Title: CLOUD SECURITY PLATFORM

Abstract: A data security system, including a security manager remotely monitoring a service that performs data-exchange transactions for end users, operative to temporarily prevent end user access to new transactions by interacting with the service so that the security manager accesses new transactions prior to end users being able to access the new transaction, and a data inspector operative to inspect data of new transactions for security clearance, wherein the security manager is further operative to restore end user access to the new transactions by further interacting with the service so that end users are able to access the new transaction, if the security inspector clears the data, and otherwise to perform a remedial action vis-a-vis the new transactions.
CLOUD SECURITY PLATFORM

PRIORITY REFERENCES

[ooo1] This application claims benefit of and hereby incorporates by reference US Provisional Application No. 62/271,234, entitled NONINLINE METHOD FOR BLOCK-AND-INSPECT BY REVOCATION-RESTORATION SYSTEM, and filed on December 27, 2015 by inventors Roy Rotem, Gil Friedrich and Avraham Zelovich.
FIELD OF THE INVENTION

[0003] The present invention relates to data security for cloud-based services.

BACKGROUND OF THE INVENTION

[0004] In today's work environment a major part of data communication is done over computer networks. Data needs to be inspected for presence of malware, and for presence of sensitive data. Often, the inspection process is not immediate and is resource-intensive, yet the receiver desires to prevent access to the data until the inspection completes and the data is allowed to continue its regular course.

[0005] Common examples of inspection of data include:

• e-mail systems must check that all incoming file-attachments are safe from computer viruses, phishing schemes or other malware;
• file sharing systems must check that files are clean from malicious content before being accessed or being shared with other parties;
• messaging/collaboration systems must ensure that data flowing between entities does not leak sensitive information;
• systems that share sensitive-data often need to encrypt that data upon send and arrival.

[0006] Reference is made to FIG. 1, which is a simplified block diagram of a prior art security system, in accordance with an embodiment of the present invention. FIG. 1 shows an enterprise data center 100 that provides various data exchange services to enterprise employees. The data exchange services include a collaborative document management service 220, such as MICROSOFT EXCHANGE SERVER® developed by Microsoft Corporation of Redmond, WA, USA, and a file sharing service
230, such as DOCUMENTUM® developed by EMC Corporation of Hopkinton, MA, USA. Employees 110A, HOB and HOC access these services using any of a variety of devices, including inter alia a smartphone, a laptop computer and a desktop computer. Also shown in FIG. 1 is a firewall 150 which scans incoming and outgoing data for malware and for leakage of sensitive data.

[0007] Firewall 150 scans incoming data scan before the receiver has access to that data, using proxy technologies that often perform block-and-inspect. Specifically, network traffic between a sender and a receiver must flow through firewall 150, which inspects data and, if the data is cleared, forwards the data to the receiver. Firewall 150 is often a store-and-forward device, which blocks the flow of the traffic until inspection completes. Other store-and-forward devices include mail-relay, routers, proxy-servers, software agents and other inspection modules. This type of inspection is referred to as being inline. Inline inspection techniques are described at https://en.wikipedia.org/wiki/Deep_packet_inspection.

[0008] Conventional inline block-and-inspect techniques suffer from several drawbacks, including:

- Inline systems cannot fully support a fail-open mode, and introduce an additional point-of-failure to services 120 and 130.
- Block-and-inspect introduces additional latency, during which traffic is re-routed to an inline device and then back to services 120 and 130.
- Introduction of inspection entities into a network disrupts normal operation of other components of services 120 and 130. For example, an added email mail transport agent adversely impacts
anti-spam filters used by an e-mail server, because an original source IP address is no longer visible.

- Traffic that is encrypted cannot be inspected easily by inline systems.
- Inline systems increase network-latency, and negatively impact the end user experience.

[0009] Even worse, in today's off-site environments such as cloud/SaaS services, firewall 150 cannot inspect traffic, because traffic often flows over a public-network to computing and storage systems that the enterprise does not own or control. Moreover, today's end users may access these services from any location, not only from an enterprise office. Some enterprises try to force traffic through a proxy to ensure that the traffic undergoes inspection, but this leads to difficulties of managing endpoint proxy configurations, and of overcoming network-topology limitations.

[0010] Reference is made to FIG. 2 is a simplified block diagram of a prior art data security system that gets bypassed by end users of cloud applications. Shown in FIG. 2 are enterprise data center 100 and firewall 150, and a variety of cloud-based data-exchange services that run in a cloud computing center 200, the services including a collaborative document management service 220, such as OFFICE 365® developed by Microsoft Corporation of Redmond, WA, a file sharing service 230, such as BOX.NET® developed by Box.net, Inc. of Palo Alto, CA, and an e-mail service 240, such as GMAIL® developed by Google Inc. of Mountain View, CA. End users 210A, 210B and 210C access these services, bypassing firewall 150, using any of a variety of devices,
including inter alia a smartphone, a laptop computer and a desktop computer.

[0011] An enterprise’s sensitive files are now in the enterprise cloud, and may be shared with external users; and malicious files may be enter or reside in the enterprise cloud. An enterprise, in fact any business, large or small, and even an individual using cloud services, requires protection against malware, against advanced persistent threats, against anomalies, against insider threats and against data leakage. An enterprise may require data sanitation, endpoint compliance, share policy management, security information and event management (SIEM), ticketing integration, and audit compliance. Cloud service providers do not always offer such comprehensive protection and management, and such protection and management is not covered by service level agreements.

[0012] Because of these drawbacks enterprises must compromise between inline prevention systems and non-inline detection-only systems for which the system sends alerts but cannot block/prevent a security breach from occurring.

[0013] It would this be of advantage to provide a robust security system that provides the requisite security, yet overcomes these drawbacks of in¬line block-and-inspect systems for today’s cloud/SaaS environments.
SUMMARY

[0014] Embodiments of the present invention provide robust security systems that overcome drawbacks of conventional in-line block-and-inspect systems for today's cloud/SaaS environments. These systems are essential not only to large enterprises, but also to business, large and small, and to individuals who use cloud services.

[0015] Embodiments of the present invention provide non-inline prevention methods, which prevent end users from accessing data during an inspection phase until access is cleared, while not being inline. Because they are not inline, these embodiments overcome the drawbacks of existing inline systems. Because they block access to data until the data is cleared, these embodiments overcome drawbacks of detection-only systems.

[0016] Embodiments of the present invention use network application programming interfaces to remotely control cloud-based services, including inter alia e-mail, file-sharing, collaborative data processing services.

[0017] Embodiments of the present invention protect against malware and data leakage for both incoming and outgoing data transactions, such as inter alia a new file that is uploaded or downloaded or shared, or a new e-mail that arrives or is sent. The data transactions may be inter-enterprise communications, between employees of an enterprise, or may be communications to or from sources external to the enterprise.

[0018] There is thus provided in accordance with an embodiment of the present invention a data security system, including a security manager remotely controlling, via a network application programming interface, a cloud-based service that performs data-exchange transactions for end
users, operative to cause the service to prevent end user access to incoming transactions so that the security manager accesses the incoming transactions prior to end users being able to access the incoming transactions, and a data inspector operative to inspect data of incoming transactions for security clearance by invoking one or more content scanners, wherein the security manager is further operative to cause the service to restore end user access to the incoming transactions, when the security inspector clears the data, and to cause the service to perform a remedial action vis-à-vis the incoming transactions, when the security inspector does not the data.

[0019] There is additionally provided in accordance with an embodiment of the present invention a method for data security, including remotely controlling, by a security processor via a network application programming interface, a cloud-based service that performs data-exchange transactions for end users, causing, by the security processor, the service to prevent end user access to incoming transactions, including controlling the service so that the security processor accesses incoming transactions prior to end users being able to access the incoming transactions, inspecting data of incoming transactions for security clearance by invoking one or more content scanners, when the inspecting clears the data, then causing, by the security processor, the service to restore end user access to the incoming transactions, and when the inspecting does not clear the data, causing, by the security processor, the service to perform a remedial action vis-à-vis the incoming transactions.

[0020] There is further provided in accordance with an embodiment of the present invention a data security system, including a security manager remotely controlling, via a network application programming
interface, a service that performs data-exchange transactions for end users, operative to cause the service to prevent transmission of outgoing transactions so that the security manager accesses outgoing transactions prior to transmission of the outgoing transactions to their destinations, and a data inspector operative to inspect data of outgoing transactions for data leakage, wherein the security manager is further operative to cause the service to transmit the transactions to their destinations, when the security inspector clears the data, and to perform a remedial action vis-à-vis the outgoing transactions, when the security inspector does not clear the data.

[0021] There is yet further provided in accordance with an embodiment of the present invention a method for data security, including remotely controlling, by a security processor via a network application programming interface, a service that performs data-exchange transactions for end users, causing, by the security processor, the service to prevent transmission of outgoing transactions, including controlling the service so that the security processor accesses outgoing transactions prior to the service transmitting the outgoing transactions to their destinations, inspecting data of outgoing transactions for data leakage breach, when the inspecting does not discover a breach, then causing, by the security processor, the service to transmit the outgoing transactions to their destinations, and when the inspecting discovers a breach, causing, by the security processor, the service to perform a remedial action vis-à-vis the outgoing transactions.
BRIEF DESCRIPTION OF THE DRAWINGS

[0023] The present invention will be more fully understood and appreciated from the following detailed description, taken in conjunction with the drawings in which:

[0024] FIG. 1 is a simplified block diagram of a prior art data security system;

[0025] FIG. 2 is a simplified block diagram of a prior art data security system that gets bypassed by end users of cloud applications;

[0026] FIG. 3 is a simplified block diagram of a cloud security system for end users, in accordance with an embodiment of the present invention;

[0027] FIG. 4 is a simplified block diagram of a cloud security system for an enterprise, in accordance with an embodiment of the present invention;

[0028] FIG. 5 is a simplified flowchart for a method for cloud security, in accordance with an embodiment of the present invention;

[0029] FIG. 6 is a simplified flowchart for a method for setting rules for cloud security, in accordance with an embodiment of the present invention;

[0030] FIG. 7 is a screen shot of a security stack interface, showing a configuration of services and data inspectors, in accordance with an embodiment of the present invention;

[0031] FIGS. 8 - 13 are respective screen shots of dashboards displaying results of several content scanners for six cloud services, in accordance with an embodiment of the present invention;
FIG. 14 is a screen shot of a main dashboard displaying overall results for several cloud services, in accordance with an embodiment of the present invention;

FIG. 15 is a screen shot of an analytics interface for cloud services, in accordance with an embodiment of the present invention;

FIG. 16 is a screen shot of an alerts interface for cloud services, in accordance with an embodiment of the present invention;

FIGS. 17 - 20 are self-explanatory simplified drawings of various policies enforced by a cloud security system, in accordance with an embodiment of the present invention; and

FIGS. 21 - 26 are self-explanatory simplified drawings of various workflows performed by a cloud security system, in accordance with an embodiment of the present invention.

For reference to the figures, the following index of elements and their numerals is provided. Similarly numbered elements represent elements of the same type, but they need not be identical elements.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>enterprise</td>
</tr>
<tr>
<td>110</td>
<td>end users</td>
</tr>
<tr>
<td>120</td>
<td>Microsoft Office software service, enterprise-based</td>
</tr>
<tr>
<td>130</td>
<td>Box.net file sharing service, enterprise-based</td>
</tr>
<tr>
<td>150</td>
<td>firewall</td>
</tr>
<tr>
<td>200</td>
<td>cloud computing center</td>
</tr>
<tr>
<td>210</td>
<td>end users</td>
</tr>
<tr>
<td>220</td>
<td>Microsoft Office software service, cloud-based</td>
</tr>
<tr>
<td>230</td>
<td>Box.net file sharing service, cloud-based</td>
</tr>
<tr>
<td>240</td>
<td>Google e-mail service, cloud-based</td>
</tr>
<tr>
<td>300</td>
<td>cloud security platform</td>
</tr>
<tr>
<td>310</td>
<td>security manager</td>
</tr>
<tr>
<td>320</td>
<td>data inspector</td>
</tr>
</tbody>
</table>

Elements numbered in the 1000's are operations of flow charts.
LIST OF APPENDICES

[0039] APPENDIX A is a listing with application programming interface (API) calls, for implementing an embodiment of the present invention.
DETAILED DESCRIPTION

[0040] In accordance with embodiments of the present invention, systems and methods are provided for non-inline security systems for cloud/SaaS computing environments.

[0041] Reference is made to FIG. 3, which is a simplified block diagram of a cloud security system for end users, in accordance with an embodiment of the present invention. Shown in FIG. 3 are enterprise data center 100 and firewall 150, and cloud-based data-exchange services that run in cloud computing center 200, the services including collaborative document management service 220, such as OFFICE 365® developed by Microsoft Corporation of Redmond, WA, USA, file sharing service 230, such as BOX.NET® developed by Box.net, Inc. of Palo Alto, CA, USA, and e-mail service 240, such as GMAIL® developed by Google Inc. of Mountain View, CA, USA. End users 210A, 210B and 210C access these services using any of a variety of devices, including inter alia a smartphone, a laptop computer and a desktop computer.

[0042] Also shown in FIG. 3 is a cloud security platform 300, including a security manager 310 and a data inspector 320. Security manager 300 remotely accesses the various data-exchange services 220, 230 and 240 using their respective application programming interfaces (APIs). Security manager 300 may itself be a cloud-based system. Security manager 310 and data inspector 320 may or may not reside on the same computer or even within the same cloud. Data inspector 320 may be, for example, its own cloud service.

[0043] Security manager 310 and data inspector 320 include programmable data processing, storage and communication circuitry for
performing the operations described below and with reference to the
flowchart of FIG. 5.

[0044] Reference is made to FIG. 4, which is a simplified block diagram
of a cloud security system for an enterprise 100, in accordance with an
embodiment of the present invention. FIG. 4 shows cloud computing
center 200 and cloud-based data-exchange services 220, 230 and 240.
FIG. 4 also shows cloud security platform 300, including security
manager 310 and data inspector 320.

[0045] Enterprise 100 generally has enterprise level subscriptions for its
employees to data-exchange services 220, 230 and 240. Enterprise
100 itself uses a firewall 150 to access data-exchange services 220, 230
and 240. However, employees of enterprise 100, such as end users
210A, 210B and 210C may access these services from locations outside
of enterprise 100, such as from their homes. Since these services are
cloud-based, the enterprise has no direct control over them.

[0046] Reference is made to FIG. 5, which is a simplified flowchart for a
method 1000 for cloud security, in accordance with an embodiment of
the present invention. Method 1000 is practiced by security manager
310 and data inspector 320, to provide security for incoming and
outgoing data transactions for services 220, 230 and 240. Many of the
operations in FIG. 5 are performed via services 220, 230 and 240
themselves; namely, security manager 310 uses a network application
programming interface (API) to control a service and thereby cause the
service to perform various operations. Although method 1000 relates to
both incoming and outgoing data transactions, it will be appreciated by
those skilled in the art that separate methods may instead be used for
incoming and outgoing data transactions.
Briefly, at operation 1005, security manager 310 monitors one or more data-exchange services, such as one or more of services 220, 230 and 240. At operation 1010, security manager 310 ascertains the scope of inspection configured by an administrator, as explained in detail hereinbelow. At operation 1015, security manager 310 optionally discovers in real-time occurrence of a new incoming or outgoing data transaction within the service, as explained in detail hereinbelow. Whether security manager 310 actively discovers the new data transaction at operation 1015, or instead relies on services 220, 230 to do so, at operation 1020, security manager 310 causes end-user access to the new transaction to be temporarily prevented, as explained in detail hereinbelow, if the new transaction is an incoming transaction, or causes transmission of the transaction to its destination to be temporarily prevented, if the new transaction is an outgoing transaction. Operation 1020 ensures that security manager 310 accesses data of the new transaction prior to end users and recipients being able to access that data. At operation 1025, security manager 310 optionally causes recipients to be alerted that the new transaction has been temporarily prevented. At operation 1030, security manager 310 retrieves the data of the new transaction. At operation 1035, data inspector 320 inspects the data of the new transaction, as explained in detail hereinbelow. At operation 1040, a determination is made as to whether or not the data of the new transaction passed the inspection of operation 1035. If so, then at operation 1045, security manager 310 optionally causes the new transaction to be marked with a special marker, indicating that the transaction passed inspection. At operation 1050, security manager 310 causes end user access to the new transaction to be restored, if the new
transaction is an incoming transaction, or causes the new transaction to be transmitted to its destination, if the new transaction is an outgoing transaction. Otherwise, if the data of the new transaction did not pass inspection, then at operation 1055, security manager 310 causes a remedial action to be performed vis-à-vis the new transaction.

[0048] Operations of the flowchart of method 1000 are now described in detail.

**Operation 1010**

[0049] At operation 1010 security manager 310 ascertains the scope of inspection configured by an administrator. The administrator configures the inspection so as to be applied/not be applied to specific users, specific groups of users, or to the entire enterprise. The administrator also configures the inspection so as to be applied/not be applied to various types of files including inter alia
e-mail attachments;
compressed files such as WINZIP® files;
editable documents such as Word documents, EXCEL® spreadsheets and POWERPOINT® presentations; and
multi-media content such as audio and video content.

**Operation 1015 (optional)**

[0050] At optional operation 1015, security manager 310 discovers in real-time occurrence of a new data transaction within the one or more data-exchange services, such as inter alia a new file that is uploaded or downloaded or shared, or a new e-mail that arrives or is sent. The new data transaction may be an inter-enterprise communication, between
employees of the enterprise, or a communication to or from a source external to the enterprise.

[0051] It will be appreciate by those skilled in the art that embodiments of the present invention have the advantage of being fail-safe; namely, if security manager 310 fails then the enterprise does not lose its access to services 220, 230 and 240.

[0052] Inter alia, operation 1015 may comprise, respectively, one or more of the following:

(i) services 220, 230 and 240 sends an event notification to security manager 310;

(ii) services 220, 230 and 240 forwards a message to security manager 310, when a data transaction occurs;

(iii) security manager 310 polls services 220, 230 and 240 for occurrence of data transactions.

In one embodiment of the present invention, polling is performed using a network API for the service.

Operations 1020 and 1050 - Incoming Transaction

[0053] At operation 1020, security manager 310 causes end-user access to a new incoming transaction to be temporarily prevented, ensuring that security manager 310 accesses the new transaction prior to end users being able to access that transaction. At operation 1050, security manager 310 cause end user access to the new transaction to be restored. In embodiments of the present invention, causing prevention and restoration are performed using a network API for the service.

[0054] In one embodiment of the present invention, operation 1020 is performed by using network API calls to quickly move a new transaction
from an end user accessible location, such as an end user's inbox, to a quarantine location, as soon as occurrence of the new transaction is discovered at operation 1015.

[0055] Operation 1020 may optionally include a workflow that is triggered when a transaction is moved to a quarantine location. The workflow may include causing the service to place a message in an end user accessible location, such as an end user's inbox, the message including an explanation of the quarantine operation, and some highlights from the quarantined transaction. The workflow may also or alternatively include causing the service to provide a link enabling an end user to request release of a transaction from quarantine, or receiving a request from a user to release a transaction and optionally prompting an administrator to allow or deny the user's request.

[0056] Operation 1050 may be performed using network API calls to move a transaction from a quarantine location to an end user accessible location, such as an end user's inbox. Alternatively, operation 1050 may be performed by using API calls to generate a transaction that is identical to the quarantined transaction, and placing the thus-generated transaction in the end user accessible location. Operation 150 may optionally cause the service to add helpful text into a restored transaction, indicating that the transaction has been scanned and cleared.

[0057] Operation 150 may further optionally check if a transaction to be restored includes hyperlinks and, if so, re-direct the hyperlinks to a safe browsing system that scans the content of a link destination upon activation of a hyperlink by a user, to check if the link destination is safe for browsing.
In another embodiment of the present invention, operation 1020 is performed by setting security platform rules within the one or more data-exchange services 220, 230 and 240, the rules causing the service to automatically move new transactions from end user accessible locations, such as end user inboxes, to quarantine locations. The security platform rules may be set manually or via network APIs. The security platform rules are created by cloud security manager 310, and set to be applied within the one or more data-exchange services. The one or more data-exchange services apply these rules prior to placing new transactions being in an end user's inbox.

If necessary for a particular cloud service, the security platform rules are set up so that they do not apply again to transactions that were restored at operation 1050. If needed, restored transactions are marked with a special marking, and the security platform rules are defined so as to ignore transactions that are thus-marked. Furthermore, when special markers are used at operation 1045, then for security purposes and as appropriate, the special markers are regularly changed from time to time, so that if a special marker is discovered by a hacker, he cannot use it for any length of time to bypass security manager 310.

The security platform rules may optionally be set up to be fail-safe: i.e., so that the enterprise does not lose access to services 220, 230 and 240 in case of a fatal failure of security manager 310. Specifically, the security platform rules may be set up so as to expire after a short time period, such as after five minutes, or so as to apply to data during a specific short time period, such as during a five minute interval, and shortly before or after the end of such time period new security platform rules are set up for a next short time period. As such, if
security manager 310 were to fail, then new transactions would not be processed by security manager 310, and enterprise data center 100 would operate as if the security layer of cloud security platform 300 had not been added. Transactions being processed during failure of security manager 310 would be re-introduced to end users after the short time period, and the enterprise would lose access to services 220, 230 and 240 for at most five minutes.

[0061] It is of advantage that the security platform rules move new transactions to quarantine locations that are not easily accessible to end users. The quarantine locations may be, for example, in a trash folder or a subfolder thereof. This embodiment has the advantage that contents of a trash folder generally go unnoticed by an end user. This embodiment has a further advantage that generally cloud services have a very simple API in place to restore a message from trash to its original location, which runs faster and scales better than restoring a message from a general quarantine folder.

[0062] The quarantine locations may be, for example, within a folder that is invisible to an end user, including inter alia a root folder, with an empty string name " " as its parent folder, or subfolder thereof. This embodiment has the advantage that the folder exists but is invisible and inaccessible to an end user.

[0063] In some embodiments of the present invention, a watch service is run, which identifies, via a network API, new transactions that are stuck in quarantine locations; i.e., transactions that were moved to a quarantine location at operation 1020, but that have not yet been restored at operation 1050 after lapse of a long time, such as 15 minutes. The watch service causes the service to move such transactions
back to user inboxes, automatically and/or conditionally, based on a policy configuration. The watch service provides a safety precaution that prevents a transaction from disappearing in case of an unexpected system problem or failure.

[0064] Reference is made to FIG. 6, which is a simplified flowchart for a method 1100 for setting rules for cloud security, in accordance with an embodiment of the present invention. Method 1100 begins at optional operation 1110, where a special marker is assigned, for marking restored data transactions, which already passed inspection. At operation 1120, security platform rules are activated, with (i) expiration after a short time period, such as five minutes, (ii) an optional setting to ignore data transactions that are marked with the special marker set at operation 1110, and (iii) with an optional first priority to precede any other rules set by an end-user but with a "no-stop" attribute so as not to preempt other rules set by an end-user. These settings ensure that the end users rules will be applied, but only after the security platform rules are applied. Regarding expiration of security platform rules, generally these rules are set so as to apply to transactions that arrive within a specific time period, such as a 5 minute time interval. If these rules are not refreshed after the time period, then they are no longer applied. As such, setting of their active time periods effectively sets their expiration.

[0065] At operation 1130, a decision is performed to determine whether or not the security platform rules have expired. If so, then the method returns to operation 1120 to re-activate the rules for a next short period of time. Otherwise, the method proceeds to optional operation 1140, where a decision is performed to determine whether or not the special marker should be changed. For security purposes, the special marker is
changed at regular periods of time, in case a hacker reverse engineers the marker and tries to by-pass the security platform rules. If the special marker should be changed, then the method returns to operation 1110 to re-set the marker. Otherwise, the method returns to operation 1130.

[0066] For an e-mail service 240, operations 1020 and 1045 may comprise, respectively, inter alia one or more of the following:

(i) security manager 310 causes service 240 to prevent end user access by causing service 240 to move the e-mail from an inbox of an end user to a quarantine folder or quarantine mailbox, and causes service 240 to restore end user access by causing service 240 to move the e-mail from the quarantine folder or quarantine mailbox to the inbox of the end user;

(ii) security manager 310 causes service 240 to prevent end user access by causing service 240 to remove an attachment from the e-mail, and causes service 240 to restore end user access by causing service 240 to reinsert the attachment into the e-mail;

(iii) security manager 310 causes service 240 to alter the e-mail, e.g., by adding text like "virus alert", so that an end user sees the e-mail in his mailbox, but the e-mail has a subject line or message body with warning text alerting the user of a risk;

(iv) security manager 310 causes service 240 to prevent end user access by causing service 240 to replace an attachment of the e-mail with a substitute attachment, and causes service 240 to restore end user access by causing service 240 to reinsert the original attachment into the e-mail;
(v) security manager 310 causes service 240 to prevent end user access by causing service 240 to encrypt an attachment of an e-mail, and causes service 240 to restore end user access by causing service 240 to decrypt the encrypted attachment.

[0067] For a file sharing service 230, operations 1020 and 1045 may comprise, respectively, inter alia one or more of the following:

(i) security manager 310 causes service 230 to prevent end user access by causing service 230 to instantly alter or revoke share access to the file, and causes service 230 to restore end user access by causing service 230 to re-share the file with its original recipients;

(ii) security manager 310 causes service 230 to prevent end user access by causing service 230 to instantly alter or revoke access rights to the file, and causes service 230 to restore end user access by causing service 230 to restore access rights to the file;

(iii) security manager 310 causes service 230 to prevent end user access by causing service 230 to move the file from a main folder to a quarantine folder, and causes service 230 to restore end user access by causing service 230 to move the file from the quarantine folder to the main folder;

(iv) security manager 310 causes service 230 to prevent end user access by causing service 230 to instantly designate the file as "online-only" such that the file cannot be downloaded, and causes service 230 to restore end user access by causing service 230 to designate the file as "downloadable";
(v) security manager 310 causes service 230 to prevent end user access by causing service 230 to instantly change an end user access level of the file so that the file is not visible to the end user, and causes service 230 to restore end user access by causing service 230 to restore the end user access level of the file;

(vi) security manager 310 causes service 230 to prevent end user access by causing service 230 to encrypt the file, and causes service 230 to restore end user access by causing service 230 to decrypt the encrypted file;

(vii) security manager 310 causes service 230 to prevent end user access by causing service 230 to replace the original file with a substitute file, and causes service 230 to restore end user access by causing service 230 to replace the substitute file with the original file.

Operations 1020 and 1050 - Outgoing Transaction

[0068] At operation 1020, security manager 310 causes transmission of a new outgoing transaction to be prevented, ensuring that security manager 310 accesses the new transaction prior to that transaction being transmitted. At operation 1050, security manager 310 transmits the new transaction to its destination. In some embodiments of the present invention, prevention and restoration are performed using a network API for the service.

[0069] Some services 220, 230 and 240 allow for "delayed sending", setting a delay time e.g., 30 seconds, during which time an end user may cancel transmission of a data transaction. Security manager 310 may
use this time period to scan the transaction using content filters and data leakage prevention scanners. If the transaction is discovered to violate an enterprise policy, then at operation 1020 the transaction is moved to a quarantine folder and blocked from being transmitted.

**Operation 1025 (optional)**

[0070] At operation 1025, security manager optionally causes the service to alerts recipient that the new transaction has been temporarily prevented, and that an inspection process is being performed. In one embodiment of the present invention, causing the alerting is performed using a network API for the service.

[0071] Operation 1025 may comprise inter alia one or more of the following:

(i) send a notification e-mail, resembling the newly arrived e-mail, to the same recipients as those of the newly arrived e-mail, with an explanation that the e-mail is being scanned, and/or instructions regarding the inspection process, and replace the notification e-mail with the newly arrived email after the latter passes inspection;

(ii) attach a notification file, instead of the original attachment, to a newly-arrived e-mail, with an explanation and/or instructions regarding the inspection process;

(iii) share a file, that has similar access rights and a similar name and folder to a newly shared file, with an explanation and/or instructions regarding the inspection process.

[0072] It is noted that operation 1020 ensures that an end user does not have a chance to interact with not-yet-inspected data, and operation
1025 informs the end user that new data is currently being inspected. Operations 1020 and 1025 may occur simultaneously, before the actual inspection is conducted. Alternatively, operation 1025 may occur at another stage in the flowchart of method 1000, and not necessarily between operations 1020 and 1030.

**Operation 1035 - Incoming Transaction**

[0073] At operation 1035, data inspector 320 inspects the data of a new incoming transaction. Data inspector 320 may use any of a variety of manual or automated content scanning technologies, as required by enterprise 100. It is noted that operation 1035 does not need to be performed on-the-fly in real time, since end uses cannot interact with the data being inspected during operation 1035.

[0074] Reference is made to FIG. 7, which is a screen shot of a security stack interface, showing a configuration of cloud services and data inspectors, in accordance with an embodiment of the present invention. For each of a plurality of cloud services, such as services 220, 230 and 240, an administrator selects which of a plurality of commercial content scanners are applied to that service. The configuration is displayed as a matrix for which each row is a different service, including inter alia:

- BOX® of Box.net, Inc. of Palo Alto, CA, USA,
- GMAIL® of Google, Inc. of Mountain View, CA, USA,
- OFFICE 365® One Drive of Microsoft Corporation of Redmond, WA, USA, OFFICE 365® E-mail of Microsoft Corporation of Redmond, WA, USA,
- DROPBOX® of Dropbox, Inc. of San Francisco, CA, USA,
• GOOGLE® Drive developed by Google Inc. of Mountain View, CA, USA,
• SLACK® developed by Slack Technologies, Inc. of San Francisco, CA, USA,
• AMAZON® S3 cloud storage developed by Amazon Technologies, Inc. of Reno, NV, USA, and
• EGNYTE® of Egnyte Inc. of Mountain View, CA, USA, and
• SHAREFILE® by Citrix Systems Inc. of Raleigh, NC, USA.

Each column is a different content scanner, SCN1, SCN2, ..., SCN 15. The content scanners may include inter alia:

• SOPHOS® developed by Sophos pic Corporation of Oxfordshire, England,
• CHECK POINT® developed by Checkpoint Software Technologies of Tel Aviv, Israel,
• AHNLAB® developed by Ahnlab, Inc. of South Korea,
• AVG® developed by AVG Netherlands B.V. of Amsterdam,
• AVIRA® developed by Avira Operations GmbH of Germany,
• BITDEFENDER® developed by Bitdefender IPR Management Ltd. of Cyprus,
• ESET® developed by Eset spol. s.r.o. Corporation of Bratislava Slovakia,
• KASPERSKY® developed by Kaspersky Lab of Moscow, Russia,
• F-PROT® developed by Cyren Inc. of McLean, VA, USA,
• CYANCE® developed by Cylance Inc., of Irvine, CA, USA, and
• SYMANTEC® developed by Symantec Corporation of Mountain View, CA, USA.
Reference is made to FIGS. 8 - 13, which are respective screen shots of dashboards displaying results of several content scanners for six cloud services (Google Drive, Box.net, Office 365 One Drive, Office 365 E-mail, Gmail and Dropbox), in accordance with an embodiment of the present invention.

FIG. 8 shows a dashboard console for Google Drive, displaying statistics of overall traffic and malicious content in a Security Stack zone, showing that Scanner #1 found 62 suspicious files out of a total of 215 files, Scanner #2 found 2 suspicious files out of the 215 files, Scanner #3 found 63 suspicious files out of the 215 files, Scanner #4 found 66 suspicious files out of the 215 files, and Scanner #5 found 69 suspicious files out of the 215 files.

FIG. 9 shows a dashboard console for Box.net, displaying statistics of overall traffic and malicious content in a Security Stack zone, showing that Scanner #1 found 41 suspicious files out of a total of 382 files, Scanner #2 found 16 suspicious files out of the 382 files, Scanner #3 found 38 suspicious files out of the 382 files, Scanner #4 found 55 suspicious files out of the 382 files, and Scanner #5 found 67 suspicious files out of the 382 files.

FIG. 10 shows a dashboard console for Office 365 One Drive, displaying statistics of overall traffic and malicious content in a Security Stack zone, showing that Scanner #1 found 1 suspicious file out of a total of 112 files, Scanner #2 found 1 suspicious file out of the 112 files, Scanner #3 found no suspicious files out of the 112 files, Scanner #4 found 3 suspicious files out of the 112 files, Scanner #5 found 1 suspicious file out of the 112 files, Scanner #6 found 1 suspicious file out of the 112 files, Scanner #7 found 3 suspicious files out of the 112 files,
Scanner #8 found 2 suspicious files out of the 112 files, Scanner #9 found no suspicious files out of the 112 files, and Scanner #10 found 3 suspicious files out of the 112 files.

[0079] FIG. 11 shows a dashboard console for Office 365 e-mail, displaying statistics of overall traffic and malicious content in a Security Stack zone, showing that Scanner #1 found 18 suspicious e-mails out of a total of 19,018 e-mails, Scanner #2 found 18 suspicious e-mails out of the 19,018 e-mails, Scanner #3 found 29 suspicious e-mails out of the 19,018 e-mails, Scanner #4 found 280 suspicious e-mails out of the 19,018 e-mails, Scanner #5 found 31 suspicious e-mails out of the 19,018 e-mails, Scanner #6 found 29 suspicious e-mails out of the 19,018 e-mails, Scanner #7 found 297 suspicious e-mails out of the 19,018 e-mails, Scanner #8 found 280 suspicious e-mails out of the 19,018 e-mails, Scanner #9 found 29 suspicious e-mails out of the 19,018 e-mails, and Scanner #10 found 300 suspicious e-mails out of the 19,018 e-mails.

[0080] FIG. 12 shows a dashboard console for Gmail, displaying statistics of overall traffic and malicious content in a Security Stack zone, showing that Scanner #1 found 1 suspicious e-mail out of a total of 409 e-mails, Scanner #2 found no suspicious e-mails out of the 409 e-mails, Scanner #3 found no suspicious e-mails out of the 409 e-mails, Scanner #4 found 323 suspicious e-mails out of the 409 e-mails, and Scanner #5 found 16 suspicious e-mails out of the 409 e-mails, Scanner #6 found 16 suspicious e-mails out of the 409 e-mails, Scanner #7 found 339 suspicious e-mails out of the 409 e-mails, Scanner #8 found 323 suspicious e-mails out of the 409 e-mails, Scanner #9 found no
suspicious e-mails out of the 409 e-mails, and Scanner #10 found 59 suspicious e-mails out of the 409 e-mails.

[0081] **FIG. 13** shows a dashboard console for Dropbox, displaying statistics of overall traffic and malicious content in a Security Stack zone, showing that no files were processed.

[0082] Reference is made to **FIG. 14**, which is a screen shot of a main dashboard displaying overall results for several cloud services, in accordance with an embodiment of the present invention. **FIG. 14** shows a dashboard visualization of statistics of overall traffic, malicious content blocked, and data-loss prevention (DLP). Specifically, **FIG. 14** indicates a total of 6,261 malicious files, and 365 malicious incoming files; a total of 1,066 files with data leaks, and 445 outgoing files with data leaks.

[0083] Reference is made to **FIG. 15**, which is a screen shot of an analytics interface for cloud services, in accordance with an embodiment of the present invention. **FIG. 15** shows results of a queries submitted by an administrator. Specifically, **FIG. 15** indicates 2 driver's license files shared externally on Box.net, 34 credit card numbers shared externally on Google Drive, 44 company confidential files on Box.net, etc.

[0084] Reference is made to **FIGS 16**, which is a screen shot of an alerts interface for cloud services, in accordance with an embodiment of the present invention. **FIG. 16** shows visualizations of statistics of alerts made by security manager 310. Specifically, **FIG. 16** indicates 11,676 alerts of which 9,266 are of high seventy, 2,287 are of medium severity, 65 are of low severity, and 58 are of no severity; 11,675 of the alerts are new alerts, and 1 alert is dismissed; 6,333 of the alerts are from Amazon S3 cloud, 210 of the alerts are from Slack, 2,352 of the alerts are from Microsoft Office 365 e-mails, 406 of the alerts are from Egnyte, 613 of
the alerts are from Gmail, 1,315 of the alerts are from Box.net, and 447 of the alerts are from Google drive.

**Operation 1035 - Outgoing Transaction**

[0085] At operation 1035, data inspector 320 inspects the data of a new outgoing transaction. Data inspector 320 may use any of a variety of manual or automated data leakage protection technologies, as required by enterprise 100.

**Operation 1055**

[0086] At operation 1055, security manager 310 causes the service to perform a remedial action vis-à-vis the new transaction. In one embodiment of the present invention, causing of the remedial action is performed using a network API for the service.

[0087] Operation 1055 may comprise inter alia one or more of the following:

(i) causing the service to permanently prevent end user access to the data of the transaction;

(ii) causing the service to replace the data of the transaction with modified data;

(iii) causing the service to alerts an administrator of enterprise 100.

[0088] In addition, security manager 310 may conducts an analysis the data of the transaction possibly via a security analysis cloud service.
Policy Enforcement

[0089] It will be appreciated by those skilled in the art that, in addition to scanning of content for data transactions, cloud security platform 300 provides security policy enforcement for enterprise 100.

[0090] Reference is made to FIGS. 17 - 20, which are self-explanatory simplified drawings of various policies enforced by cloud security system 300, in accordance with an embodiment of the present invention.

[0091] FIG. 17 shows a policy for data leakage scan and protection.

[0092] FIG. 18 shows a policy for malware protection.

[0093] FIG. 19 shows a policy for compliance enforcement for Saas login, using cloud security platform 300 as a single sign-on (SSO) proxy and using the EPOLICY ORCHESTRATOR® (ePO) developed by McAfee, Inc. of Santa Clara, CA.

[0094] FIG. 20 shows a policy for detection of anomalies including data leakage, internal threats, administrative breaches and lost devices.

[0095] Reference is made to FIGS. 21 - 26, which are self-explanatory simplified drawings of various workflows performed by cloud security system 300, in accordance with an embodiment of the present invention.

[0096] FIG. 21 shows a workflow for time limits of external sharing.

[0097] FIG. 22 shows a workflow for consent of sharing data with external users.

[0098] FIG. 23 shows a workflow for external users offloading data.

[0099] FIG. 24 shows a workflow for forwarding of shared content.

[0100] FIG. 25 shows a workflow for termination of an enterprise employee including operations of blocking the employee login to enterprise services, un-sharing that employee’s externally shared files, and remote wiping of that employee’s synchronized files.
FIG. 26 shows a workflow for backup and version management.

Implementation Details

Appendix A includes a listing with APIs for implementing an embodiment of the present invention, for specific use with Microsoft Office 365 and Google Gmail. Appendix A is structured as follows.

OFFICE 365

1. API for obtaining internal user data
   Example of obtaining internal user data
2. API for managing rules
   Example of rule creation
   Example of rule removal
3. API for creating a folder
4. API for subscribing to web notification
5. API for downloading a message/attachment
6. API for moving a message to Inbox (release)
7. API for quarantining a message to Inbox (quarantine)
   Example of quarantining a message
8. API for restoring a message to Inbox (restore)
   Example of restoring a message

GMAIL

1. Creating an API object
2. API for obtaining user data
3. API for managing rules
4. API for creating labels
5. API for subscribing to notifications
6. API for reading emails from notifications
7. API for downloading attachments
8. API for changing labels (release)
9. API for quarantine and admin notification

Advantages

[00103] It will thus be appreciate by those skilled in the art that embodiments of the present invention provide the following advantages vis-à-vis conventional enterprise security systems. Because these embodiments are network-API-based, then do not rely on a proxy and are not inline. They are immediately deployable and do not require changes to an enterprise network configuration, nor to the end user experience.

- These embodiments effectively block transactions in real-time without being inline.
- These embodiments apply to remote data services, cloud-based services, and SaaS applications not under control of the enterprise.
- These embodiments do not alter flow of data over a network, and thus do not adversely impact other functional components such as spam-filters, firewalls, routers and proxy servers.
- These embodiments do not adversely impact latency and responsiveness of services 220, 230 and 240.
- These embodiments are transparent to end users, because their transactions are restored immediately upon completion of inspection.
- These embodiments allow participants to partially continue a transaction while inspection of the sensitive components is taking
place. For example, an email message will be available while its original attachments are being scanned.

• These embodiments are fail-safe, and if security manager 310 fails, the flow of data continues normally. By not proxying traffic and being fail-safe, these embodiments do not introduce any new points of failure.

• These embodiments use network-APIs provided by services 220, 230 and 240, thus not requiring reverse-engineering, and not breaking support contracts.

• These embodiments use the same configuration and network infrastructure for monitoring/detection and blocking/prevention. As such, the enterprise may start with detection and upgrade to prevention, by changing the configuration of cloud security platform 300, without requiring changes to enterprise networks or to enterprise systems.

• These embodiments may be selectively applied to specific users, so that they ignore other users. This has a clear advantage over proxy solutions, which proxy data for all users even if only a few users are being protected.

• These embodiments are scalable for large enterprises.

[00104] In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made to the specific exemplary embodiments without departing from the broader spirit and scope of the invention. Accordingly, the specification
and drawings are to be regarded in an illustrative rather than a restrictive sense.
APPENDIX A

OFFICE 365

1. API for obtaining internal user data
Azure AD Graph API (1) is used to fetch the organization's domain in Azure Active Directory. Then, for each domain, the same API is used to fetch the users for each domain (2).

References
(2) https://msdn.microsoft.com/en-us/library/azure/ad/graph/api/users-operations#GetUsers

Example of obtaining internal user data

```json
{  
  "mailNickname": "john21_hotmail.com#EXT#",  
  "postalCode": null,  
  "surname": null,  
  "passwordProfile": {  
    "enforceChangePasswordPolicy": false,  
    "password": null,  
    "forceChangePasswordNextLogin": true
  },  
  "assignedLicenses": [],  
  "lastDirSyncTime": null,  
  "userPrincipalName": "john21_hotmail.com#EXT#@avan.onmicrosoft.com",  
  "passwordPolicies": null,  
  "userType": "Guest",  
  "usageLocation": null,  
  "objectType": "User"
}
```

Atty. Docket No. AVN-N-P002PCT
"city": null,
"assignedTo": [],
"objectId": "3a6e6629-afed-46d6-a45f-77312ebe045d",
"facsimileTelephoneNumber": null,
"creationType": null,
"streetAddress": null,
"state": null,
"department": null,
"mail": "shaillechuk@gmail.com",
"preferredLanguage": null,
"accountEnabled": true,
"refreshTokensValidFromDate": "2016-01-10T12:47:29Z",
"companyName": null,
"jobTitle": null,
"isCompromised": null,
"immutableId": null,
"otherMails": ["ifat21@hotmail.com"],
"proxyAddresses": [
  "smtp:john21_hotmail.com#EXT#@avanan.onmicrosoft.com",
  "SMTP:blussmith@gmail.com"
],
"physicalDeliveryOfficeName": null,
"sipProxyAddress": null,
"telephoneNumber": null,
"displayName": "john doe",
"provisionedPlans": [],
"deletionTimestamp": null,
"email": null,
"country": null,
"privacyTermsAccepted": [],
"signInNames": [],
"dirSyncEnabled": null,
"onPremisesSecurityIdentifier": null,
"givenName": null}
2. API for managing rules

Example of rule creation

```xml
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
    xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

    <soap:Header>
        <t:RequestServerVersion Version="Exchange 2013"/>
        <t:ExchangeImpersonation>
            <t:ConnectingSID>
                <t:PrincipalName>{user_email}</t:PrincipalName>
            </t:ConnectingSID>
        </t:ExchangeImpersonation>
    </soap:Header>

    <soap:Body>
        <m:UpdateInboxRules>
            <m:RemoveOutlookRuleBlob>false</m:RemoveOutlookRuleBlob>
            <m:Operations>
                <t:CreateRuleOperation>
                    <t:Rule>
                        <t:DisplayName>{name}</t:DisplayName>
                        <t:Priority>{priority}</t:Priority>
                        <t:IsEnabled>true</t:IsEnabled>
                        <t:Conditions>
                            <t:ContainsHeaderStrings>
                                <t:String>spf</t:String>
                            </t:ContainsHeaderStrings>
                            <t:WithinDateRange>
                                <t:StartDateTime>{start_dt}</t:StartDateTime>
                            </t:WithinDateRange>
                        </t:Conditions>
                    </t:Rule>
                </m:Operations>
            </m:RemoveOutlookRuleBlob>
        </m:UpdateInboxRules>
    </soap:Body>
</soap:Envelope>
```
Example of rule removal

```xml
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:xm="http://schemas.microsoft.com/exchange/services/2006/messages"
               xmlns:x="http://schemas.microsoft.com/exchange/services/2006/types"
               xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2013"/>
    <t:ExchangeImpersonation>
      <t:ConnectingSID/>
      <t:PrincipalName>{{user_email}}</t:PrincipalName>
    </t:ExchangeImpersonation>
  </soap:Header>
  <soap:Body>
    <!-- Rule creation XML goes here -->
  </soap:Body>
</soap:Envelope>
```
<soap:Header>
  <soap:Envelope>
    <soap:Body>
      <m:UpdateInboxRules>
        <m:RemoveOutlookRuleBlob>false</m:RemoveOutlookRuleBlob>
        <m:Operations>
          <t:DeleteRuleOperation>
            <t:RuleId>{ rule_id }</t:RuleId>
          </t:DeleteRuleOperation>
        </m:Operations>
      </m:UpdateInboxRules>
    </soap:Body>
  </soap:Envelope>
</soap:Header>

3. **API for creating a folder**
Outlook Mail REST API (1) is used to perform folders creation.
Request:
  ```
  api.user_folder_add(<user_email>, <folder_name>, <parent_folder_id>, <is_hidden>)
  ```
Response:
  ```
  status, folder_id
  ```
References:
  (1) [https://msdn.microsoft.com/en-us/office/office365/api/mail-rest-operations](https://msdn.microsoft.com/en-us/office/office365/api/mail-rest-operations)

4. **API for subscribing to web notification**
Outlook Push Notification REST API (1) is used to create a user subscription to receive web notifications.
References:
5. **API For downloading a message/attachment**

Outlook Mail REST API (1) is used to download email messages and attachments.

**References**

- To download email messages:
  
  (1) https://mcdn.microsoft.com/en-us/office/office365/api/mail-rest-operations#get-messages

- To download attachments:
  
  (1) https://mcdn.microsoft.com/en-us/office/office365/api/mail-rest-operations#get-attachments

6. **API for moving a message to Inbox (release)**

Outlook Mail REST API (1) is used to move messages in release process.

**References**

(1) https://mcdn.microsoft.com/en-us/office/office365/api/mail-rest-operations#move-or-copy-messages
7. API for quarantining a message to Inbox (quarantine)

Export message

API Reference:

Example of quarantining a message

```xml
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010_SP1"/>
    <t:ExchangeImpersonation>
      <t:ConnectingSID>
        <t:PrincipalName>{user_email}</t:PrincipalName>
      </t:ConnectingSID>
    </t:ExchangeImpersonation>
  </soap:Header>
  <soap:Body>
    <m:GetItem>
      <m:ItemShape>
        <t:BaseShape>IdOnly</t:BaseShape>
        <t:IncludeMimeContent>true</t:IncludeMimeContent>
      </m:ItemShape>
      <m:ItemIds>
        <t:ItemId Id="{message_id}"/>
      </m:ItemIds>
    </m:GetItem>
  </soap:Body>
</soap:Envelope>
```
8. API for restoring a message to Inbox (restore)

Import message


Example of restoring a message

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
    <soap:Header>
        <t:RequestServerVersion Version="Exchange2010_SP1" />
        <t:ExchangeImpersonation>
            <t:PrincipalName>{user_email}</t:PrincipalName>
        </t:ExchangeImpersonation>
    </soap:Header>
    <soap:Body>
        <m:CreateItem MessageDisposition="Save Only">
            <m:SavedItemFolderId Id="{folder_id}" />
        </m:CreateItem>
        <m:Items>
            <t:Message>
                <t:MimeContent CharSet="UTF-8">{message}</t:MimeContent>
                <t:ExtendedProperty>
                    <t:ExtendedFieldURI PropertyTag="0x0e07" PropertyType="Integer" />
                    <t:Value>K</t:Value>
                </t:ExtendedProperty>
            </t:Message>
        </m:Items>
    </soap:Body>
</soap:Envelope>
```
For send notification use user_message_create method of API Request:
api.user_message_create (<user_email>, <data>)
Response:
   status, sent message id
API reference:
htfcos://msdn.microsoft.com/en-us/office/office365/api/mail-operations#create-and-send-messages
Creating an API object

```python
from googleapiclient.discovery import build
from oauth2client.service_account import ServiceAccountCredentials

credentials = ServiceAccountCredentials.from_json_keyfile_name('path_to_json_keyfile',
    scopes= List of scopes)

http = httplib2.Http()
http = credentials.authorize(http)

api = build('admin', 'directory_v1', http=http)
```

API for obtaining user data

**Request:**
```python
api.users().list(<domain>)
```

**Response:**
```python
status, [users], next_page
```

**Reference:**
3. API for managing rules
Request:
   api.rule_create (<user_email>, <data>)
Response:
   status, rule_id
Reference:
   https://developers.google.com/gmail/api/v1/reference/users/settings/filters

4. API for creating labels
Request:
   api.labels_create (<user_email>, <user_id>, <data>)
Response:
   status, label_id
Reference:
   https://developers.google.com/gmail/api/v1/reference/users/labels/create

5. API for subscribing to notifications
Request:
   api.watch_email (<user_email>, <domain>)
Response:
   status
Reference:
   https://developers.google.com/gmail/api/cfuides-push
6. **API for reading emails from notifications**

   Notification for each user contains only history_id.

   Afterwards - get list of the changes for the user with the following API:

   **API Request:**
   ```
   api.history_list(<user_email>, <user_id>, <history_id>, <page_token>)
   ```

   **Response:**
   status, [users], next_page

   **Reference:**
   http://developers.google.com/gmail/api/v1/reference/users/history/list

7. **API for downloading attachments**

   **API**
   ```
   api.attachments_get(<user_email>, <owner_id>, <message_id>, <attachment_id>)
   ```

   **Response:**
   status, data

   **Reference:**
   https://developers.google.com/mail/api/v1/reference/users/messages/attachments/get
8. API for changing labels (release)

API
Request:
```
api.messages_modify(<user_email>, <user_id>, <message_id>, <changed_labels>)
```
Changed labels for "message release" have this form:
```
changed_labels = {
    removeLabelIds': [scan_folder_id],
    'addLabelIds': ['INBOX']
}
```
Response:
```
status
```
Reference
https://developers.google.com/gmail/api/v1/reference/users/messages/modify
9. **API for quarantine & admin notification**

**API for get message:**

```
https://developers.google.com/graai/api/v1/reference/users/messages/get
```

Request:

```html
api.messages_send(<user_email>, <user_id>, <data>)
```

Response:

```
status
```

**API for send message:**

```
https://developers.google.com/graai/api/v1/reference/users/messages/send
```

Request:

```html
api.messages_send(<user_email>, <user_id>, <message_id>)
```

Response:

```
status, data
```

**API for trash message:**

```
https://developers.google.com/graai/api/v1/reference/users/messages/trash
```

Request:

```html
api.messages_send(<user_email>, <user_id>, <message_id>)
```

Response:

```
status
```
CLAIMS

What is claimed is:

1. A data security system, comprising:
   a security manager remotely controlling, via a network application programming interface, a cloud-based service that performs data-exchange transactions for end users, operative to cause the service to prevent end user access to incoming transactions so that the security manager accesses the incoming transactions prior to end users being able to access the incoming transactions; and
   a data inspector operative to inspect data of incoming transactions for security clearance by invoking one or more content scanners, wherein said security manager is further operative to cause the service to restore end user access to the incoming transactions, when said security inspector clears the data, and to cause the service to perform a remedial action vis-à-vis the incoming transactions, when said security inspector does not the data.

2. The system of claim 1 wherein said security manager is operative to discover in real-time arrival of an incoming transaction by receiving a notification of the incoming transaction from the service, or by polling the service for existence of an incoming transaction, and, in response to discovery of the incoming transaction, to quickly prevent end user access to the discovered transaction.

3. The system of claim 1 wherein end users access the service from their endpoints over a public network that does not pass through said security manager.
4. The system of claim 1 wherein said security manager controls the service by automatically generating one or more security platform rules that are applied by the service and cause the service to automatically move incoming transactions from end user accessible locations to a quarantine location, prior to end users being able to recognize the incoming transactions.

5. The system of claim 4 wherein the quarantine location is within a root folder that is invisible to end users or a subfolder of such a root folder, or within a trash folder or a subfolder of a trash folder.

6. The system of claim 4 wherein each security platform rule ceases to apply after a short time duration, and wherein said security manager updates a security platform rule or generates a replacement security platform rule, shortly prior to or subsequent to expiration of a previous security platform rule, the replacement security platform rule also applied by the service to move incoming transactions from end user accessible locations to the quarantine location.

7. The system of claim 4 wherein said security manager is operative to cause the service to mark restored transactions so that the service does not apply the one or more security platform rules to restored transactions, and to dynamically change the marking of restored transactions from time to time.

8. The system of claim 4 wherein, when the service moves an incoming transaction from an end user accessible location to a quarantine location, said security manager is operative to cause the service to transmit a notice to an end user, the notice comprising a link that enables the end user to request that the incoming transaction be released from the quarantine location and made accessible to the end user.
9. The system of claim 1 wherein the service is an e-mail service, wherein the incoming transaction is arrival of an e-mail, and wherein said security manager causes the service to prevent end user access by causing the service to move an e-mail from an inbox of an end user to a quarantine folder or a quarantine mailbox, and causes the service to restore end user access by causing the service to move the e-mail from the quarantine folder or the quarantine mailbox to the inbox of the end user, or by causing the service to generate a copy of the e-mail that is in the quarantine folder or the quarantine mailbox and to place the copy in the inbox of the end user.

10. The system of claim 1 wherein the service is an e-mail service, wherein the incoming transaction is arrival of an e-mail comprising an attachment, and wherein said security manager causes the service:
   (i) to prevent end user access by removing the attachment from the e-mail, and to restore end user access by reinserting the attachment into the e-mail; or
   (ii) to prevent end user access by altering the e-mail so it has a subject line or message body with warning text, alerting end users of a risk;
   (iii) to prevent end user access by replacing the attachment of the e-mail with a substitute attachment, and to restore end user access by reinserting the original attachment into the e-mail; or
   (iv) to prevent end user access by encrypting the attachment of an e-mail, and to restore end user access by decrypting the encrypted attachment.

11. The system of claim 1 wherein the service is a file sharing service, wherein the incoming transaction is sharing of a file, and wherein said security manager causes the service:
(i) to prevent end user access by altering or revoking share access to the file, and to restore end user access by re-sharing the file with its original recipients; or
(ii) to prevent end user access by altering or revoking access rights to the file, and to restore end user access by restoring access rights to the file; or
(iii) to prevent end user access by moving the file from a main folder to a quarantine folder, and to restore end user access by moving the file from the quarantine folder to the main folder; or
(iv) to prevent end user access by designating the file as "online-only" such that the file cannot be downloaded, and to restore end user access by designating the file as "downloadable"; or
(v) to prevent end user access by changing an end user access level of the file so that the file is not visible to the end user, and to restore end user access by restoring the end user access level of the file; or
(vi) to prevent end user access by encrypting the file, and to restore end user access by decrypting the encrypted file; or
(vii) to prevent end user access by replacing the original file with a substitute file, and to restore end user access by replacing the substitute file with the original file.

12. The system of claim 1 wherein the remedial action comprises one or more of the following: (i) permanently prevent end user access to the data of the incoming transaction, (ii) replace the data of the incoming transaction with safe modified data, and (iii) alert an administrator.

13. A method for data security, comprising:
   remotely controlling, by a security processor via a network application programming interface, a cloud-based service that performs data-exchange transactions for end users;
causing, by the security processor, the service to prevent end user access to incoming transactions, comprising controlling the service so that the security processor accesses incoming transactions prior to end users being able to access the incoming transactions;

inspecting data of incoming transactions for security clearance by invoking one or more content scanners;

when said inspecting clears the data, then causing, by the security processor, the service to restore end user access to the incoming transactions; and

when said inspecting does not clear the data, causing, by the security processor, the service to perform a remedial action vis-à-vis the incoming transactions.

14. The method of claim 13 further comprising discovering, by the security processor in real-time, occurrence of an incoming transaction within the service, said discovering comprising receiving a notification of the incoming transaction from the service, or polling the service for existence of an incoming transaction, and wherein said causing the service to prevent is performed quickly in response to said discovering.

15. The method of claim 13 wherein said causing the service to prevent is performed by generating one or more security platform rules that are applied by the service and cause the service to automatically move incoming transactions from end user accessible locations to a quarantine location, prior to end users being able to recognize the incoming transactions.

16. The method of claim 15 wherein the quarantine location is within a root folder that is invisible to end users or a subfolder of such a root folder, or within a trash folder or a subfolder of a trash folder.
17. The method of claim 15 wherein each security platform rule ceases to apply after a short time duration, the method further comprising updating, by the security processor, a security platform rule or generating, by the security processor, a replacement security platform rule, shortly prior to or subsequent to expiration of a previous security platform rule, the replacement security platform rule also applied by the service to move incoming transactions from end user accessible locations to the quarantine location.

18. The method of claim 15 further comprising:
   causing, by the security processor, the service to mark restored transactions so that the service does not apply the one or more security platform rules to restored transactions; and
   causing, by the security processor, the service to dynamically change the marking of the restored data from time to time.

19. The method of claim 15 further comprising, when the service moves an incoming transaction from an end user accessible location to the quarantine location, causing, by the security processor, the service to transmit a message to an end user, the message comprising a link that enables the end user to request that the incoming transaction be released from the quarantine location and made accessible to the end user.

20. The method of claim 13 wherein the service is an e-mail service, wherein the incoming transaction is arrival of an e-mail, wherein said causing the service to prevent comprises causing the service to move the e-mail from an inbox of an end user to a quarantine folder or quarantine mailbox, and wherein said causing the service to restore comprises causing the service to move the e-mail from the quarantine folder or quarantine mailbox to the inbox of the end user, or to generate a copy of the e-mail that is in the quarantine
folder or the quarantine mailbox and placing the copy in the inbox of the end user.

21. The method of claim 13 wherein the service is an e-mail service, wherein the incoming transaction is arrival of an e-mail comprising an attachment, and wherein:

(i) said causing the service to preventing comprises causing the service to remove an attachment from an e-mail, and said causing the service to restore comprises causing the service to reinsert the attachment into the e-mail; or

(ii) said causing the service to prevent comprises causing the service to alter the e-mail so it has a subject line or message body with warning text, alerting end users of a risk;

(iii) said causing the service to prevent comprises causing the service to replace the attachment of the e-mail with a substitute attachment, and said causing the service to restore comprises causing the service to reinsert the original attachment into the e-mail; or

(iv) said causing the service to prevent comprises causing the service to encrypt the attachment of the e-mail, and said causing the service to restore comprises causing the service to decrypt the encrypted attachment.

22. The method of claim 13 wherein the service is a file sharing service, wherein the incoming transaction is sharing of a file, and wherein:

(i) said causing the service to prevent comprises causing the service to alter or revoke share access to the file, and said causing the service to restore comprises causing the service to re-share the file with its original recipients; or

(ii) said causing the service to prevent comprises causing the service to alter or revoke access rights to the file, and said causing the service to restore comprises causing the service to restore access rights to the file; or
(iii) said causing the service to prevent comprises causing the service to move
the file from a main folder to a quarantine folder, and said causing the service
to restore comprises causing the service to move the file from the quarantine
folder to the main folder; or
(iv) said causing the service to prevent comprises causing the service to
designate the file as "online-only" such that the file cannot be downloaded, and
said causing the service to restore comprises causing the service to designate
the file as "downloadable"; or
(v) said causing the service to prevent comprises causing the service to
change an end user access level of the file so that the file is not visible to the
end user, and said causing the service to restore comprises causing the service
to restore the end user access level of the file;
(vi) said causing the service to prevent comprises causing the service to
encrypt a file, and said causing the service to restore comprises causing the
service to decrypt the encrypted file; or
(vii) said causing the service to prevent comprises causing the service to
replace an original file with a substitute file, and said causing the service to
restore comprises causing the service to replace the substitute file with the
original file.

23. The method of claim 13 wherein said performing a remedial action
comprises one or more of the following:
(i) causing the service to permanently prevent end user access to the data of
the incoming transaction;
(ii) causing the service to replace the data of the incoming transaction with
modified data; and
(iii) causing the service to alert an administrator.

24. A data security system, comprising:

-58-

SUBSTITUTE SHEET (RULE 26)
a security manager remotely controlling, via a network application programming interface, a service that performs data-exchange transactions for end users, operative to cause the service to prevent transmission of outgoing transactions so that the security manager accesses outgoing transactions prior to transmission of the outgoing transactions to their destinations; and

a data inspector operative to inspect data of outgoing transactions for data leakage,

wherein said security manager is further operative to cause the service to transmit the transactions to their destinations, when said security inspector clears the data, and to perform a remedial action vis-à-vis the outgoing transactions, when said security inspector does not clear the data.

25. A method for data security, comprising:

remotely controlling, by a security processor via a network application programming interface, a service that performs data-exchange transactions for end users;

causing, by the security processor, the service to prevent transmission of outgoing transactions, comprising controlling the service so that the security processor accesses outgoing transactions prior to the service transmitting the outgoing transactions to their destinations;

inspecting data of outgoing transactions for data leakage breach;

when said inspecting does not discover a breach, then causing, by the security processor, the service to transmit the outgoing transactions to their destinations; and

when said inspecting discovers a breach, causing, by the security processor, the service to perform a remedial action vis-à-vis the outgoing transactions.
ABSTRACT
A data security system, including a security manager remotely controlling, via a network application programming interface, a cloud-based service that performs data-exchange transactions for end users, operative to cause the service to prevent end user access to incoming transactions so that the security manager accesses the incoming transactions prior to end users being able to access the incoming transactions, and a data inspector operative to inspect data of incoming transactions for security clearance by invoking one or more content scanners, wherein the security manager is further operative to cause the service to restore end user access to the incoming transactions, when the security inspector clears the data, and to cause the service to perform a remedial action vis-à-vis the incoming transactions, when the security inspector does not the data.
1000 MONITOR ONE OR MORE SERVICES THAT PROVIDES DATA TRANSACTIONS FOR END USERS

1010 ASCERTAIN SCOPE OF INSPECTION CONFIGURED BY ADMINISTRATOR

1015 DISCOVER IN REAL-TIME OCCURRENCE OF AN INCOMING OR AN OUTGOING TRANSACTION

1020 FOR INCOMING TRANSACTION: CAUSE END USER ACCESS TO TRANSACTION TO BE PREVENTED
FOR OUTGOING TRANSACTION: CAUSE TRANSMISSION OF TRANSACTION TO BE PREVENTED

1025 CAUSE RECIPIENTS TO BE ALERTED

1030 RETRIEVE DATA OF THE TRANSACTION

1035 INSPECT DATA OF THE TRANSACTION

1040 DATA PASSED INSPECTION?

1045 YES
MARK THE TRANSACTION WITH SPECIAL MARKER

1050 FOR INCOMING TRANSACTION: RESTORE END USER ACCESS TO THE TRANSACTION
FOR OUTGOING TRANSACTION: TRANSMIT THE TRANSACTION

1055 NO
CAUSE REMEDIAL ACTION TO BE PERFORMED VIS-À-VIS THE TRANSACTION

FIG. 5
FIG. 6

1100

1110

SET SPECIAL MARKER

ACTIVATE RULES FOR DATA
TRANSACTIONS, WITH:
- EXPIRATION AFTER SHORT TIME;
- (OPTIONAL) SETTING TO IGNORE DATA;
- TRANSACTIONS WITH SPECIAL MARKER;
- (OPTIONAL) FIRST PRIORITY, NO-STOP

1130

RULES EXPIRED?

1140

TIME TO CHANGE SPECIAL MARKER?

1120

NO

YES

NO

YES
OFFICE 365 ONE DRIVE DASHBOARD

Cloud Security Platform

Files Shared Across Co. 112
Folders Shared Across Co. 24
Externally Shared Files 37
Externally Shared Folders 15

Active Users 9
Applications 2
External Users 0
Incoming Files 0

Security Stack

Scanner #1 Scanner #2 Scanner #3 Scanner #4 Scanner #5 Scanner #6 Scanner #7 Scanner #8 Scanner #9 Scanner #10
Found 1 16 3 1 1 1 1 2 3 Total: 112
Total: 112 112 112 112

Users with Full Access to Files
user@email.com 2
user@email.com 1

Users with View Access to Files
user@email.com 48
user@email.com 29

FIG. 10
### OFFICE 365 EMAILS DASHBOARD

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Emails</td>
<td>16,531</td>
</tr>
<tr>
<td>Incoming Emails</td>
<td>16,322</td>
</tr>
<tr>
<td>Outgoing Emails</td>
<td>79</td>
</tr>
<tr>
<td>System Users</td>
<td>8</td>
</tr>
<tr>
<td>All Attachments</td>
<td>19,018</td>
</tr>
<tr>
<td>Incoming Attachments</td>
<td>28,727</td>
</tr>
<tr>
<td>Outgoing Attachments</td>
<td>158</td>
</tr>
<tr>
<td>Recent Attachments</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Security Stack

<table>
<thead>
<tr>
<th>Scanner #1</th>
<th>Scanner #2</th>
<th>Scanner #3</th>
<th>Scanner #4</th>
<th>Scanner #5</th>
<th>Scanner #6</th>
<th>Scanner #7</th>
<th>Scanner #8</th>
<th>Scanner #9</th>
<th>Scanner #10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Found</td>
<td>Found</td>
<td>Found</td>
<td>Found</td>
<td>Found</td>
<td>Found</td>
<td>Found</td>
<td>Found</td>
<td>Found</td>
<td>Found</td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>29</td>
<td>280</td>
<td>31</td>
<td>29</td>
<td>297</td>
<td>280</td>
<td>29</td>
<td>300</td>
</tr>
<tr>
<td>Total:</td>
<td>Total:</td>
<td>Total:</td>
<td>Total:</td>
<td>Total:</td>
<td>Total:</td>
<td>Total:</td>
<td>Total:</td>
<td>Total:</td>
<td>Total:</td>
</tr>
<tr>
<td>19018</td>
<td>19018</td>
<td>19018</td>
<td>19018</td>
<td>19018</td>
<td>19018</td>
<td>19018</td>
<td>19018</td>
<td>19018</td>
<td>19018</td>
</tr>
</tbody>
</table>

#### Top Internal Senders

<table>
<thead>
<tr>
<th>Domain</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain.com</td>
<td>4</td>
</tr>
<tr>
<td>domain.com</td>
<td>3</td>
</tr>
<tr>
<td>domain.com</td>
<td>2</td>
</tr>
<tr>
<td>domain.com</td>
<td>2</td>
</tr>
</tbody>
</table>

### FIG. 11
FIG. 12
FIG. 13
## MAIN DASHBOARD

### Alerts by Severity

<table>
<thead>
<tr>
<th>Category</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaaS</td>
<td>50,820</td>
</tr>
<tr>
<td>Total</td>
<td>2%</td>
</tr>
<tr>
<td>Box - Internal Users</td>
<td>7</td>
</tr>
<tr>
<td>Dropbox - Internal Users</td>
<td>0</td>
</tr>
</tbody>
</table>

### Alerts

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Description</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016-10-12 00:00:00</td>
<td>CC Number - Amazon (Symantec)</td>
<td>Good</td>
</tr>
<tr>
<td>2016-10-12 00:00:00</td>
<td>US Social Security Numbers - Egnyte</td>
<td>Good</td>
</tr>
<tr>
<td>2016-10-12 00:00:00</td>
<td>American Express Number - Egnyte</td>
<td>Good</td>
</tr>
<tr>
<td>2016-10-12 00:00:00</td>
<td>American Express Number - Egnyte</td>
<td>Good</td>
</tr>
<tr>
<td>2016-10-12 00:00:00</td>
<td>Password Files - Egnyte - Outgoing</td>
<td>Good</td>
</tr>
<tr>
<td>2016-10-12 00:00:00</td>
<td>High Severity - Egnyte Outgoing</td>
<td>Good</td>
</tr>
<tr>
<td>2016-10-12 00:00:00</td>
<td>Egnyte CheckPoint AV Outgoing</td>
<td>Good</td>
</tr>
<tr>
<td>2016-10-12 00:00:00</td>
<td>Egnyte CheckPoint AV Outgoing</td>
<td>Good</td>
</tr>
</tbody>
</table>

### DLP

<table>
<thead>
<tr>
<th>Category</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>1066</td>
</tr>
</tbody>
</table>

### Malicious Findings

---

### DLP Findings

---

**FIG. 14**
<table>
<thead>
<tr>
<th>Queries</th>
<th>Filter by</th>
<th>Actions</th>
<th>Name</th>
<th>Matched</th>
<th>Severity</th>
<th>Status</th>
<th>Scheduled Export</th>
<th>Security App.</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaaS</td>
<td>Entity</td>
<td>Type</td>
<td>File</td>
<td>2</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>CC Numbers Shared Ex.</td>
<td>34</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>Company Confidential..</td>
<td>44</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dropbox File</td>
<td>Company Confidential..</td>
<td>8</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Mail Att..</td>
<td>Company Confidential..</td>
<td>2</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Mail Mes..</td>
<td>Company Confidential..</td>
<td>0</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Mail Mes..</td>
<td>Company Confidential..</td>
<td>16</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>Company Confidential..</td>
<td>31</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attachment</td>
<td>Company Confidential..</td>
<td>11</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email Message</td>
<td>Company Confidential..</td>
<td>2</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Office File</td>
<td>Company Confidential..</td>
<td>20</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>Google All Files Test</td>
<td>215</td>
<td>Low</td>
<td>Running</td>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Policy Example - Data Leakage Scan / Protection

Benefits:
- Automation
- High accuracy
- Compliance
- Unified DLP Policy
- Full audit trail

Actions:
- Quarantine
- DRM-Protect
- Password Protect
- Online-only
- Consent page
- Unshare (External)

Options:
- Share File Internally
- Share File Externally

Corporate User
- External User

FIG. 17
Policy Example - Compliance Enforcement to SaaS Login

FIG. 19
Policy Example - Anomaly Detection

Cloud Security Platform Big-Data Analytics

Take Action

Cloud Security Platform

Antivirus Software

International Leakage:
- Massive download
- Export/Import Activity
- Massive sharing

Internal Threat:
- Massive Deletion
- Out-of-context activity
- Sensitive Actions

Admin Breach:
- Off-hour admin activity
- Geo-suspicious activity
- Sensitive config. change

Lost ID/Device:
- Device per user
- Reappearance of device
- Geo Activity

FIG. 20
Policy Workflow Example - Time Limit External Shares

Benefits:
- Automated data reclaiming
- End users are self-educated
- Policy compliance kept
- Full audit-trail

FIG. 21
Policy Workflow Example - External User Consent

Benefits:
- B2B compliance
- Full audit-trail

FIG. 22
Policy Workflow Example - External User Taking the Data Offline

Cloud Security Platform

Benefits:
- Secure sharing of sensitive info
- Maintaining compliance
- Control data after download
- Audit Trail

FIG. 23

External User

Corporate User
Policy Workflow Example - Control Forward Sharing

FIG. 24
Policy Workflow Example - Employment Termination

Cloud Security Platform

Automatic update when user employment terminates

Corporate User Directory

External User

Corporate User
Policy Workflow Example - Backup & Version Management

FIG. 26
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC - H04L12/26, H04L29/02 (2017.01)
CPC - H04L63/0281, H04L63/04, H04L63/20, H04L63/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document.

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 2012/0240183 A1 (SINHA, A) 20 September 2012; abstract; figures 5, 8, 17A; paragraphs [0026], [0027], [0064], [0066], [0070], [0074], [0080]; Claim 1</td>
<td>1,25</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:
  "A" document defining the general state of the art which is not considered to be of particular relevance
  "E" earlier application or patent but published on or after the international filing date
  "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  "O" document referring to an oral disclosure, use, exhibition or other means
  "P" document published prior to the international filing date but later than the priority date claimed
  "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
  "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
  "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
  "&" document member of the same patent family

Date of the actual completion of the international search: 2 February 2017 (02.02.2017)
Date of mailing of the international search report: 07 APR 2017

Authorized officer: Shane Thomas
PCT Helpdesk: 671.272.4300
PCT OSP: 671.272.7774

Form PCT/ISA/210 (second sheet) (January 2015)
# INTERNATIONAL SEARCH REPORT

**International application No.**

PCT/US16/64309

**Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)**

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. □ Claims Nos.:
   because they relate to subject matter not required to be searched by this Authority, namely:

2. □ Claims Nos.:
   because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. □ Claims Nos.:
   because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

**Observations where unity of invention is lacking (Continuation of item 3 of first sheet)**

This International Searching Authority found multiple inventions in this international application, as follows:

<table>
<thead>
<tr>
<th>Group</th>
<th>Claims</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1-23</td>
</tr>
<tr>
<td>II</td>
<td>24-25</td>
</tr>
</tbody>
</table>

**Remark on Protest**

□ The additional search fees were accompanied by the applicant’s protest and, where applicable, the payment of a protest fee.

□ The additional search fees were accompanied by the applicant’s protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☒ No protest accompanied the payment of additional search fees.

Form PCT/ISA/7.10 (continuation of first sheet (%)) (January 7015)
This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fee must be paid.

Group I: Claims 1-23 are directed towards preventing and restoring end user access to incoming transactions by inspecting security clearance.

Group II: Claims 24-25 are directed towards preventing transmission of outgoing transactions to their destinations by inspecting data leakage.

The inventions listed as Groups I-II do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

The special technical features of Group I include at least a cloud-based service operative to cause the service to prevent end user access to incoming transactions so that the security manager accesses the incoming transactions prior to end users being able to access the incoming transactions, inspect data of incoming transactions for security clearance by invoking one or more content scanners, and cause the service to restore end user access to the incoming transactions, which are not present in Group II.

The special technical features of Group II include at least causing a service to prevent transmission of outgoing transactions so that the security manager accesses outgoing transactions prior to transmission of the outgoing transactions to their destinations, inspecting data of outgoing transactions for data leakage, and causing the service to transmit the transactions to their destinations, which are not present in Group I.

The common technical features shared by Groups I-II are a data security system and method, comprising: a security processor remotely controlling, via a network application programming interface, a service that performs data-exchange transactions for end users; and a data inspector operative to inspect data; when said security inspector does not clear the data, perform a remedial action.

However, these common features are previously disclosed by US 2010/0125903 A1 to DEVARAJAN et al. hereinafter ("Devarajan"). Devarajan discloses a data security system and method, comprising: a security processor remotely controlling, via a network application programming interface, a service (virtual gateway node (security processor) managing tunnels for a security server (remote controlling) via an interface front end 130 through which users of the external systems can define security policies (network application programming interface); Abstract: Paragraph [0041]) that performs data-exchange transactions for end users (managing tunnels through which packets are transmitted (data-exchange transactions) for users; Paragraphs [0199], [0207]); and a data inspector operative to inspect data (data inspection engine (inspector); Paragraph [0033]); when said security inspector does not clear the data, perform a remedial action (based on the classification of content as a threat (does not clear the data), preclude distribution of the content item (remedial action); Paragraph [0034]).

Since the common technical features are previously disclosed by the Devarajan reference, these common features are not special and so Groups I-II lack unity.