variety of protocols for data transfer (e.g., CDMA, GPRS, 802.11). The GGSN 306 acts as a gateway between the mobile network 302 and the Internet 304, converting access-specific packet data to IP packets and vice-versa. As discussed above, the intercepted packets are processed by the malware scanner 204 and the resulting malware data is provided to the analysis component 104."</p>
<p>[15PRE] A system for network based malware detection in a service provider network, the system comprising:</p>	<p><i>See [1PRE].</i></p>
<p>[15A] a plurality of network sensors coupled to the service provider network, each of the network sensors for:</p>	<p>Tuvell discloses and/or renders obvious a plurality of network sensors coupled to the service provider network, each of the network sensors.</p> <p>Tuvell at Abstract: "A network management system monitors malware within a mobile network. The system comprises a receiver component that obtains data regarding malware in the mobile network. The data is obtained from a first source and a second source, where the first source is of a different type than the second source. The monitoring system also includes an analysis component that generates a malware analysis of the mobile network as a function of the data."</p> <p>Tuvell at ¶ 2: "The present invention relates generally to systems, devices, and methods for detecting malware in mobile networks and mobile devices."</p>

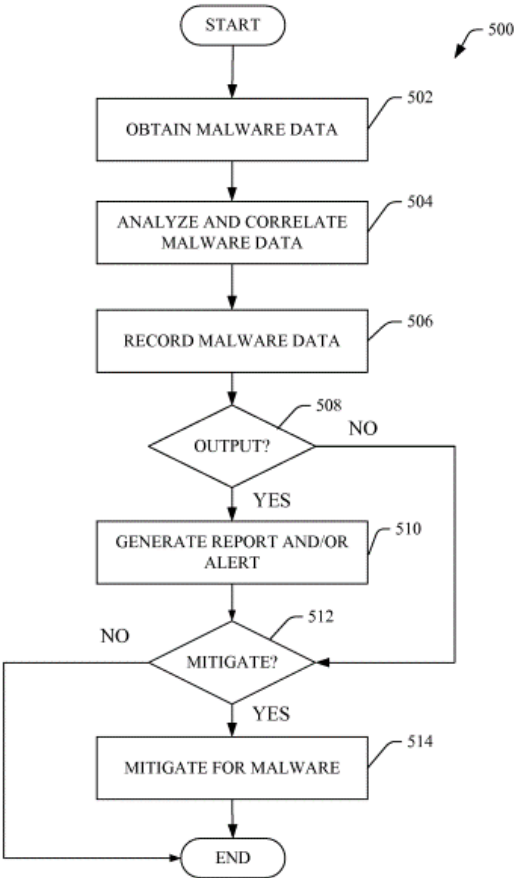
Claim	Exemplary Citation from Tuvell
	<p>Tuvell at ¶ 5: "Currently, once malware has been identified and analyzed, it can be detected using signatures extracted from the malware and cleaned according to its specific ways of spreading and infecting. The more difficult problem is in identifying new malware as early as possible to prevent it from proliferating. Although firewalls are used in the mobile network to limit or forbid suspicious behavior, no existing methods provide a comprehensive security solution towards eliminating all new malware. This is at least in part because the forms and functionalities of new malware are unpredictable. Also, malware can propagate through any number of locations making it impossible to capture all new malware samples at a single location. To effectively combat new malware, new malware samples need to be quickly gathered, identified, and analyzed as soon as they appear on the network so that cleaning schemes using signature schemes or other methods can be implemented before the malware has had a chance to widely proliferate. The sooner a sample of new malware is obtained, the sooner the mobile network can be protected against the new malware and the less damage the malware will ultimately cause."</p> <p>Tuvell at ¶ 9: "The present invention is a system and method for reporting and visualizing worm and other virus attacks on mobile networks. The system and method provides a comprehensive means for collecting, reporting, and providing visual depictions of information regarding the propagation and effect of worms, viruses and other malware on a network. Malware and virus as used hereafter are meant to encompass a broad definition of malicious or harmful software."</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 1 and corresponding text:</p>  <p style="text-align: center;">FIG. 1</p> <p>Tuvell at ¶ 25: "FIG. 1 depicts an exemplary network management system 100, also referred to herein as CoreStats, that provides for reporting and visualizing viruses on mobile networks. As used herein, the term "exemplary" indicates a sample or example. It is not indicative of preference over other aspects or embodiments. The network management system 100 includes a receiver component 102 that obtains</p>

Claim	Exemplary Citation from Tuvell
	<p>or receives malware data for a mobile network (not shown). As used herein, the term "component" refers to hardware, software, firmware or any combination thereof."</p> <p>Tuvell at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a Network Management System (100) enclosed in a dashed box. It includes several sub-components and a data store:</p> <ul style="list-style-type: none"> Receiver Component (102): A dashed box containing a Network Analyzer (202) and a Malware Scanner (204). Client Data Server (206): A box connected to the Receiver Component. User Interface (108): A box connected to the central system. Analysis Component (104): A box connected to the central system. Malware Data Store (106): A cylinder representing a database, connected to the central system. Mitigation Component (110): A dashed box containing four sub-components: <ul style="list-style-type: none"> Scanner Update Component (208) Network Analyzer Update Component (210) Firewall Update Component (212) Mobile Device Communication Component (214) <p>FIG. 2</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 3 and corresponding text:</p>  <p style="text-align: center;">FIG. 3</p>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 4 and corresponding text:</p>  <pre> graph TD subgraph MD [MOBILE DEVICE 308] CMS[CLIENT MALWARE SCANNER 310] LF[LOG FILE 402] end IR[/INFECTION REPORT 404/] NMS[NETWORK MANAGEMENT SYSTEM 100] ACK[/ACK 406/] CMS --> IR IR --> NMS NMS --> ACK ACK --> LF </pre>

Claim	Exemplary Citation from Tuvell
	<p>Tuvell at Fig. 5 and corresponding text:</p>  <pre> graph TD START([START]) --> 502[OBTAIN MALWARE DATA] 502 --> 504[ANALYZE AND CORRELATE MALWARE DATA] 504 --> 506[RECORD MALWARE DATA] 506 --> 508{OUTPUT?} 508 -- YES --> 510[GENERATE REPORT AND/OR ALERT] 510 --> 512{MITIGATE?} 508 -- NO --> 512 512 -- YES --> 514[MITIGATE FOR MALWARE] 514 --> END([END]) 512 -- NO --> END </pre> <p>FIG. 5</p> <p>Tuvell at ¶ 43-44: "Referring now to FIG. 3, an exemplary deployment of a sample CoreStats system 100 in a network environment is depicted. FIG. 3 illustrates a deployment of CoreStats 100 between the edge of a mobile network 302 and the Internet 304, although it can also be deployed effectively at various other points in the mobile network 302 depending upon the network topology and desired coverage.</p>

Claim	Exemplary Citation from Tuvell
	<p>The network analyzer 202 monitors and evaluates all traffic going from the mobile network 302 to the public data networks (e.g., the Internet 304) and vice-versa. The network analyzer 202 can intercept packets on either side of a firewall (not shown). In an embodiment, CoreStats 100 monitors a mobile network (or operator's network) 302 by monitoring or packet sniffing IP packets passing from the Gateway General Packet Radio Service ("GPRS") Support Node or Gateway GPRS Support Node ("GGSN") 306 and the Internet 304. In an embodiment, CoreStats 100 is deployed between the edge of the mobile network 302 and the Internet 304. The GGSN 306 links the access dependent Radio Access Network (RAN), shown on the figure as the mobile network 302, to the access independent Internet 304. RAN comprises the entire radio/wireless network with a variety of protocols for data transfer (e.g., CDMA, GPRS, 802.11). The GGSN 306 acts as a gateway between the mobile network 302 and the Internet 304, converting access-specific packet data to IP packets and vice-versa. As discussed above, the intercepted packets are processed by the malware scanner 204 and the resulting malware data is provided to the analysis component 104."</p> <p>Tuvell at ¶ 77-81:</p> <p>"Collection Agents A collection agent 1102, 202 is a device which is placed at various points in the mobile network 302 in order to collect samples being transmitted over the network executables, wherein a sample is transmitted data containing executable code. The type of collection agent 1102, 202 and the protocols monitored by the protocol handlers 1114 a, 1114 b are dependent not only upon the anticipated data loads and protocols being transmitted, but also on the mechanism used by the malware to accomplish its tasks, if known. The use of two types of collection agents, e.g., honeypots 1102 and network sniffers or analyzers 202, provides a network service provider the best opportunity for early detection of malicious applications before they have had a chance to proliferate widely across a service provider's mobile network 302. Honeypots: Honeypots, collection agent 1102, are typically stand-alone devices that have open network ports for unobtrusively accepting messages that are broadcast or specifically sent to them from malware infected mobile devices 1104, 1106. A typical feature of many malicious applications is that they attempt to forward copies of themselves automatically to other networked devices 308, thereby allowing themselves to spread through the mobile network 302 like a virus. It is possible for malicious applications to copy themselves to nearby mobile devices 308 using ad hoc or similar point-to-point type networks, instead of across the much larger service provider's mobile network 302. This makes it difficult, if not impossible, for the service provider to detect malware because the malware may not be transmitting across the service</p>

Claim	Exemplary Citation from Tuvell
	<p>provider's mobile network 302. A person with a malware infected mobile device 1104, 1106 may during the course of single day come into range of tens, if not hundreds, of other mobile devices 308, possibly infecting many of them. In such cases, the malware may be discovered only at a later date when much of the damage has already been done. Therefore, honeypots 1102 allow earlier detection of malicious applications by virtue of the fact that they are not in the core of the service provider's mobile network 302, as a network analyzer 202 collection agent would be, but rather are spread strategically in the periphery. Honeypots 1102 can be configured with a Bluetooth protocol handler 1114 a and a Wi-Fi protocol handler 1114 b. Bluetooth enabled honeypots 1102 are mobile devices 308 or laptops that are placed in areas where there is typically a lot of wireless communication. The aim is to capture Bluetooth broadcast messages 1108 b containing malicious executables sent from other nearby Bluetooth enabled, malware infected mobile devices 308. Target areas include airports, restaurants, downtown areas, and public parks. Wi-Fi enabled honeypots 1102 are mobile devices 308 or laptops that are placed in areas where there is a possibility of hacking and illegal access taking place. The aim is to allow illegal access of the honeypot collection agent 1102 in order to capture the malicious executable files sent using the Wi-Fi protocol 1108 a from malware infected mobile devices 1104. Target areas include banks and stock exchanges. Because such honeypot collection agents 1102 can be installed in locations outside of the provider's mobile network 302, calls from such collection agents 1102 may be switched across the Public Switched Telephone Network, or PSTN (not shown). Preferably such collection agents 1102 are switched through the provider's mobile network 302, when possible, as shown by wireless data connection 1108 e, to reduce potential calling costs with other service providers. In different embodiments, honeypot collection agents 1102 use a number of communication interfaces to connect to a sample collection center 1112 of a network management system 100. For example, such communication interfaces may include placing calls over telephony interfaces such as POTS lines or Plain Old telephone Service, ISDN, or other bearer channel technologies, or using data communication networks such as legacy serial or packet-based networks, TCP/IP, xDSL, and fiber-based technologies. Additionally, such collection agents 1102 use wireless interfaces including, but not limited to, WiFi, IEEE 802.11 or more generically 802.x wireless interfaces."</p>
<p>[15B] receiving one or more transmission control protocol (TCP) packets originating from an access</p>	<p>See [1A].</p>

Claim	Exemplary Citation from Tuvell
device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;	
[15C] determining an operating system identifier (OS ID) associated with the TCP session and the computing device associated;	<i>See</i> [1B].
[15D] determining if malware is present in the TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets; and	<i>See</i> [1C].
[15E] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.	<i>See</i> [1D].
[16PRE] The system of claim 15 wherein the plurality of network sensors	<i>See</i> [2PRE].

Claim	Exemplary Citation from Tuvell
in determining the OS ID further:	
[16A] determine a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first OS ID;	<i>See</i> [2A].
[16B] determine an application parameter associated with the TCP session if present, and identifying a second OS ID from the application parameter; and	<i>See</i> [2B].
[16C] wherein the identified OS ID in the alert comprises one of the first OS ID or the second OS ID associated with one or more computing devices coupled to the access device.	<i>See</i> [2C].
[17] The system of claim 15 wherein the plurality of network sensors further: send a notification to a subscriber associated with the network address associated with the access device, the notification identifying a remediation	<i>See</i> claim 3

Claim	Exemplary Citation from Tuvell
portal to remove the determined malware in the alert.	
[18] The system of claim 17 wherein the plurality of network sensors send the notification to the subscriber associated with the access device by retrieving a subscriber ID from a subscriber ID repository and replacing the network address of the access device with the subscriber ID.	See claim 12.
[19] The system of claim 16 wherein the plurality of network sensors determine the first OS ID by comparing one or more protocol parameters of the one or more TCP packets having a SYN flag associated with the session, the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS ID associated with parameters.	See claim 7.
[20] The system of claim 19 wherein the plurality of network sensors determine	See claim 8.

Claim	Exemplary Citation from Tuvell
the application parameter associated with the session and identifies the second OS ID from the application parameter by inspecting an HTTP user agent string to determine the second OS ID.	
[21] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the session is identified by the OS ID providing a more reliable OS ID.	<i>See claim 9.</i>
[22] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID either OS ID is used.	<i>See claim 10.</i>
[23] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID matches	<i>See claim 11.</i>

Claim	Exemplary Citation from Tuvell
the first OS ID but the second OS ID provides additional information, the second OS ID is used.	
[24PRE] A computer readable memory containing instructions for network based malware detection in a service provider network, the instructions when executed by a processor performing:	<i>See</i> [1PRE].
[24A] receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;	<i>See</i> [1A].
[24B] determining an operating system identifier (OS ID) associated with the TCP session and the computing device;	<i>See</i> [1B].
[24C] determining if malware is present in the	<i>See</i> [1C].

Claim	Exemplary Citation from Tuvell
TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets; and	
[24D] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.	<i>See</i> [1D].
[25PRE] The system of claim 16 further comprising:	<i>See</i> [4PRE].
[25A] a remediation portal coupled to the service provider network, the remediation portal for:	<i>See</i> [4A].
[25B] identifying a subscriber accessing the remediation portal;	<i>See</i> [4A].
[25C] identifying the OS associated with the computing device used by the subscriber to access the remediation portal;	<i>See</i> [4B].
[25D] comparing the OS associated with the computing device to the OS ID identified in the alert; and	<i>See</i> [4C].

Claim	Exemplary Citation from Tuvell
[25E] providing malware remediation to the computing device.	<i>See</i> [4D].

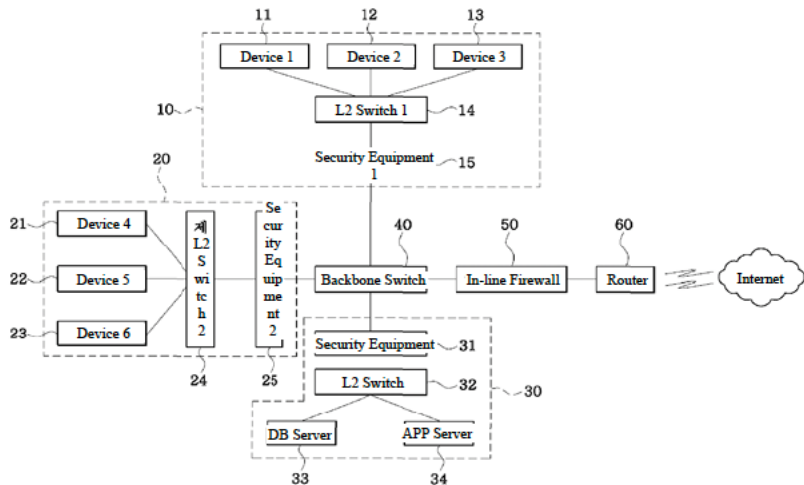
Ex. I-8 - Invalidity of U.S. Patent No. 8,635,697 against KR100765340B1 to Lee ("Lee") and Certified Translation

Fortinet, Inc. ("Fortinet") provides this chart subject to all reservations, objections, statements, and disclaimers set forth herein and in Fortinet's Preliminary Invalidity Contentions Cover Pleading, as well as any amendment, supplement, or modification thereof, which are incorporated herein by reference in their entirety.

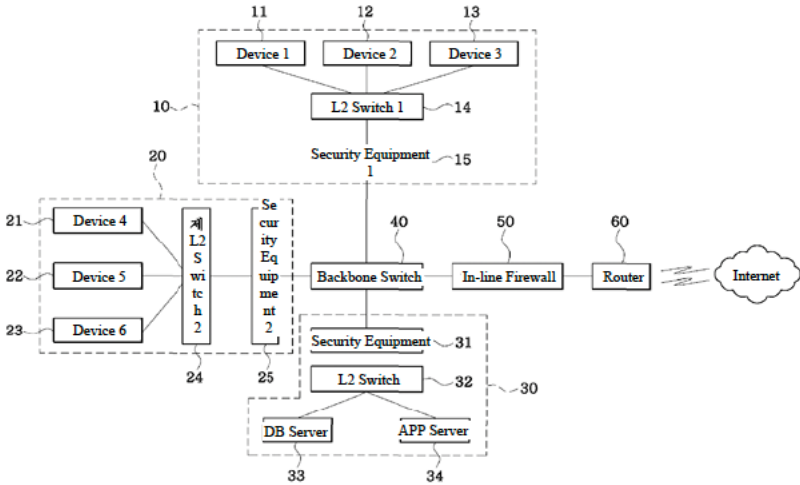
On information and belief, and subject to further investigation and discovery, Lee was published April 26, 2006 and is thus available as prior art under at least §102(a) and (b).

As illustrated in the chart below, Lee anticipates the asserted claims of the '697 Patent. To the extent the Lee is found not to expressly disclose certain limitations in the asserted claims, such limitations are inherent. To the extent the Lee is found not to anticipate any asserted claims or claim elements of the '697 Patent, the reference nevertheless renders those claims or claim elements obvious under 35 U.S.C. § 103, either alone or in combination with other art identified in the cover pleading or herein. These Preliminary Invalidity Contentions are not an admission by Fortinet that the accused products, including any current or past versions of the accused products, are covered by, or infringe the asserted claims, but are based instead on the recognition that if the claims are interpreted to be broad enough to encompass the accused products, the claims must also be construed to have that same scope when considering whether they are invalid.

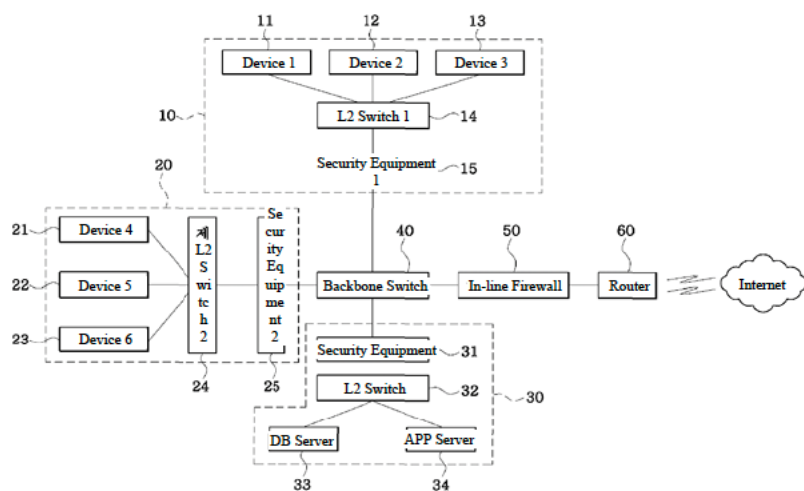
The following chart is partially based on, but is not limited by, the claim constructions implicit in Plaintiff's Infringement Contentions, to the extent that such constructions are apparent from the Infringement Contentions. Fortinet notes that in many instances, Plaintiff's Infringement Contentions fail to provide adequate notice of Plaintiff's construction of the asserted claims and fail to comply with the Court's scheduling order and other applicable rules. Fortinet does not accept the assumptions concerning the scope and meaning implicit in Plaintiff's Infringement Contentions, to the extent those assumptions are discernible, and reserves the right to challenge Plaintiff's proposed (or implied) constructions. Fortinet also reserves the right to revise and supplement these charts if and when Plaintiff is permitted to provide revised Infringement Contentions or otherwise make its positions known. To the extent that these Preliminary Invalidity Contentions rely on or otherwise embody particular constructions of terms or phrases in the asserted claims, Fortinet does not necessarily advocate any such construction as proper constructions of those terms or phrase. Fortinet also reserves the right to revise and supplement these charts after the Court construes the claims. Citations given in the chart below are merely representative of the respective elements and are not meant to be exhaustive.

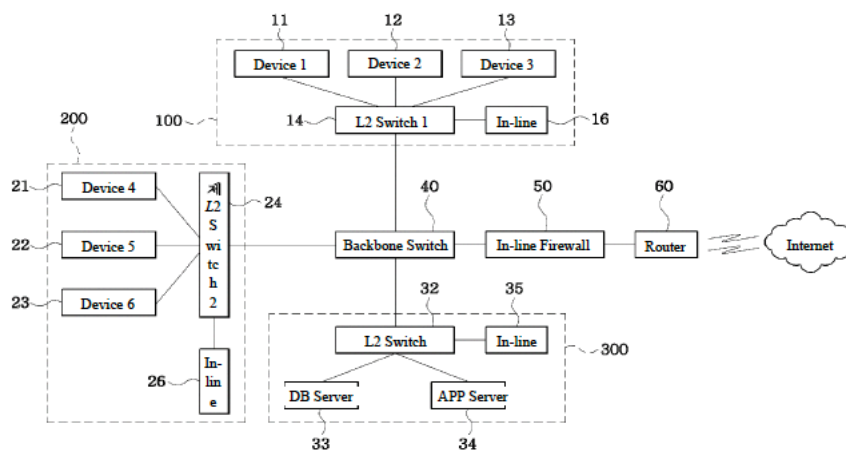
Claim	Exemplary Citation from Lee
<p>[1PRE] A method of network based malware detection in a service provider network, the method comprising:</p>	<p>Lee discloses and/or renders obvious a method of network based malware detection in a service provider network with the following steps.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>The diagram illustrates a network architecture. At the top, a dashed box labeled 10 contains three devices (11, 12, 13) connected to an L2 Switch (14). Below this switch is Security Equipment 1. To the left, another dashed box labeled 20 contains three devices (21, 22, 23) connected to an L2 Switch (24). Below this switch is Security Equipment 2. In the center, a Backbone Switch (40) is connected to Security Equipment 1 and Security Equipment 31. To the right of the backbone switch is an In-line Firewall (50), which is connected to a Router (60). The router is connected to the Internet (represented by a cloud). Below the backbone switch, a dashed box labeled 30 contains an L2 Switch (32) connected to a DB Server (33) and an APP Server (34). Security Equipment 31 is also connected to the L2 Switch (32).</p> <p><i>Drawings</i> Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p> <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
<p>[1A] receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;</p>	<p>Lee discloses and/or renders obvious receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1</p>

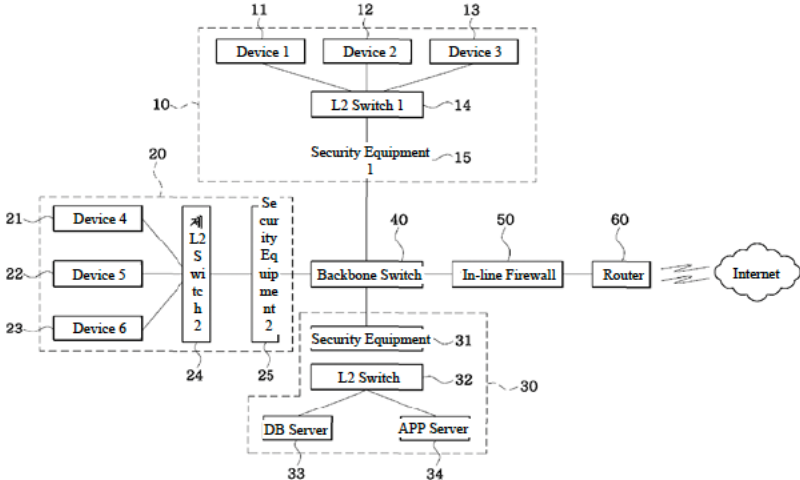
Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p> <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

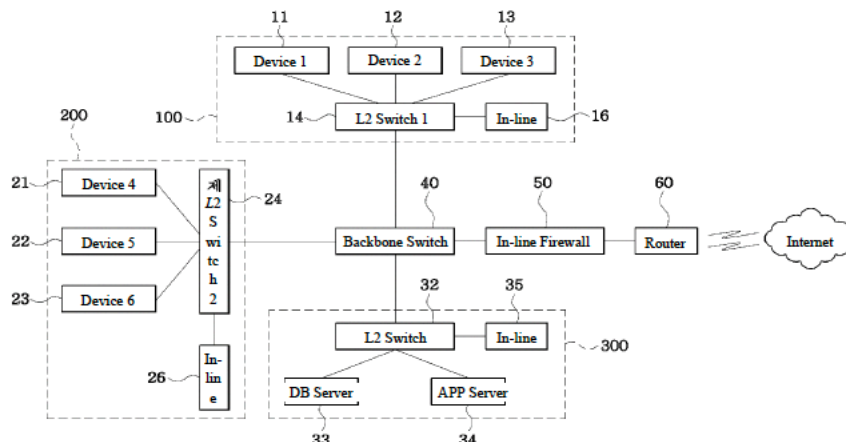
Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14:</p> <p>"The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p>
<p>[1B] determining an operating system identifier (OS ID) associated with the TCP session and the computing device;</p>	<p>Lee discloses and/or renders obvious determining an operating system identifier (OS ID) associated with the TCP session and the computing device.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

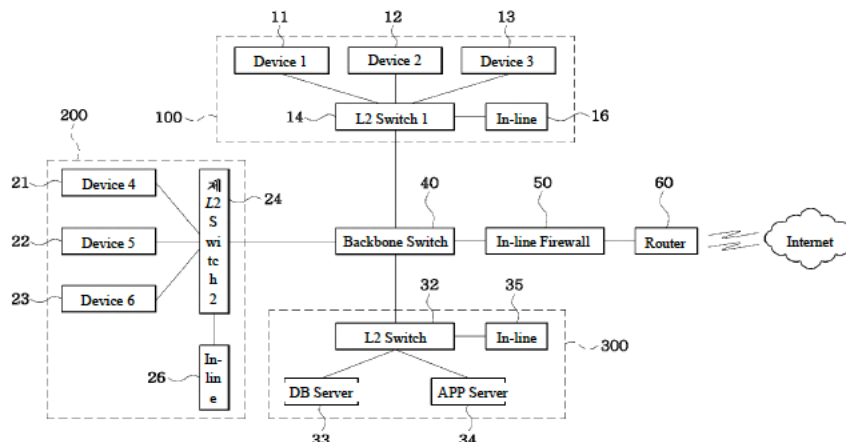
Claim	Exemplary Citation from Lee
<p>[1C] determining if malware is present in the TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets; and</p>	<p>Lee discloses and/or renders obvious determining if malware is present in the TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1</p> <p>Lee at Fig. 2 and corresponding text:</p>

Claim	Exemplary Citation from Lee
	 <p>The diagram illustrates a network architecture. At the top, three devices (11, 12, 13) are connected to L2 Switch 1 (14). L2 Switch 1 is connected to an In-line device (16). Below this, a Backbone Switch (40) is connected to L2 Switch 1. The Backbone Switch is also connected to an In-line Firewall (50) and a Router (60). The Router is connected to the Internet. On the left, three devices (21, 22, 23) are connected to an L2 Switch (24). The L2 Switch is connected to an In-line device (26). Below the Backbone Switch, there is another L2 Switch (32) connected to a DB Server (33) and an APP Server (34). An In-line device (35) is also connected to the L2 Switch (32). A dashed box labeled 300 encloses the L2 Switch (32), the DB Server (33), the APP Server (34), and the In-line device (35). Another dashed box labeled 200 encloses the L2 Switch (24) and the In-line device (26). A dashed box labeled 100 encloses the top section of the network, including devices 11, 12, 13, L2 Switch 1, and In-line device 16.</p> <p>Lee at Abstract: "This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

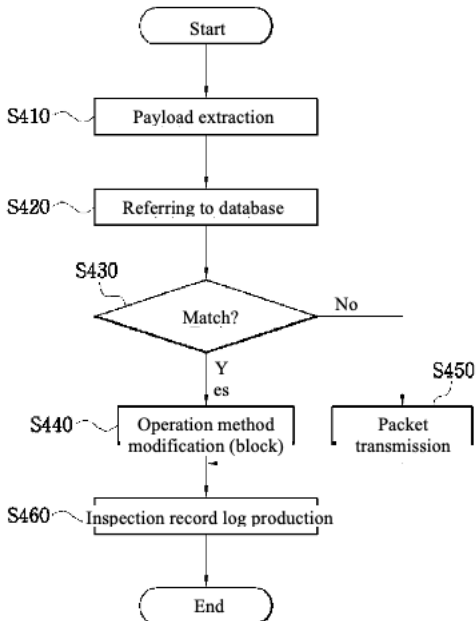
Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[1D] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.</p>	<p>Lee discloses and/or renders obvious generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.</p> <p>Lee at Fig. 1 and corresponding text:</p>

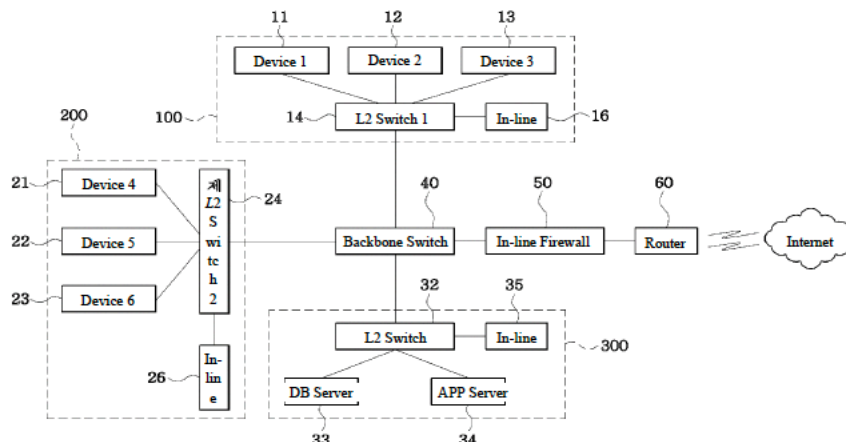
Claim	Exemplary Citation from Lee
	<div data-bbox="651 251 1449 738"> <pre> graph LR subgraph 10 [Security Equipment 1] D1[Device 1] --- L2S1[L2 Switch 1] D2[Device 2] --- L2S1 D3[Device 3] --- L2S1 L2S1 --- SE1[Security Equipment 1] end subgraph 20 [Security Equipment 2] D4[Device 4] --- L2S2[L2 Switch 2] D5[Device 5] --- L2S2 D6[Device 6] --- L2S2 L2S2 --- SE2[Security Equipment 2] L2S2 --- L2S3[L2 Switch] L2S3 --- DBS[DB Server] L2S3 --- APPS[APP Server] end SE1 --- BS[Backbone Switch] SE2 --- BS BS --- IF[In-line Firewall] IF --- R[Router] R --- Internet((Internet)) </pre> </div> <p data-bbox="598 738 651 820" style="transform: rotate(-90deg); transform-origin: left top;"> <i>Drawings</i> <i>Fig. 1</i> </p> <p data-bbox="598 852 1071 885">Lee at Fig. 2 and corresponding text:</p>

Claim	Exemplary Citation from Lee
	 <p>The diagram illustrates a network architecture. At the top, three devices (11, 12, 13) are connected to L2 Switch 1 (14). An In-line device (16) is connected to L2 Switch 1. L2 Switch 1 is connected to a Backbone Switch (40). The Backbone Switch is connected to an In-line Firewall (50), which is connected to a Router (60). The Router is connected to the Internet. On the left, three devices (21, 22, 23) are connected to L2 Switch 2 (24). An In-line device (26) is connected to L2 Switch 2. L2 Switch 2 is connected to the Backbone Switch. At the bottom, L2 Switch 3 (32) is connected to the Backbone Switch. L2 Switch 3 is connected to a DB Server (33) and an APP Server (34). An In-line device (35) is connected to L2 Switch 3. The In-line Firewall (50) is also connected to L2 Switch 3.</p> <p>Lee at Abstract: "This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p>  <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[2PRE] The method of claim 1 wherein determining the operating system ID further comprises:</p>	<p>See claim 1.</p>
<p>[2A] determining a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first</p>	<p>Lee discloses and/or renders obvious determining a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first operating system identifier (ID).</p> <p>Lee at Fig. 1 and corresponding text:</p>

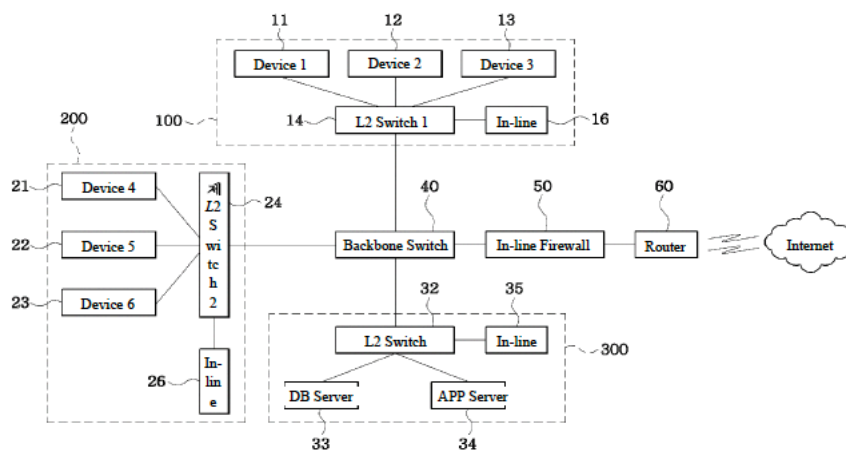
Claim	Exemplary Citation from Lee
operating system identifier (ID);	<div data-bbox="651 243 1449 730"> </div> <p data-bbox="598 738 651 820" style="transform: rotate(-90deg); transform-origin: left top;"> <i>Drawings</i> <i>Fig. 1</i> </p> <p data-bbox="598 852 1071 885">Lee at Fig. 2 and corresponding text:</p>

Claim	Exemplary Citation from Lee
	 <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

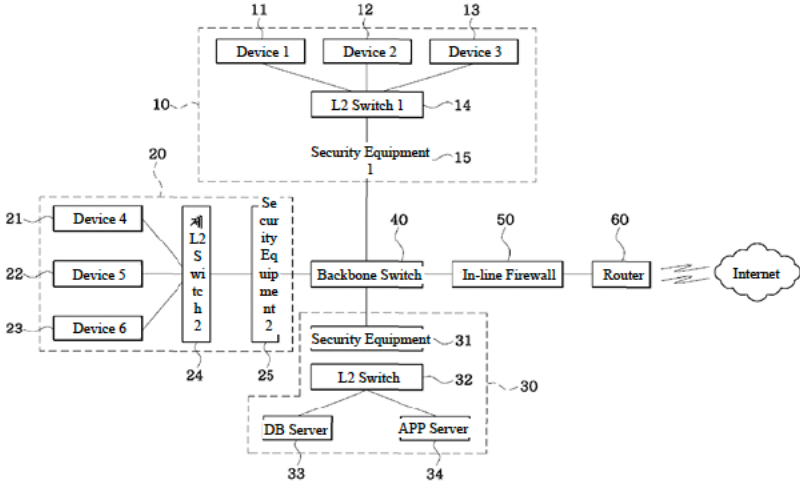
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

Claim	Exemplary Citation from Lee
<p>[2B] determining an application parameter associated with the TCP session if present, and identifying a second OS ID from the application parameter; and</p>	<p>Lee discloses and/or renders obvious determining an application parameter associated with the TCP session if present, and identifying a second OS ID from the application parameter.</p> <p>Lee at Fig. 1 and corresponding text:</p> <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

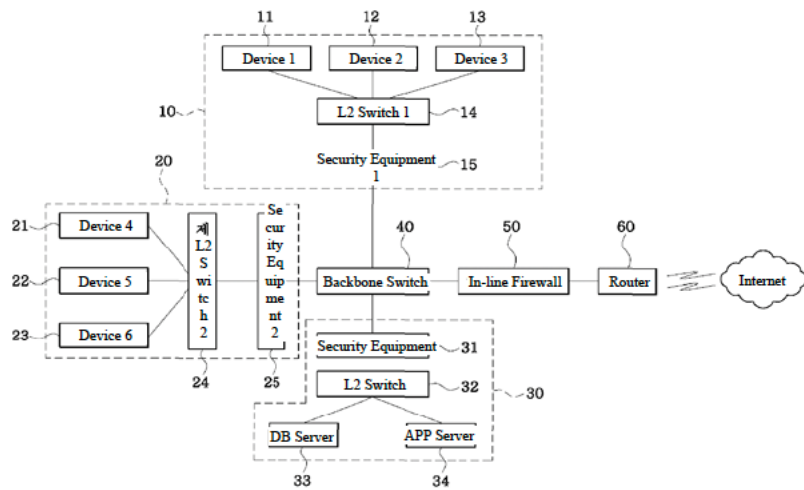
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

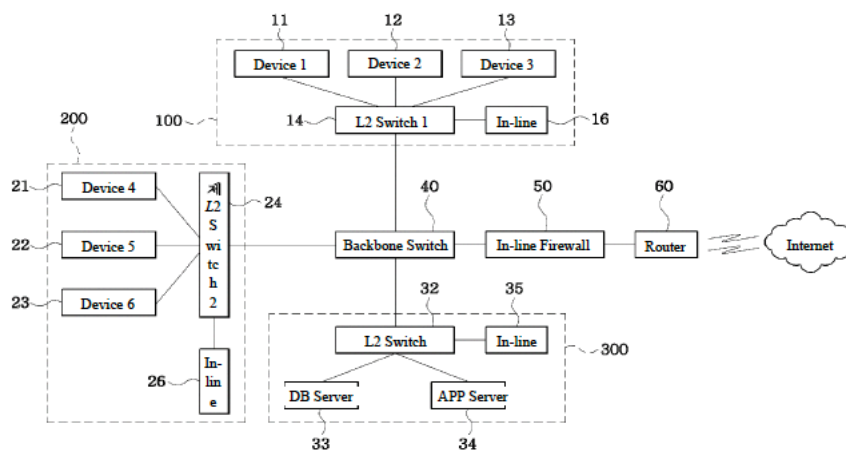
Claim	Exemplary Citation from Lee
<p>[2C] wherein the OS ID in the alert comprises one of the first OS ID or second OS ID associated with one or more computing devices coupled to the access device.</p>	<p>Lee discloses and/or renders obvious that the OS ID in the alert comprises one of the first OS ID or second OS ID associated with one or more computing devices coupled to the access device.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p> <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

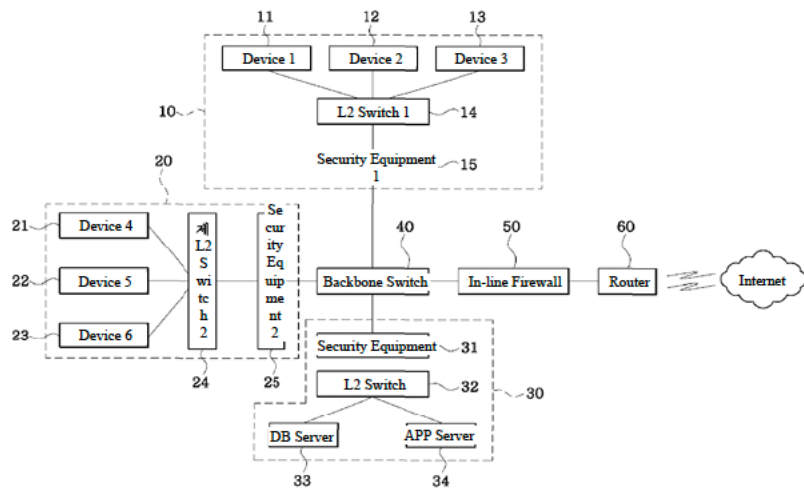
Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[3] The method of claim 2 further comprising sending a notification to a subscriber associated with the network address associated with the access device, the notification identifying a remediation portal to remove the determined malware in the alert.</p>	<p>Lee discloses and/or renders obvious sending a notification to a subscriber associated with the network address associated with the access device, the notification identifying a remediation portal to remove the determined malware in the alert.</p>

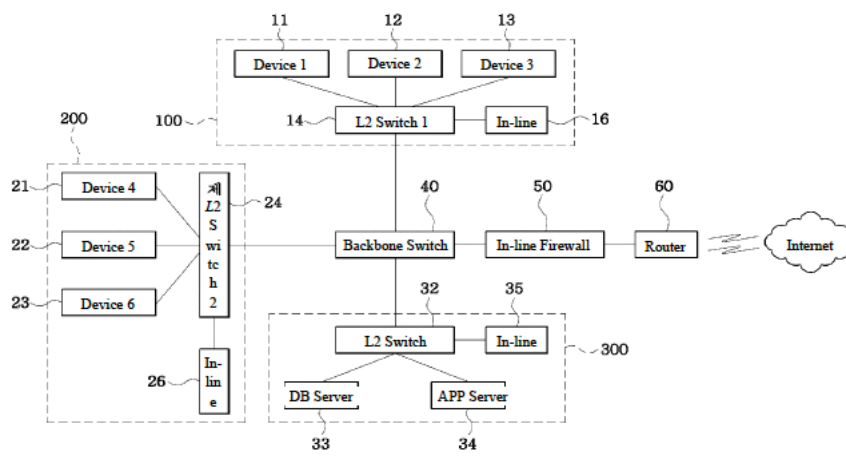
Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 272">Lee at Fig. 1 and corresponding text:</p>  <p data-bbox="598 787 651 852"><i>Drawings</i> <i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

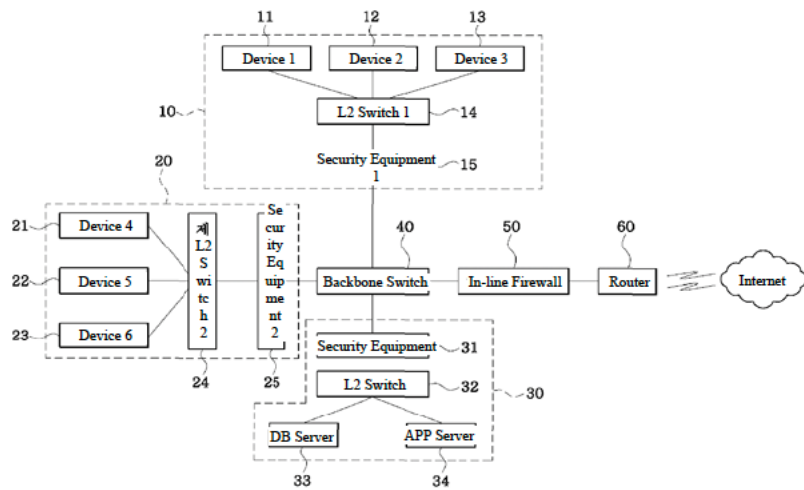
Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[4PRE] The method of claim 3 further comprising:</p>	<p>See Claim 3.</p>
<p>[4A] identifying at the remediation portal the subscriber accessing the remediation portal;</p>	<p>Lee discloses and/or renders obvious identifying at the remediation portal the subscriber accessing the remediation portal.</p>

Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 272">Lee at Fig. 1 and corresponding text:</p>  <p data-bbox="598 787 651 852"><i>Drawings</i> <i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

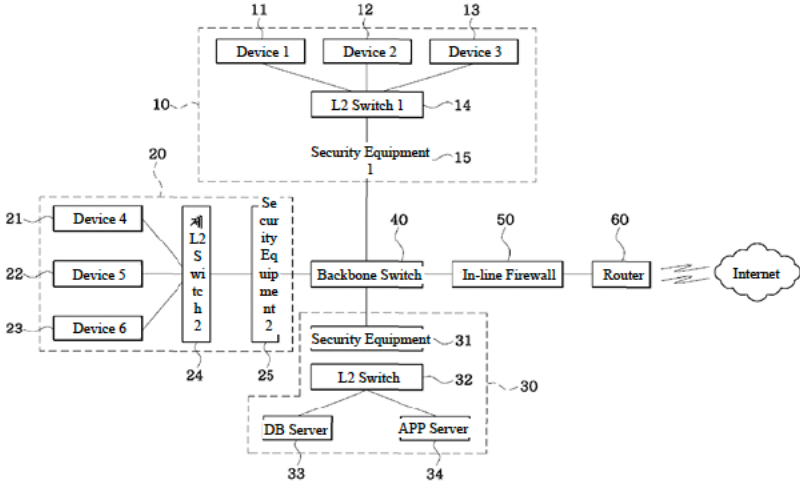
Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[4B] identifying the OS associated with the computing device used by the subscriber to access the remediation portal;</p>	<p>Lee discloses and/or renders obvious identifying the OS associated with the computing device used by the subscriber to access the remediation portal.</p>

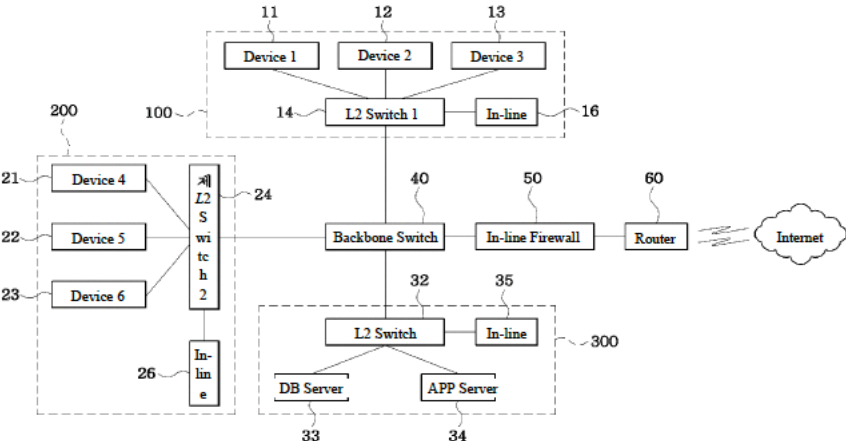
Claim	Exemplary Citation from Lee
	<p data-bbox="598 238 1071 267">Lee at Fig. 1 and corresponding text:</p>  <p data-bbox="598 776 651 852"><i>Drawings</i> <i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p> <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

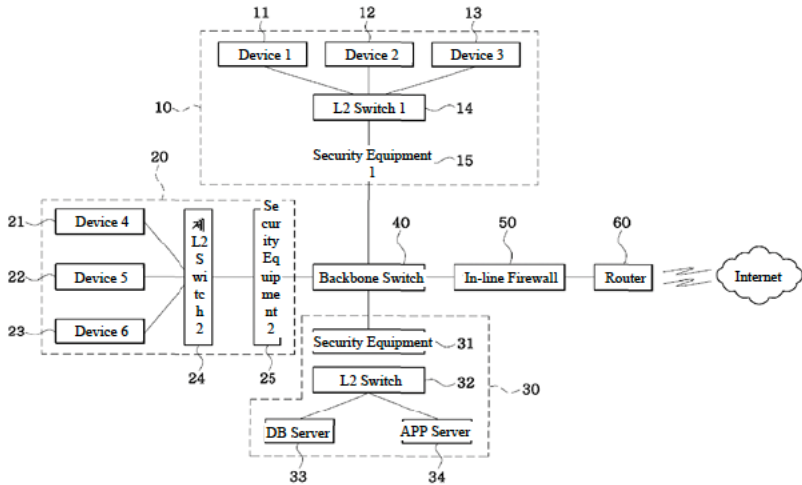
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

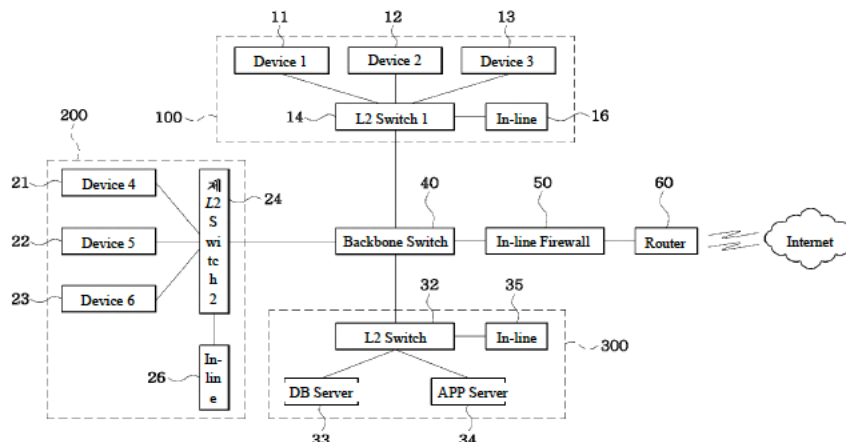
Claim	Exemplary Citation from Lee
<p>[4C] comparing the OS associated with the computing device to the at least one of the first OS ID or second OS ID identified in the alert; and</p>	<p>Lee discloses and/or renders obvious comparing the OS associated with the computing device to the at least one of the first OS ID or second OS ID identified in the alert.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 267">Lee at Fig. 2 and corresponding text:</p>  <p data-bbox="598 889 808 917">Lee at Abstract:</p> <p data-bbox="598 925 1879 1429">"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

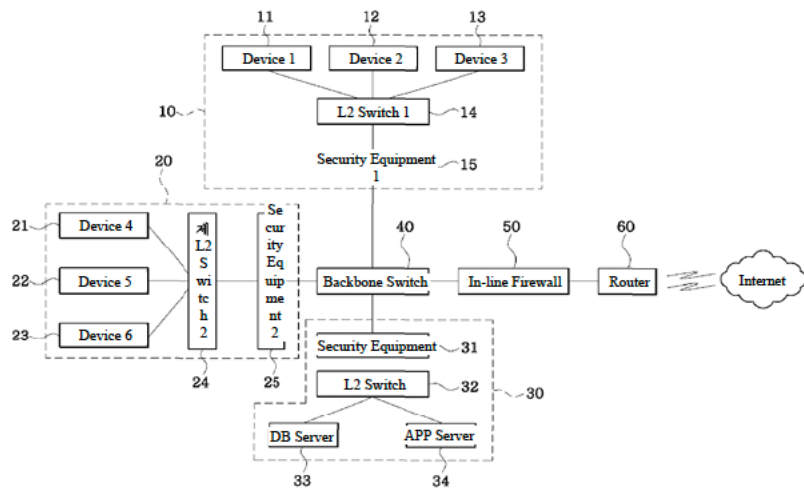
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

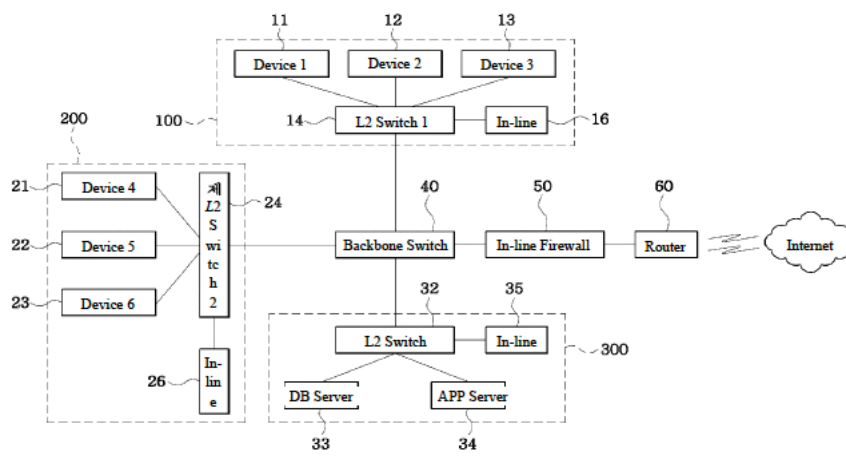
Claim	Exemplary Citation from Lee
<p>[4D] providing malware remediation to the computing device when the at least one of the first OS ID or second OS ID matches the OS associated with the computing device.</p>	<p>Lee discloses and/or renders obvious providing malware remediation to the computing device when the at least one of the first OS ID or second OS ID matches the OS associated with the computing device.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1 is a network diagram illustrating a system architecture. It shows two main sections, 10 and 20, connected to a central Backbone Switch (40). Section 10 includes Device 1 (11), Device 2 (12), and Device 3 (13) connected to L2 Switch 1 (14), which is connected to Security Equipment 1 (15). Section 20 includes Device 4 (21), Device 5 (22), and Device 6 (23) connected to L2 Switch 2 (24), which is connected to Security Equipment 2 (25). The Backbone Switch (40) is also connected to Security Equipment 3 (31) and L2 Switch 3 (32). L2 Switch 3 (32) is connected to DB Server (33) and APP Server (34). The Backbone Switch (40) is connected to an In-line Firewall (50), which is connected to a Router (60), which in turn connects to the Internet.</p> <p>Lee at Fig. 2 and corresponding text:</p>

Claim	Exemplary Citation from Lee
	 <p>The diagram illustrates a network architecture. At the top, three devices (11, 12, 13) are connected to L2 Switch 1 (14). L2 Switch 1 is connected to a Backbone Switch (40). To the left, three devices (21, 22, 23) are connected to L2 Switch 2 (24). L2 Switch 2 is connected to the Backbone Switch (40). Below the Backbone Switch, there is an L2 Switch (32) connected to a DB Server (33) and an APP Server (34). To the right of the Backbone Switch, there is an In-line Firewall (50) connected to a Router (60), which is connected to the Internet. An In-line device (26) is connected to L2 Switch 2. A dashed box labeled 100 encloses L2 Switch 1 and In-line device 16. Another dashed box labeled 300 encloses L2 Switch 32 and In-line device 35. A dashed box labeled 200 encloses L2 Switch 24 and In-line device 26.</p> <p>Lee at Abstract: "This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

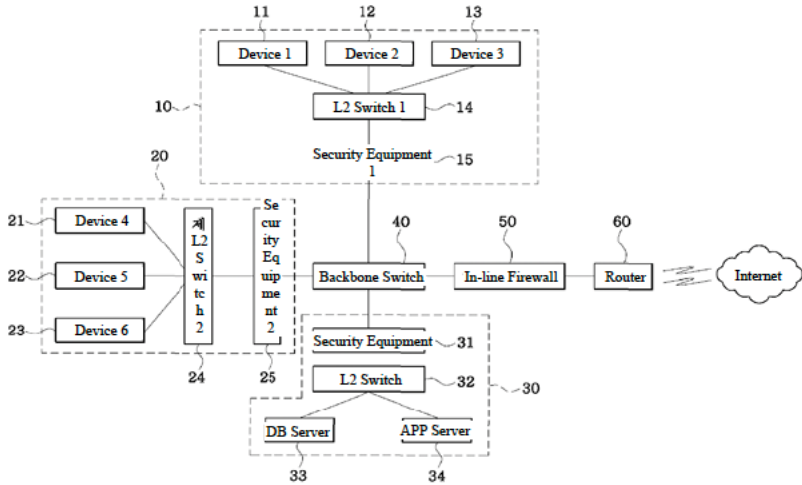
Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[5] The method of claim 3 wherein if the OS of the computing device is not the at least one of the first OS ID or second OS ID identified in the alert an indication is provided to the subscriber to connect to the remediation portal with an other computing device connected to the access device.</p>	<p>Lee discloses and/or renders obvious that, if the OS of the computing device is not the at least one of the first OS ID or second OS ID identified in the alert an indication is provided to the subscriber to connect to the remediation portal with an other computing device connected to the access device.</p>

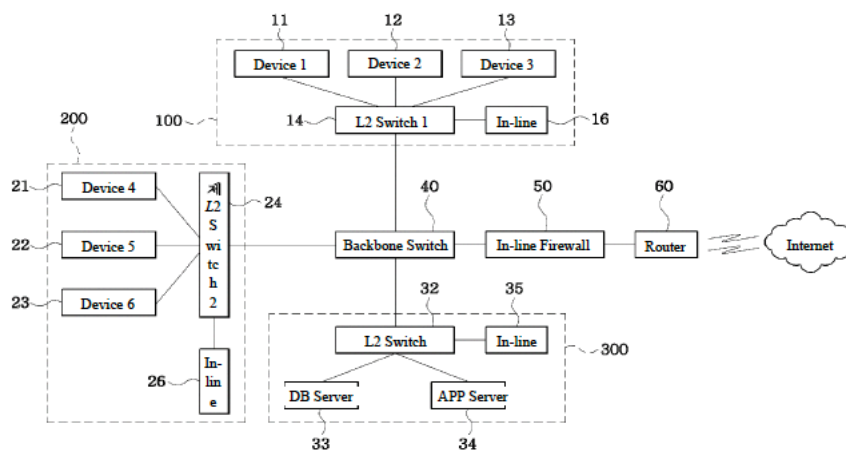
Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 272">Lee at Fig. 1 and corresponding text:</p>  <p data-bbox="598 787 651 852"><i>Drawings</i> <i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[6] The method of claim 3 wherein the alert is cleared when all computing devices associated with the access device and the computing device having the at least one of the first OS ID or second OS ID have accessed the remediation portal.</p>	<p>Lee discloses and/or renders obvious that the alert is cleared when all computing devices associated with the access device and the computing device having the at least one of the first OS ID or second OS ID have accessed the remediation portal.</p>

Claim	Exemplary Citation from Lee
	<p data-bbox="598 238 1071 267">Lee at Fig. 1 and corresponding text:</p>  <p data-bbox="598 787 651 852"><i>Drawings</i> <i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a network architecture. At the top, a dashed box 100 contains Device 1 (11), Device 2 (12), and Device 3 (13), which are connected to L2 Switch 1 (14). An In-line device (16) is also connected to L2 Switch 1. Below this, a dashed box 200 contains Device 4 (21), Device 5 (22), and Device 6 (23), connected to another L2 Switch (24). An In-line device (26) is connected to this switch. A central Backbone Switch (40) is connected to L2 Switch 1 (14), L2 Switch (24), and L2 Switch (32). L2 Switch (32) is connected to an In-line device (35) and a DB Server (33). An APP Server (34) is also connected to L2 Switch (32). An In-line Firewall (50) is connected to the Backbone Switch (40) and a Router (60). The Router (60) is connected to the Internet. A dashed box 300 encloses the L2 Switch (32), In-line device (35), DB Server (33), and APP Server (34).</p> <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

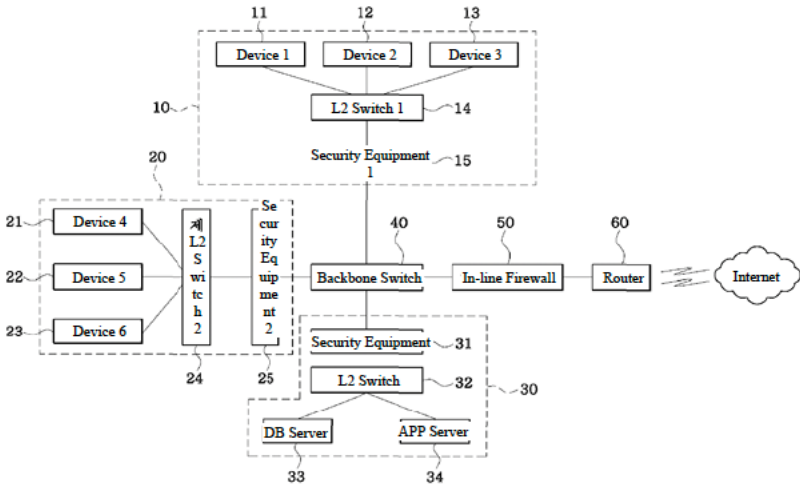
Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[7] The method of claim 2 wherein determining the first OS ID is performed by comparing one or more protocol parameters of the one or more TCP packets having a SYN flag associated with the session, wherein the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS</p>	<p>Lee discloses and/or renders obvious that determining the first OS ID is performed by comparing one or more protocol parameters of the one or more TCP packets having a SYN flag associated with the session, wherein the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS ID associated with parameters.</p>

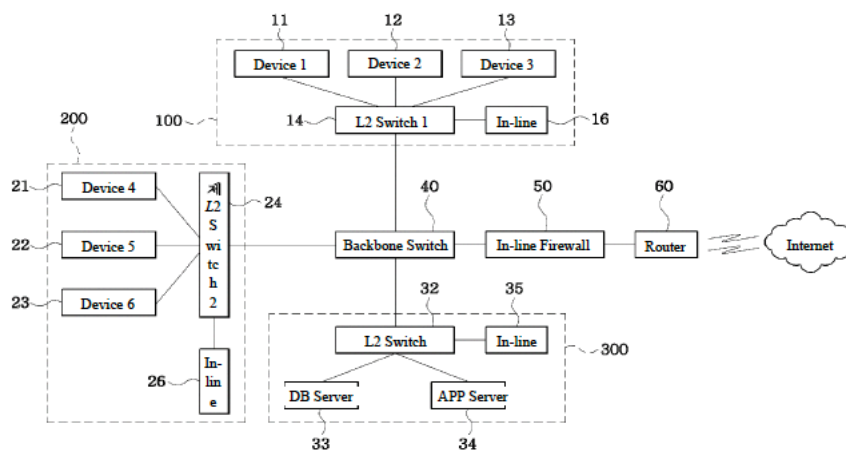
Claim	Exemplary Citation from Lee
<p>ID associated with parameters.</p>	<p>Lee at Fig. 1 and corresponding text:</p> <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p> <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

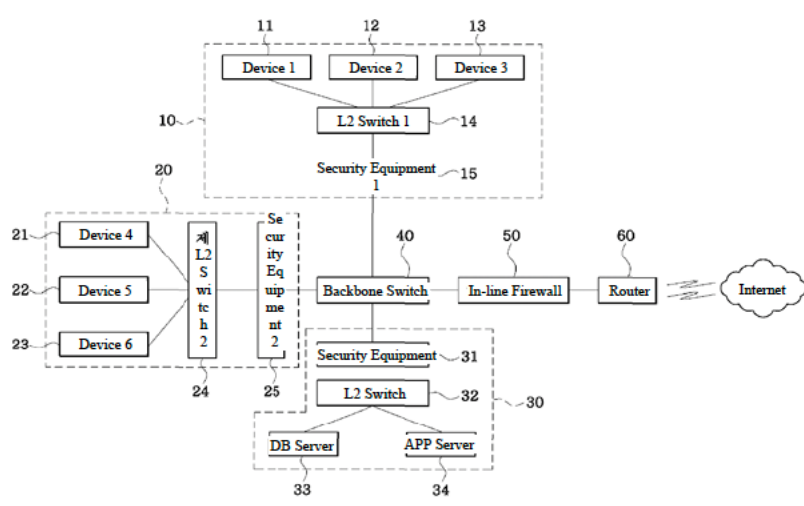
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

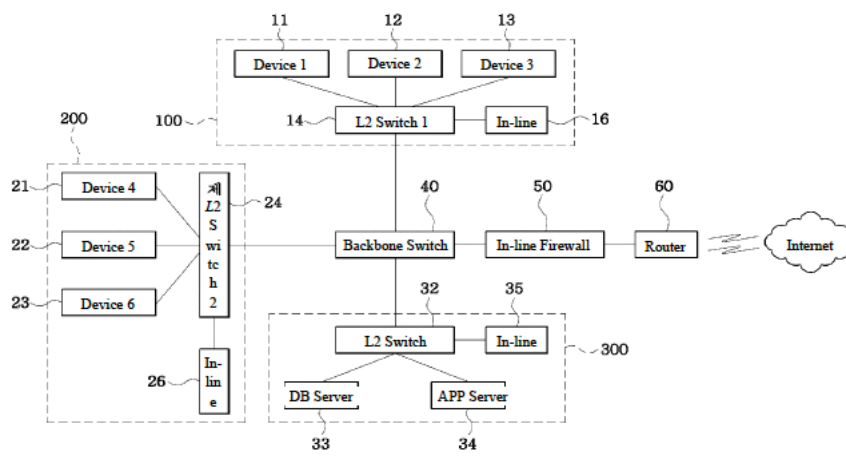
Claim	Exemplary Citation from Lee
<p>[8] The method of claim 2 wherein determining the application parameter associated with the session and identifying the second OS ID from the application parameter by inspecting an HTTP user agent string to determine the second OS ID.</p>	<p>Lee discloses and/or renders obvious that determining the application parameter associated with the session and identifying the second OS ID from the application parameter by inspecting an HTTP user agent string to determine the second OS ID.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

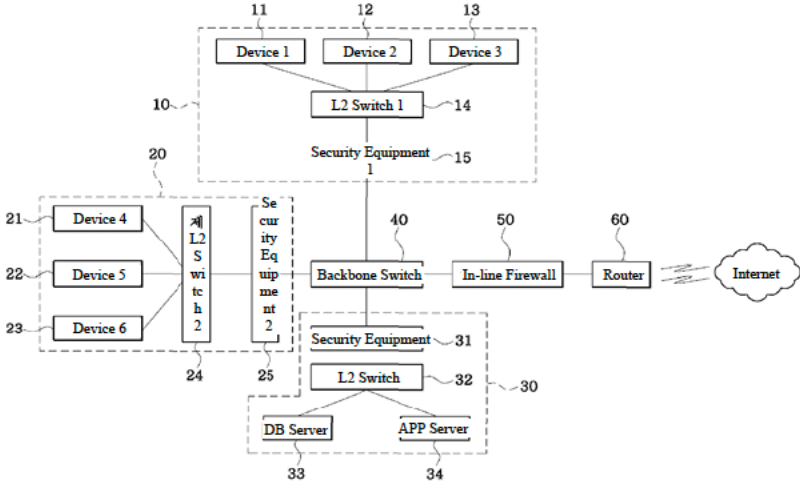
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

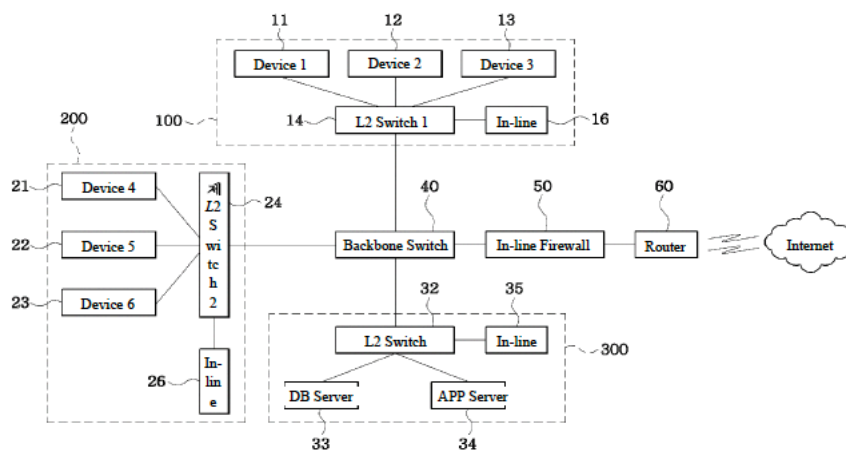
Claim	Exemplary Citation from Lee
<p>[9] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the session is identified by the OS ID identified as a more reliable OS ID.</p>	<p>Lee discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the session is identified by the OS ID identified as a more reliable OS ID.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>The diagram, labeled Fig. 1, illustrates a network architecture. It features two main network segments, 10 and 20, connected by a central Backbone Switch (40). Segment 10, enclosed in a dashed box, contains Device 1 (11), Device 2 (12), and Device 3 (13) connected to L2 Switch 1 (14), which is in turn connected to Security Equipment 15. Segment 20, also in a dashed box, contains Device 4 (21), Device 5 (22), and Device 6 (23) connected to L2 Switch 2 (24), which is connected to Security Equipment 25. The Backbone Switch (40) is connected to an In-line Firewall (50), which then connects to a Router (60). The Router (60) is connected to the Internet, represented by a cloud icon. Additionally, there is a third network segment, 30, at the bottom, containing a DB Server (33) and an APP Server (34) connected to L2 Switch 32 (32), which is connected to Security Equipment 31. The Backbone Switch (40) is also connected to this L2 Switch 32 (32).</p> <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

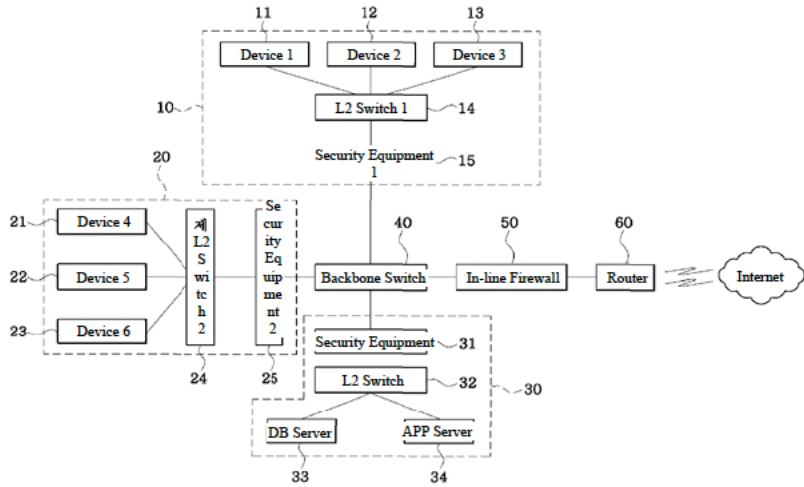
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

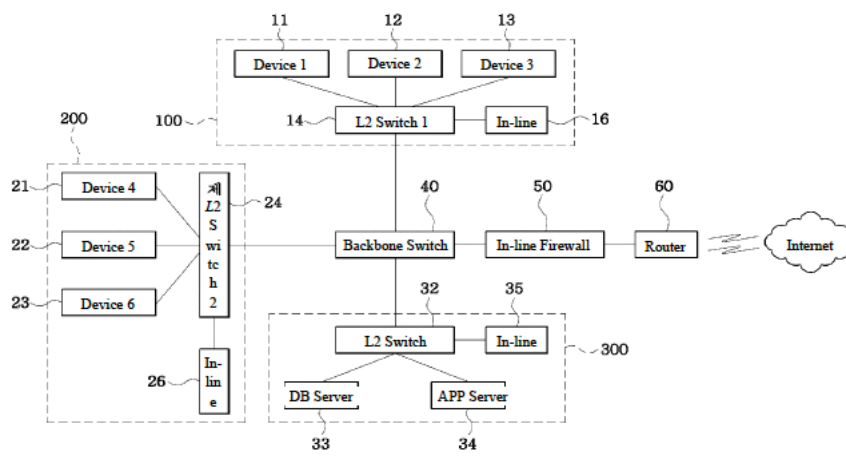
Claim	Exemplary Citation from Lee
<p>[10] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID either OS ID is used.</p>	<p>Lee discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID either OS ID is used.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p style="text-align: center;"><i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

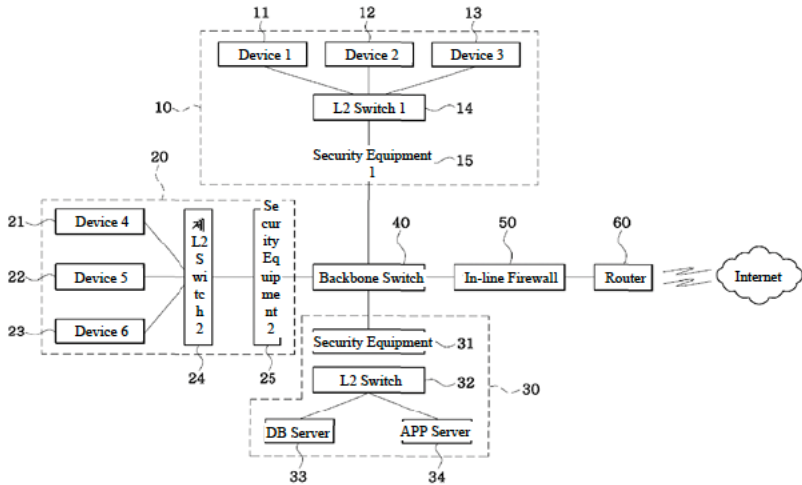
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td><u>00:13:D4:04:BD:C0</u></td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td><u>00:00:FB:09:0F:17</u></td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	<u>00:13:D4:04:BD:C0</u>	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	<u>00:00:FB:09:0F:17</u>	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
<u>00:13:D4:04:BD:C0</u>	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
<u>00:00:FB:09:0F:17</u>	192.168.1.3	Windows XP		ANYTIME	Separation																				

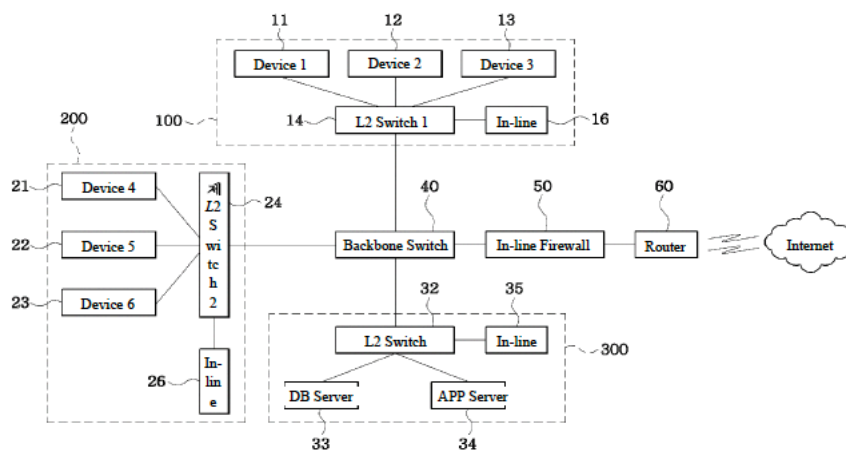
Claim	Exemplary Citation from Lee
<p>[11] The method of claim 2 further comprising verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID but the second OS ID provides additional information, the second OS ID is used.</p>	<p>Lee discloses and/or renders obvious verifying the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID but the second OS ID provides additional information, the second OS ID is used.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 267">Lee at Fig. 2 and corresponding text:</p>  <p data-bbox="598 885 798 912">Lee at Abstract:</p> <p data-bbox="598 922 1879 1425">"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 30:</p> <p>"In an in-line network mechanism consisting of a network mechanism achieved by inclusion of multiple internal networks encompassing multiple devices and an L2 switch (hub) connected to the aforesaid devices, an L2 switch (hub) connected to a DB (database) server and APP (application) server, a backbone switch connected to the aforesaid hubs, in-line firewall connected to the aforesaid backbone switch, a router connected to the aforesaid in-line firewall and the internet and construction of an in-line system on each one of the aforesaid internal network's hub and L2 switch (hub) connected to the DB server and APP server, the invention designed to achieve aforesaid objectives is characterized by a virtual in-line network security method comprising of: A 1st process that determines whether a particular equipment is subject to virtual path routing (S100), which consists of the step of monitoring MAC request packet generated in the aforesaid device (node) (S110), the step of extracting the origin address and MAC address of the aforesaid MAC request packet (S120), the step of adding the aforesaid origin address and MAC address of the aforesaid extracted MAC request packet to the node table (TABLE) that can store and check MAC address, IP address, OS (operating system), user, time of use, and operation method (S130) and the step of estimating (guessing) the operating system (OS) of the aforesaid extracted node (S140); a 2nd process that routes the virtual path of applicable equipment (S200) when a particular equipment is considered suitable for virtual path routing through the aforesaid 1st process (S100), which consists of the step of configuring the method of operation of the node of the aforesaid node table (S210), a step of linking security policy to the node of the node table (S220) and the step of responding with MAC address of the virtual path system to all MAC address requests from the "virtual path" node configured through the aforesaid method of operation (S230); a 3rd process that inspects whether the traffic of the aforesaid virtual path routed is in conformity with the security policy (S300), which consists of the step of extracting the protocol, destination address, destination port number, etc. from packets entering into the aforesaid virtual path system (S310) and the step of determining whether to respond by referring to the security policy table that can store and verify the destination address, destination port, time zone, and execution (S320); a 4th process that inspects whether the traffic of the aforesaid virtual path is a harmful traffic (S400), which consists of the step of extracting a packet's payload (S410), the step of referring to a harmful/malicious database (DB) (S420), the step of determining whether the aforesaid packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation on the aforesaid node table (S440) if the aforesaid packet's payload matches the harmful/malicious database (DB) in the aforesaid determination step (S430), the</p>

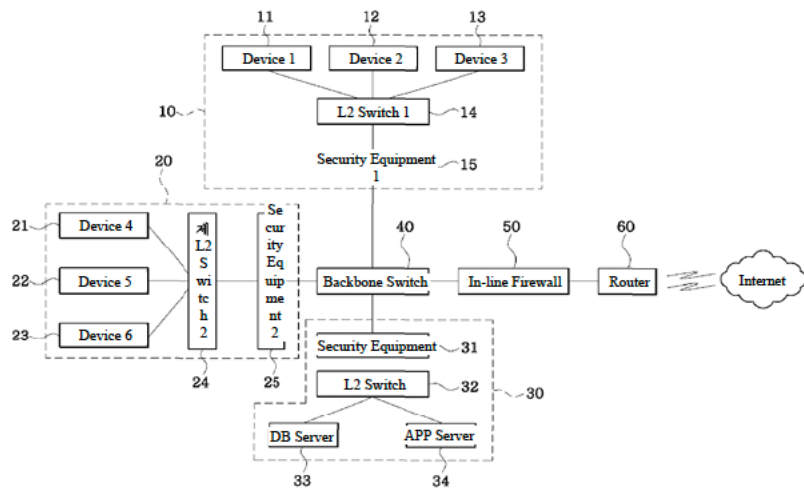
Claim	Exemplary Citation from Lee																								
	<p>step of transmitting the packet (S450) if the aforesaid packet's payload does not match the harmful/malicious database (DB) in the aforesaid determination step (S430) and the step of producing an inspection record log of the action (S460); and, a 5th process that separates the aforesaid virtual path's system from the network (S500), which consists of the step of classifying malicious node by referring to the method of operation (mode) of the aforesaid node table (S510), the step of periodically transmitting false MAC address response packets to the aforesaid malicious node (S520) and the step of sending packets to false MAC addresses in response to all MAC address requests from the aforesaid malicious node (S530)."</p> <p>Lee at ¶ 48:</p> <p>"As shown, the aforesaid 1st process (S100) consists of the step of monitoring MAC request packet generated in a device (hereinafter referred to as "node") (S110), the step of extracting the origin address and MAC address of the MAC request packet (S120), the step of adding the origin address and MAC address of the extracted MAC request packet to the node (device) table that can store and verify MAC address, IP address, OS (operating system), user, usage time and the method of operation (mode), etc. (S130), and the step of estimating (Guessing) the operating system (OS) of the extracted node (S140)."</p> <p>Lee at ¶ 55:</p> <p>"In addition, the origin address and MAC address of the extracted MAC request packet are added to the node table (TABLE) and it becomes possible to recognize the operating system (OS) of the aforesaid extracted node."</p> <p>Lee at Table 1 and corresponding text:</p> <p><Table 1></p> <p>[Node Table]</p> <table><tr><th>MAC address</th><th>IP address</th><th>OS</th><th>User</th><th>Time</th><th>Operation method</th></tr><tr><td>00:13:D4:04:BD:C0</td><td>192.168.1.2</td><td>Linux</td><td></td><td></td><td>Allow</td></tr><tr><td>00:0E:EA:6D:71:2A</td><td>192.168.1.1</td><td>Windows XP</td><td>Kimpd</td><td></td><td>Virtual path</td></tr><tr><td>00:00:FB:09:0F:17</td><td>192.168.1.3</td><td>Windows XP</td><td></td><td>ANYTIME</td><td>Separation</td></tr></table>	MAC address	IP address	OS	User	Time	Operation method	00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow	00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path	00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation
MAC address	IP address	OS	User	Time	Operation method																				
00:13:D4:04:BD:C0	192.168.1.2	Linux			Allow																				
00:0E:EA:6D:71:2A	192.168.1.1	Windows XP	Kimpd		Virtual path																				
00:00:FB:09:0F:17	192.168.1.3	Windows XP		ANYTIME	Separation																				

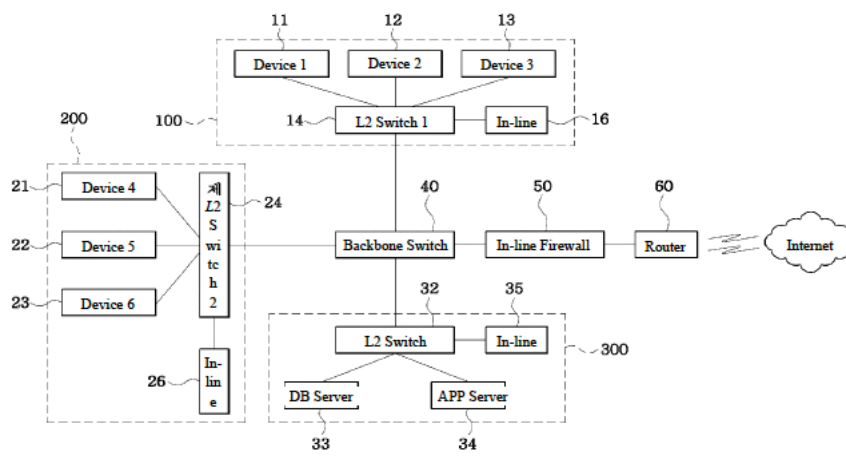
Claim	Exemplary Citation from Lee
<p>[12] The method of claim 3 wherein sending the notification to the subscriber associated with the access device further comprises replacing the network address of the access device with a subscriber ID.</p>	<p>Lee discloses and/or renders obvious that sending the notification to the subscriber associated with the access device further comprises replacing the network address of the access device with a subscriber ID.</p> <p>Lee at Fig. 1 and corresponding text:</p>  <p>Fig. 1</p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

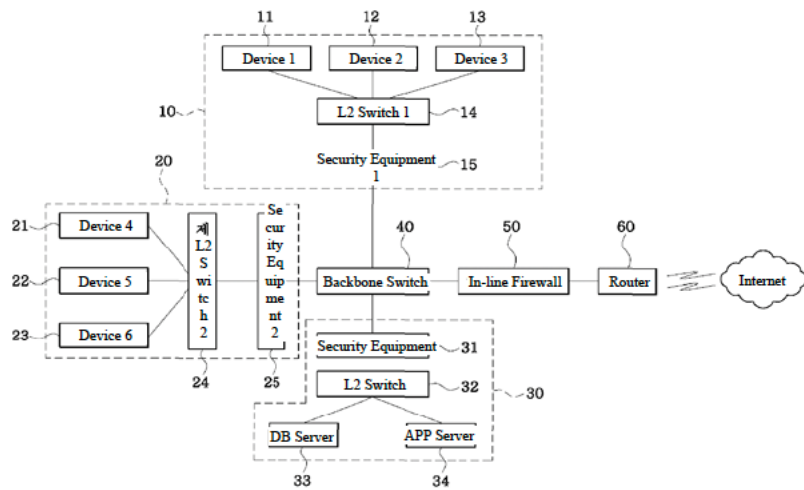
Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[13] The method of claim 12 wherein sending the notification to the subscriber associated with the access device further comprises aggregating one or more detailed alerts having the subscriber ID into a single alert summary associated with the subscriber ID.</p>	<p>Lee discloses and/or renders obvious that sending the notification to the subscriber associated with the access device further comprises aggregating one or more detailed alerts having the subscriber ID into a single alert summary associated with the subscriber ID.</p>

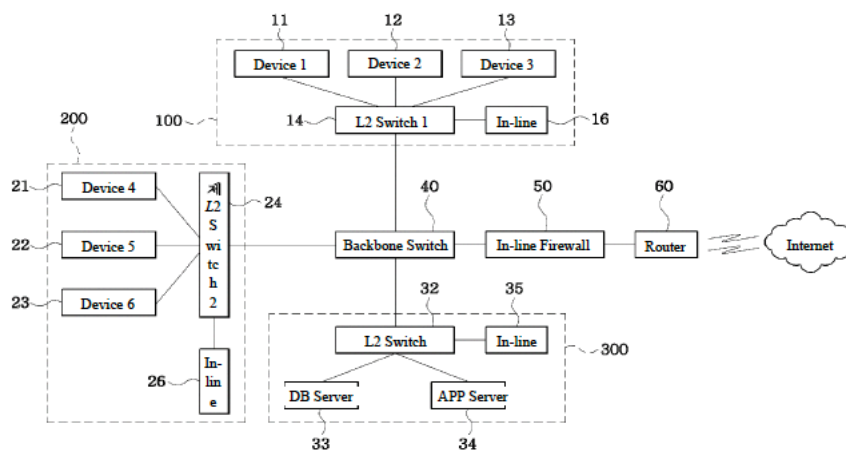
Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 272">Lee at Fig. 1 and corresponding text:</p>  <p data-bbox="598 787 651 852"><i>Drawings</i> <i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>The diagram illustrates a network architecture. At the top, three devices (11, 12, 13) are connected to L2 Switch 1 (14). L2 Switch 1 is connected to a Backbone Switch (40). To the right of the Backbone Switch is an In-line Firewall (50), which is connected to a Router (60). The Router is connected to the Internet. Below the Backbone Switch, there is another L2 Switch (32) connected to a DB Server (33) and an APP Server (34). To the left of the Backbone Switch, there is a vertical stack of three devices (21, 22, 23) connected to an In-line switch (24). The In-line switch is connected to the Backbone Switch. Below the In-line switch is an In-line device (26). The entire system is enclosed in a dashed box labeled 100. A separate dashed box labeled 200 encloses the top three devices and L2 Switch 1. Another dashed box labeled 300 encloses the bottom L2 Switch and its connected servers.</p> <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

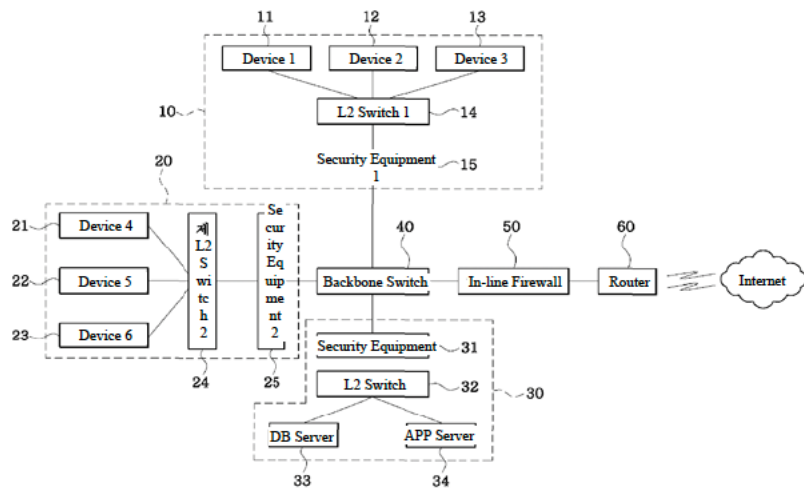
Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14: "The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p> <p>Lee at ¶ 70-73: "As shown, the aforesaid 4th process (S400) consists of the step of extracting a payload of a packet (S410), the step of referring to the harmful/malicious database (DB) (S420), the step of determining whether the packet's payload matches the harmful/malicious database (DB) (S430), the step of modifying (blocking) the method of operation in the node table (S440) if the payload (payload) of the aforesaid packet matches the harmful/malicious data base (DB) in the aforesaid determination step (S430), the step of transmitting the corresponding packet (S450) if the payload of the aforesaid packet does not match the harmful/malicious database (DB) in the aforesaid determination step (S430), and the step of producing the audit record log for the action (S460). To examine the operation, the 4th process (S400), which inspects whether the traffic in the virtual path is harmful traffic extracts the packet's payload, refers to the harmful/malicious database (DB) and determines whether the packet's payload matches the harmful/malicious database (DB). If the aforesaid determination results in a match between the aforesaid packet's payload and the harmful/malicious database (DB), the method of operation is changed (blocked) in the node table. If the payload of the aforesaid packet does not match the harmful/malicious database (DB), the corresponding packet is transmitted. Next, an inspection record log of the action is produced."</p>

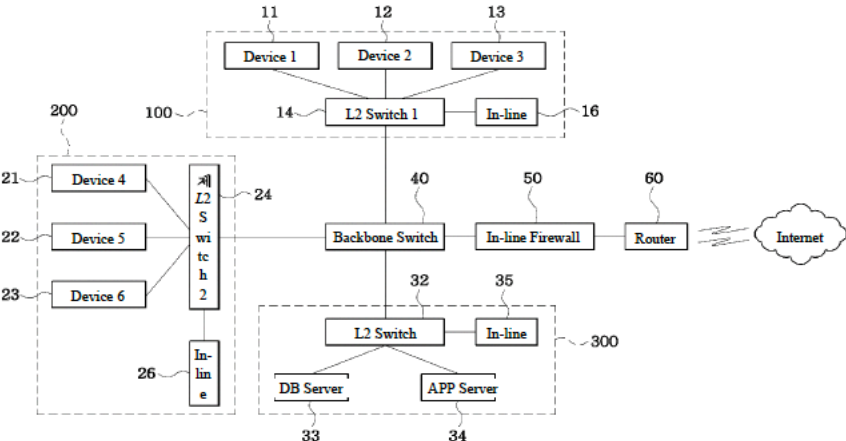
Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 7 and corresponding text:</p> <p><i>Fig. 7</i></p> <pre> graph TD Start([Start]) --> S410[S410 Payload extraction] S410 --> S420[S420 Referring to database] S420 --> S430{S430 Match?} S430 -- No --> S450[S450 Packet transmission] S430 -- Yes --> S440[S440 Operation method modification (block)] S440 --> S460[S460 Inspection record log production] S460 --> End([End]) </pre>
<p>[14] The method of claim 12 wherein the one or more computing devices are coupled to the access device through a local area network (LAN), the access device providing network address translation (NAT) or is coupled to NAT device to share the network address of the access device.</p>	<p>Lee discloses and/or renders obvious that the one or more computing devices are coupled to the access device through a local area network (LAN), the access device providing network address translation (NAT) or is coupled to NAT device to share the network address of the access device.</p>

Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 272">Lee at Fig. 1 and corresponding text:</p>  <p data-bbox="598 787 651 852"><i>Drawings</i> <i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p>Lee at Fig. 2 and corresponding text:</p>  <p>Lee at Abstract:</p> <p>"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14:</p> <p>"The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p>
<p>[15PRE] A system for network based malware detection in a service provider network, the system comprising:</p>	<p><i>See</i> [1PRE].</p>
<p>[15A] a plurality of network sensors coupled to the service provider network, each of the network sensors for:</p>	<p>Lee discloses and/or renders obvious a plurality of network sensors coupled to the service provider network, each of the network sensors.</p>

Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 272">Lee at Fig. 1 and corresponding text:</p>  <p data-bbox="598 787 651 852"><i>Drawings</i> <i>Fig. 1</i></p>

Claim	Exemplary Citation from Lee
	<p data-bbox="598 240 1071 267">Lee at Fig. 2 and corresponding text:</p>  <p data-bbox="598 889 808 917">Lee at Abstract:</p> <p data-bbox="598 925 1879 1421">"This invention is a system that analyzes network packets to determine conformity with security policy and presence/absence of intrusion, and it pertains to a virtual in-line network security method that modulates the responses to MAC address requests of devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined in order to perform the access control. The composition of the invention includes: A 1st process (S100) that determines whether a particular equipment is suitable for virtual path routing; a 2nd process (S200) that routes the virtual path of the applicable equipment when the specific equipment is determined to be suitable for virtual path routing through the aforesaid 1st process (S100); a 3rd process (S300) that inspects whether the traffic of the virtual path routed is in conformity with the security policy; a 4th process (S400) that inspects whether the traffic of the aforesaid virtual path is a harmful traffic; and, a 5th process (S500) that separates the system of the aforesaid virtual path from the network."</p>

Claim	Exemplary Citation from Lee
	<p>Lee at ¶ 13-14:</p> <p>"The invention pertains to a system for analyzing network packets to determine the conformity with security policy and presence/absence of intrusion as in network intrusion blocking systems (firewall) and intrusion prevention systems, and more particularly a security method for virtual in-line network that modulates responses to MAC address requests from devices requiring control among the devices present in a network segment or the same VLAN into a security equipment's MAC address without installing a physical in-line structure in order to cause the communication details of the devices to go through the security equipment, where the conformity with the security policy is determined to perform access control. Network security equipment such as firewalls or intrusion prevention systems like the ones noted can be divided into a router mode or a bridge mode according to the installation configuration."</p>
<p>[15B] receiving one or more transmission control protocol (TCP) packets originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;</p>	<p><i>See</i> [1A].</p>
<p>[15C] determining an operating system identifier (OS ID) associated with the TCP session and the computing device associated;</p>	<p><i>See</i> [1B].</p>
<p>[15D] determining if malware is present in the</p>	<p><i>See</i> [1C].</p>

Claim	Exemplary Citation from Lee
TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets; and	
[15E] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.	<i>See</i> [1D].
[16PRE] The system of claim 15 wherein the plurality of network sensors in determining the OS ID further:	<i>See</i> [2PRE].
[16A] determine a protocol associated with the TCP packets and matching an OS fingerprint from one or more protocol parameters if present in the TCP packets to determine a first OS ID;	<i>See</i> [2A].
[16B] determine an application parameter associated with the TCP session if present, and identifying a second OS ID from the application parameter; and	<i>See</i> [2B].

Claim	Exemplary Citation from Lee
[16C] wherein the identified OS ID in the alert comprises one of the first OS ID or the second OS ID associated with one or more computing devices coupled to the access device.	<i>See</i> [2C].
[17] The system of claim 15 wherein the plurality of network sensors further: send a notification to a subscriber associated with the network address associated with the access device, the notification identifying a remediation portal to remove the determined malware in the alert.	<i>See</i> claim 3
[18] The system of claim 17 wherein the plurality of network sensors send the notification to the subscriber associated with the access device by retrieving a subscriber ID from a subscriber ID repository and replacing the network address of the access device with the subscriber ID.	<i>See</i> claim 12.

Claim	Exemplary Citation from Lee
<p>[19] The system of claim 16 wherein the plurality of network sensors determine the first OS ID by comparing one or more protocol parameters of the one or more TCP packets having a SYN flag associated with the session, the OS fingerprints identify the protocol parameters of the SYN flagged packet and the OS ID associated with parameters.</p>	<p><i>See claim 7.</i></p>
<p>[20] The system of claim 19 wherein the plurality of network sensors determine the application parameter associated with the session and identifies the second OS ID from the application parameter by inspecting an HTTP user agent string to determine the second OS ID.</p>	<p><i>See claim 8.</i></p>
<p>[21] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID does not match the first OS ID the</p>	<p><i>See claim 9.</i></p>

Claim	Exemplary Citation from Lee
session is identified by the OS ID providing a more reliable OS ID.	
[22] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID either OS ID is used.	<i>See claim 10.</i>
[23] The system of claim 20 wherein the plurality of network sensors verify the first OS ID against the second OS ID, wherein if the second OS ID matches the first OS ID but the second OS ID provides additional information, the second OS ID is used.	<i>See claim 11.</i>
[24PRE] A computer readable memory containing instructions for network based malware detection in a service provider network, the instructions when executed by a processor performing:	<i>See [1PRE].</i>
[24A] receiving one or more transmission control protocol (TCP) packets	<i>See [1A].</i>

Claim	Exemplary Citation from Lee
<p>originating from an access device coupled to the service provider network, the one or more TCP packets defining a TCP session between a computing device coupled to the access device, and a destination coupled to the service provider network;</p>	
<p>[24B] determining an operating system identifier (OS ID) associated with the TCP session and the computing device;</p>	<p><i>See</i> [1B].</p>
<p>[24C] determining if malware is present in the TCP session and an associated malware ID by comparing a malware signature to the one or more TCP packets; and</p>	<p><i>See</i> [1C].</p>
<p>[24D] generating an alert identifying a network address associated with the access device, the malware ID and the OS ID associated with TCP session that generated the alert.</p>	<p><i>See</i> [1D].</p>
<p>[25PRE] The system of claim 16 further comprising:</p>	<p><i>See</i> [4PRE].</p>

Claim	Exemplary Citation from Lee
[25A] a remediation portal coupled to the service provider network, the remediation portal for:	<i>See</i> [4A].
[25B] identifying a subscriber accessing the remediation portal;	<i>See</i> [4A].
[25C] identifying the OS associated with the computing device used by the subscriber to access the remediation portal;	<i>See</i> [4B].
[25D] comparing the OS associated with the computing device to the OS ID identified in the alert; and	<i>See</i> [4C].
[25E] providing malware remediation to the computing device.	<i>See</i> [4D].