Abstract:
A method and a system for preventing leakage of sensitive digital information on a digital communication network are provided. The method includes intercepting at a network gateway device a digital communication being sent from an internal network to an external network; extracting one or more context information items from the digital communication on the network gateway device; determining a security risk associated with the digital communication based on the one or more context information items; and sending an alert based on the security risk to at least one device connected to the internal network.
System and Method for Preventing Leakage of Sensitive Digital Information On A Digital Communication Network

Technical Field

[0001] Embodiments relate generally to a method and a system for preventing leakage of sensitive digital information on a digital communication network.

Background

[0002] Information Leakage Detection and Prevention ("ILDP") is an emerging and fast-growing area in the field of information security. The business drivers to prevent information leakage have existed since the Information Age. Due to the limitation of technological options in the past, organisations have been relying on measures with limited effectiveness, such as legal penalties. However, such measures are corrective in nature but do not prevent leakages from occurring. With information going digital and the growing prevalence of Internet access, the risk of sensitive corporate information / intellectual assets being leaked out poses a problem.

[0003] One common shortcoming of existing ILDP solutions is that they aim to protect every single valuable information, which leads to lengthy and laborious attempts to try to understand how every employee uses potentially sensitive information. Some ILDP solutions, especially those with client-side agents, require complex and time-consuming
installation and configuration. Other conventional solutions require users to copy sensitive information to centralised locations, resulting in interruption to business users.

[0004] In addition, organisations generally do not know the data context and hence are not able to create the relevant rules. The general approach of the other ILDP solutions makes this problem worse by requiring the organisations to understand the data context fully.

[0005] Most ILDP solutions do not possess context awareness and implement policies in a one-sided manner - by looking at the sender or source - without identifying who the recipients are. This further exacerbates the perception that ILDP obstructs, more than provide benefits to, business.

[0006] In addition, there is no existing ILDP solution that is able to detect information that is already leaked out to the Internet sites. With the increased popularity of Web 2.0 applications, the speed of spreading of information has increased, which makes timely discovery of public domain leakages more important.

[0007] Another shortcoming of the existing ILDP solutions is that there is no segregation of access to collected information from an administrator. This means all sensitive information that is captured by the ILDP system will be made available to the administrators.

[0008] Therefore, there is a need to provide a new method and system which overcome at least one of the above-mentioned problems.

Summary
In an embodiment, there is provided a method for preventing leakage of sensitive digital information on a digital communication network, the method including: intercepting at a network gateway device a digital communication being sent from an internal network to an external network; extracting one or more context information items from the digital communication on the network gateway device; determining a security risk associated with the digital communication based on the one or more context information items; and sending an alert based on the security risk to at least one device connected to the internal network.

In another embodiment, there is provided a system for preventing leakage of sensitive digital information on a digital communication network, the system including: a network gateway device that intercepts a digital communication being sent from an internal network to an external network, the network gateway device comprising a network connection to the internal network; a message store, that stores the digital communication; and a processor for evaluating a security risk of the digital communication based on context information associated with the digital communication the processor configured to: extract one or more context information items from the digital communication; and determine the security risk associated with the digital communication based on the one or more context information items; wherein the network gateway device is configured to send an alert to at least one device connected to the internal network, depending on the determined security risk.

Brief Description of the Drawings
In the drawings, like reference characters generally refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the various embodiments. In the following description, various embodiments are described with reference to the following drawings, in which:

[0012]Figure 1 shows a flowchart of a process for preventing leakage of sensitive digital information implemented in a digital communication network in accordance with an embodiment.

[0013]Figure 2 shows a schematic diagram of a system for preventing leakage of sensitive digital information implemented in a digital communication network in accordance with an embodiment.

[0014]Figure 3 shows a schematic diagram of a network gateway device of the system.

[0015]Figure 4 illustrates different types of rules of Type 1 correlation rules.

[0016]Figure 5 shows a flowchart of a process for determining a security risk of an inspected digital communication under a Type 2 correlation analysis.

[0017]Figure 6 shows a flowchart of a process for obtaining an actual identity of a sender of the digital communication.

[0018]Figure 7A shows an exemplary message including text to be determined whether there is source code included therein.

[0019]Figure 7B illustrates dividing the message of Figure 7A into segments.

[0020]Figure 7C shows an example text segment of the message of Figure 7A and the respective context lines of text.
[0021] Figure 8 shows a flowchart of a process for determining whether the message includes source code.

[0022] Figure 9 shows an exemplary piece of source code.

[0023] Figure 10 shows a table of elements extracted from a piece of source code being classified as unique identifying elements or generic elements.

[0024] Figure 11 shows a flowchart of a process for detecting leakage of sensitive source code on network-accessible sites.

[0025] Figure 12 shows an exemplary identity-relationship graph.

[0026] Figure 13 shows a flowchart of a process for tracing a source of leaked information owned by an organization after the information has been leaked on an external network.

[0027] Figure 14 shows a schematic diagram of a computer system.

**Detailed Description**

[0028] Exemplary embodiments of a method and a system for preventing leakage of sensitive digital information on a digital communication network are described in detail below with reference to the accompanying figures. It will be appreciated that the exemplary embodiments described below can be modified in various aspects without changing the essence of the invention.

[0029] Figure 1 shows a flowchart 100 of a process for preventing leakage of sensitive digital information implemented in a digital communication network. In 102, a digital communication being sent from an internal network to an external network may be intercepted at a network gateway device. In some embodiments, this may include
intercepting and decoding communications may be encoded or encrypted. For example, SSL (secure sockets layer)-encrypted network traffic may be intercepted and decrypted through the means of updating internal domain name system (DNS) records of external destinations and internally-generated SSL certificates may be provided by an internal certificate authority to user applications which have pre-trusted the internal certificate authority.

In [0030], one or more context information items, such as the time of the communication, whether the communication was encrypted, the intended recipient of the communication, or other information relating to the context of the communication may be extracted on the network gateway device. In [0031], a security risk associated with the digital communication may be determined based on the one or more context information items.

In [0031], the context information items may be matched against one or more predetermined rules on the network gateway device to determine the security risk. Matching the context information items against one or more predetermined rules may include matching the context information against a rule having one or more conditions, wherein the rule may be matched when any of the one or more conditions are met by the context information items. Further, matching the context information items against one or more predetermined rules may include matching the context information against a rule having one or more conditions, wherein the rule may be matched when all of the one or more conditions are met by the context information items. Matching the context information items against a rule having one or more conditions may further include matching against the one or more conditions a predetermined number of times greater
than one during a predetermined time window. Matching the context information against one or more predetermined rules may include matching the context information against a sequence rule that may include a plurality of sub-rules, wherein the sequence rule may be matched when the plurality of sub-rules have been matched in a predetermined sequence.

[0032] In 110, an alert may be sent based on the security risk to at least one device connected to the internal network. The alert may be sent if the context information items match at least one of the predetermined rules.

[0033] Figure 2 shows a schematic diagram of a system 200 for preventing leakage of sensitive digital information implemented in a digital communication network 202. The system 200 may have three components, namely a network gateway device 204, a management device 206 and a crawler server 208. In different embodiments, the system 200 may comprise different components and the number of components for the system 200 may also vary.

[0034] The network gateway device 204 may analyze the digital information transmitted over the network and may apply relevant policies to a digital communication. The network gateway device 204 may intercept the digital communication being sent from an internal network to an external network. The internal network may be a network controlled by an organization. The external network may be a network that is not controlled by the organization that controls the internal network. The external network may include but may not be limited to the Internet.

[0035] Figure 3 shows a schematic diagram of an embodiment of the network gateway device 204. The network gateway device 204 may include three parts, namely a correlation engine 302, a source code detection module 304 and a network traffic
analyzer 306. The network gateway device 204 may further include a network connection 308 to the internal network and a message store 310 that stores the digital communication. The correlation engine 302 may be implemented as a processor for evaluating a security risk of the digital communication based on context information associated with the digital communication. In some embodiments, the network gateway device 204 may also include a second network connection (not shown) to an external network, such as the Internet.

[0036]The correlation engine 302 implemented as a processor may be configured to extract one or more context information items from the digital communication, and to determine the security risk associated with the digital communication based on the one or more context information items. The correlation engine 302 may also be implemented as dedicated hardware, or as a software module, executing along with other software modules on a processor in the network gateway device 204. The network gateway device 204 may be configured to send an alert to at least one device connected to the internal network, depending on the determined security risk.

[0037]In different embodiments, the network gateway device 204 may have different parts and the number of parts of the network gateway device 204 may also vary.

[0038]The management device 206 of the system 200 may be a management and administration tool that can be used to control the network gateway device 204 and the crawler server 208, and to provide management reports. The system may comprise a plurality of the management devices 206 to provide scalability. The crawler server 208 of the system 200 may search Internet sites for leakages of information. The system 200
may provide the ability to control the digital communication of protected information, hence providing comprehensive protection to digital information assets.

Some features of the system 200 include but are not limited to instant protection of structured content such as source codes and personnel records, context-aware monitoring capabilities, detection of public leakages, and segregation of evidence access from administrators. Details of the above features of the system 200 are described below.

Instant protection of source codes and personnel records

The system 200 may provide instant protection against leakages of source codes and personnel records. The system 200 may include a source code detection module which has recognition algorithms for many or all popular programming languages. The built-in recognition algorithms can also detect obfuscated source codes and protect them from being leaked. Location-specific recognition algorithms may also be built in for similar protection of personnel records, as personnel records differ between different geographical locations. For example, an individual's identification record or number in Singapore is different from that in the United States.

Context-aware monitoring capabilities

The system 200 may include context-aware monitoring capabilities in the form of a correlation engine. Unlike other conventional technologies, the system 200 may perform a contextual correlation of the digital information within an organization's network or the digital information found in the public domain. The context-aware correlation engine may provide a more fine-grained control for the organization and may aim to protect, instead of obstructing, the business more effectively.

Detection of public leakages
The system 200 may include a crawler server 208, which may allow the system 200 to search, detect and monitor for leakage of valuable source codes or personnel records to popular Internet sites. The crawler server 208 may allow the organization to be informed of any public leakages, including those from outside the organization's network.

[0043] Segregation of evidence access from administrators

The system 200 may make use of an asymmetrical key method to prevent the administrators from accessing the collected evidence. The private key may be held by the business owners, who can be assured of the confidentiality of the evidence collected by the system. Administrators of the system may not be allowed to view the captured source codes that can be blocked from being leaked from the development team.

[0044] Details of the functions of the network gateway device 204, the management device 206 and the crawler server 208 of the system 200 are described in the following.

[0045] Management Device

The management device 206 may provide the centralized reporting and policy management for the system 200. The management device may include a management dashboard, a message bus, and an administration console. The management device 206 may have a policy distribution module for disseminating rules. Trust between each component of the management device 206 may be established using digital certificates.

[0046] Management Dashboard

The management dashboard may be provided for management reporting to business users and easy access to reports on the system 200. The management dashboard may also
support exporting of data via comma separated value (CSV) files to allow users to further customize their reports to their needs. The management dashboard may be web-based.

[0047] A replay function may be provided in the management dashboard to allow users to replay the leakage incidents that are captured. Only authorized business users can have full access to the replay function, which may require their private keys to view the full replay and content. For authorized users without the private key, only summary information may be displayed.

[0048] Message_bus

A message bus, which may be of enterprise-grade, can be used to collect information from the network gateway device 204 and the crawler server 208. Certificate-based mutual authentication may be used to prevent spoofing of any components in the system 200. The message bus may provide reliable and scalable transmission of information.

[0049] Administration Console

Besides the management dashboard, there may also be an administration console. The administration console may be web-based. The administration console may allow daily administration and operation of the system 200. The administration console may allow administrators to perform a plurality of administrative tasks. The administrative tasks may include but may not be limited to registering new components for certificate-based mutual authentication, configuring backup storage for archive, creating correlation rules, configuring alerts, configuring user sensitivity module, configuring user accounts for access to the administration console and the management dashboard, deploying patches to other components in the system 200, and configuring integration with existing Security Incident and Event Management products.
Another administrative task can be to generate a pair of public and private keys for encrypting/decrypting sensitive contents, e.g. email content, file attachments, IM conversations and etc. The generated private key may be stored on a separate medium under the custody of business owners. The private keys may be held temporarily in a volatile memory. The public keys may be stored in the system 200. The evidence collected may be encrypted with the public keys of the business owners who are authorized to review the evidence. To perform a full review of evidence, the private key may be required to be supplied by the business owners for decrypting the evidence for display. Administrators of the system may not be able to view the content of collected evidence as they do not have access to the private key.

Network Gateway Device

The network gateway device 204 of the system 200 may be used for detecting sensitive information and preventing leakages. It can be operated in a plurality of modes (e.g. two modes, namely monitoring mode and active protection mode). In the monitoring mode, a sniffer sub-component of the network gateway device 204 may be activated and can capture digital communication from within a network hub, a network tap or a span port of a core switch to capture the digital communication.

In the active protection mode, the sniffer sub-component may be deactivated. ICAP (Internet Content Adaptation Protocol) and MTA (Mail Transfer Agent) server components may be activated to receive the digital communication from proxies and email servers. These components may support integration with existing enterprise proxies and email servers.
Detector components for instant messaging, web-based instant messaging, voice/video-over-IP, P2P traffic, email, web-based email and other proprietary traffic sent over the Internet, and instant messaging, email and other non-HTTP/HTTPS traffic sent over HTTP or HTTPS, and other HTTP/HTTPS traffic, may be activated in both monitoring and active protection modes to detect protected content in the digital communication network and to provide analysis against contextual correlation rules as configured in the management device 206. The protected structured content may be automatically detected against a detection engine consisting of heuristic recognition patterns. The protected structured content may be automatically detected even when the content is obfuscated, scrambled or compressed.

Correlation Engine

As shown in Fig. 3, the network gateway device 204 may include a correlation engine 302 for evaluating the security risk of the digital communication based on context information items associated with the digital communication. The digital communication may be stored in a message store 310 of the network gateway device 204 and at least a subset of the one or more context information items may be stored in a historical data store (not shown). The network gateway device 204 may further include a source code detection module 304 for detecting source codes in the digital communication sent, and a network traffic analyzer 306 for obtaining an actual identity of a sender of the digital communication. The network gateway device 204 may also include a network connection 308 to the internal network.

The context information items associated with the digital communication may include but may not be limited to a time at which the digital communication is sent; a size
of information in the digital communication; a type of information contained in the
digital communication; a source of the digital communication (e.g. source IP, hostname
and etc); an identity of a sender of the digital communication; a sensitivity of the sender
of the digital communication; an intended destination for the digital communication (e.g.
destination IP, hostname and etc); an identity of an intended recipient of the digital
communication; whether the digital communication is encrypted (e.g. decipherability of
information sent); and whether the digital communication contains digital rights-
protected content.

[0056] In different embodiments, the number of context information items may be
different. The context information items used for Type 1 correlation analysis may be
different in other embodiments.

[0057] The correlation engine 302 may be designed to support a plurality of specific types
of communication analysis (e.g. two types, namely Type 1 and Type 2) to determine the
security risk associated with the digital communication. The correlation engine 302 may
support different numbers of specific types of communication analysis in other
embodiments.

[0058] The first kind of correlation analysis (Type 1) may be based on a set of
predetermined rules to identify communication links that breach one or more
predetermined rules. The identified communication link may be managed as a security
breach incident. This correlation analysis may be applied to real-time traffic inspection.

[0059] The second kind of correlation analysis (Type 2) may be based on a probabilistic
formula and risk rules to identify non-incident communication links that have high level
risk of information leakage. This correlation analysis may be used on demand.
In Type 1 correlation rule construction, an administrator may be able to construct a plurality of types of predetermined rules (e.g. five types) using the management device 206. The predetermined rules may then be sent to the network gateway device 204 for determining security risks. The five types of predetermined rules are described in the following and are illustrated in Figure 4.

A simple rule 402 may be made up of one or more conditions. Users can use an "OR" relationship 404 to indicate that the simple rule 402 may be matched when any one of the conditions is met by the context information items or an "ALL" relationship 406 to indicate that the simple rule 402 may be matched when all ("ALL") the conditions are met by the context information items. For each condition within a simple rule 402, users can define the criteria that the condition should match.

For example, users can make use of multiple conditions to specify that an alert is to be sent for any traffic sent on weekday AND later than 18:00 OR any traffic sent on weekend or holidays.

An aggregate rule 408 may be made up of a single simple rule 402. An aggregate rule may be further defined by a group 410 consisting of a duration window, event count and a "group by" parameter. For example, users can define a duration window of 24 hours, event count of 3 and grouped by the same source user in an aggregate rule 408 using the simple rule 402 described above. With this configuration, alerts may be sent only when the specified simple rule 402 is matched 3 times in a day for the same sender, instead of every occurrence of the simple rule 402.
A composite rule 412 may be made up of multiple simple rules 402. The multiple simple rules 402 may belong to an "ALL" relationship such that all simple rules 402 within a composite rule 412 are matched to trigger the composite rule 412. A composite rule 412 may be further defined by a group 410 consisting of a duration window, event count and a "group by" parameter, similar to those found in an aggregate rule 412. For example, users can also define a composite rule 412 consisting of a simple rule 414 which checks the total amount of source codes detected exceeds 200KB, together with one or more other simple rules 402. The duration window is 24 hours, event count is 3 and grouped by same department. The composite rule 412 may be triggered when 3 digital communication, each containing more than 200KB of source codes are sent by any members of the same department after office hours in a day. With this configuration, alerts may be sent only when there are 3 occurrences of a single digital communication that matches both the specified simple rules 402 and 414 in a day within the same department, instead of every occurrence of the simple rule 402 or 414.

A sequence rule 416 may be made up of multiple simple/aggregate/composite rules (402, 408, 412) and has a group 410 consisting of a duration window, event count and a "group by" parameter. It may include an additional criteria 418 which defines the order at which each of the simple/aggregate/composite rule (402, 408, 412) is matched. The sequence rule 416 may only be triggered when the order of all simple/aggregate/composite rules (402, 408, 412) is matched.

A free form/custom rule 420 may be defined by a process 422 of entering the full programming script codes based on our pre-defined script syntax to systematically create a simple/aggregate/composite/sequence rule (402, 408, 412, 416). This option may allow
greater flexibility and usage of certain pre-defined functions not available from the other rules including, but not limited to "UNION", "INTERSECTION" and "GATE".

[0067] In Type 1 correlation analysis, each communication link may be analyzed independently against all active simple rules 402. Multiple incidents may also be analyzed collectively to determine if there is a pattern match against one or more aggregate/composite/sequence rules (408, 412, 416). In different embodiments, different number of types of predetermined rules may be constructed for Type 1 correlation analysis.

[0068] **Type 2 correlation analysis.**

In Type 2 correlation analysis, the correlation engine 302 may depend on a pre-defined probabilistic formula and a set of a plurality of context information items to identify high security risk communication links between an internal identity and his/her contacts. A high security risk communication link may be defined as a communication link that did not trigger any incident rule of the Type 1 correlation analysis, but may be likely to contain sensitive information leakage based on context and degree of variation from the sender's historical patterns during a specific time period. The security risk may be determined by the correlation engine 302 based on at least in part on the data from previous communications. The data from previous communications may be stored in a historical data store. The security risk may be determined by the correlation engine 302 based on past recorded context information associated with the sender of the digital communication.

[0069] The context information items may include but may not be limited to time at which the digital communication is sent; a size of information in the digital communication; a
type of information contained in the digital communication; a source of the digital communication (e.g. source IP, hostname and etc); an identity of a sender of the digital communication; a sensitivity of the sender of the digital communication; an intended destination for the digital communication (e.g. destination IP, hostname and etc); an identity of an intended recipient of the digital communication; whether the digital communication is encrypted (e.g. decipherability of information sent); and whether the digital communication contains digital rights-protected content.

[0070] In different embodiments, the number of context information items may be different. The context information items used for Type 2 correlation analysis may be different in other embodiments.

[0071] To determine the sensitivity of the sender of the digital communication, the correlation engine 302 may depend on a list of user-defined inputs entered by the management device 206. The user-defined inputs may include but may not be limited to the involvement of the sender in sensitive projects within the organization; a last day of work for the sender; and preference of a supervisor of the sender.

[0072] To determine a type of information contained in the digital communication, the source code detection module 304 of the network gateway device 204 may determine whether the digital communication contains source code.

[0073] Since the security risk of the digital communication may be determined based on past recorded context information, a time period may be used to define the set of past recorded context information to be used for computing the degree of variation of the inspected digital communication against the set of past recorded context information belonging to each user.
From the set of past recorded context information as defined by the time period, the correlation engine 302 may obtain a plurality of mode values (e.g. top ten mode values) for each context information item. In different embodiments, the number of mode values for each context information item may be different. A plurality of mode values may be determined based on the information in the historical data store. Each mode value may represent a frequency with which a predetermined condition occurs in the data from previous communications in the historical data store.

Each context information item may be given a different weight score and a coefficient value may be determined for each context information item based on a corresponding mode value. Both the weight score and the coefficient value may be used to determine a total risk score. An example illustration of determining the risk score is described below.

The correlation engine 302 may compute e.g. the top ten mode values for the sender in terms of e.g. the hour value of sent time of the digital communication. The 1st mode value, e.g. most frequently used hour, is given a mode score of 1, whereas the 10th mode value is given a mode score of 10. The coefficient value is derived based on the formula below:

\[
\text{Coefficient value} = \text{mode score} - 1
\]

A coefficient value may range from 0 to 9 in the event that ten mode values are used. The range of the coefficient values may vary in different embodiments. The coefficient value may include a value, e.g. 20, for contextual variable values that lie
outside the top ten mode values. In different embodiments, the coefficient value for contextual variable values that lie outside the top ten mode values may be different.

[0078] If the sent hour of the digital communication matches the 1st mode value, a value of 0 may be assigned as the coefficient value. If the sent hour of the digital communication matches the second mode value, a value of 1 may be assigned as the coefficient value. If the sent hour does not match any of the top ten mode values, a value of 20 may be assigned as the coefficient value.

[0079] After the coefficient values of all the context information items are assigned, the correlation engine 302 may calculate the risk score of the inspected digital communication using the following formula:

\[
\text{Risk Score} = (\text{CV 1 weight score} \times \text{coefficient 1}) + (\text{CV 2 weight score} \times \text{coefficient 2}) + \ldots + (\text{CV 10 weight score} \times \text{coefficient 10})
\]

[0080] If the risk score is below a risk threshold value, the current inspected digital communication may be deemed to be low risk. If the risk score matches or exceeds the threshold value, the current inspected digital communication may be deemed to be high risk and appropriate actions may be applied. The risk threshold value may be adjustable for individual organizations.

[0081] Figure 5 shows a flowchart 500 of a process for determining the security risk of the inspected digital communication under the Type 2 correlation analysis. In 502, at least a subset of the one or more context information items may be stored in a historical data store. The security risk may be determined based at least in part on the data stored
from previous communications stored in the historical data store. In 504, a plurality of
mode values may be determined based on the information in the historical data store. In
506, a plurality of risk coefficient values may be determined based on the plurality of
mode values. In 508, a risk score may be determined based on the plurality of risk
coefficient values and the context information.

[0082]In 510, the risk score may be checked to determine if the risk score is below a risk
threshold value or matches or exceeds the risk threshold value. If the risk score is below
the risk threshold value, the inspected digital communication may be deemed to be low
risk in 512. If the risk score matches or exceeds the risk threshold value, the inspected
digital communication may be deemed to be high risk in 514 and appropriate actions, e.g.
sending an alert to at least one device connected to the internal network, may be applied.

[0083]An example scenario is described below to illustrate how the context-aware
correlation engine may operate in real-life.

[0084]Through the correlation rule editor for Type 1 correlation analysis, the
administrator may configure the system such that each developer can be allowed to email
small bits of their code, for example up to 1 KB, to authorized technical advisers over a
period of one week. Posts to developer forums can be subjected to more restrictive
controls such as e.g. 500 bytes per week. Posting via instant messaging can be prohibited.

[0085]Type 2 correlation analysis may start when a developer, who has been posting e.g.
no more than 200 bytes of source codes per week to SITE A during office hours for the
past 1 month, begins to send e.g. 300 bytes of source codes to SITE B after office hours.

[0086]For both Type 1 and Type 2 correlation analysis, when the inspected digital
communication is deemed to be high risk, an alert may be sent to at least one device
connected to the internal network. The network gateway device 204 may be configured to send the alert. An alert message that causes a device on the internal network to log the sending of the digital communication may be sent. The management device 206 may be configured to receive the alert message, and to display the alert message to a user. An alert message and information on the content of the digital communication may also be sent to the management device 206 on the internal network. The information on the content of the digital communication may be encrypted so that the information cannot be viewed by a system manager without authorization from management. A public key encryption algorithm may be used to encrypt the information. Further, the digital communication may be quarantined by storing it on a device on the internal network. Sending of the message to the external network may also be blocked. The network gateway device 204 may be configured to block the sending of the message.

Network Traffic Analyzer

As discussed above with reference to Fig. 3, the network gateway device 204 may include a network traffic analyzer 306. The network traffic analyzer 306 may integrate with enterprise directory and Dynamic Host Configuration Protocol (DHCP) servers to obtain the real user identities from the captured internet protocol (IP) addresses or machine hostnames. Further, the reporting hierarchy, i.e. the reporting officer, for each user may also be extracted from the enterprise directory. This may facilitate the automatic escalation of detected incidents to the appropriate supervisor. The network traffic analyzer 306 may include an identity resolution module (not shown) which obtains the identity of the sender from the source IP address and content of the captured digital communication.
Figure 6 shows a flowchart 600 of a process for obtaining the actual identity of the sender of the digital communication. Such a process may be used, for example, by the network traffic analyzer 306 discussed above with reference to Fig. 3. In 602, "User Name" may be obtained from e.g. a MICROSOFT® ACTVE DIRECTORY® technology Windows Event ID 672 with the source IP address of the captured digital communication. If "User Name" is not found in 602, the digital communication may be checked in 604 to determine if it is of email type, regardless whether the digital communication is native or web-based. If the digital communication is of email type, user identity may be obtained in 606 by matching the extracted sender's email address against an existing Identity-Relationship Database.

If the digital communication is not of email type, the digital communication is checked in 608 to determine if it is of instant messaging type, regardless whether the digital communication is native or web-based. If the digital communication is of instant messaging type, user identity may be obtained in 610 by matching the extracted sender login ID against the existing Identity-Relationship Database. If the digital communication is not of instant messaging type, the digital communication may be treated as from "Unknown" user in 612 and the relevant correlation rules may be applied accordingly.

Source Code Detection Module

As discussed above with reference to Fig. 3, the network gateway device 204 may include a source code detection module 304. The source code detection module 304 may detect source codes. The source codes may be in plain text or binary documents such as a MICROSOFT® VISUAL STUDIO® integrated development environment project file or MICROSOFT® WORD® word processor document. The source code detection module
may allow users to enjoy instant protection without the tedious effort of building the complex rules for each programming language.

Figure 7A shows an exemplary message 700 including text to be determined whether there is source code included therein. The message 700 may be divided into one or more segments, each segment including a predetermined number of lines. A sliding window offset may be used to cause the starting line of each segment to be varied during multiple passes over the text to determine whether source code is present. Figure 7B illustrates dividing the message 700 into segments.

Each segment 710 may include a predetermined number of lines of text from the message 700. The number of lines of text of each segment 710 may be represented by a configurable parameter "segment_size". In this example, the segment_size is set to be 4.

The message 700 may be divided into a plurality of segments 710 by using a sliding window 720. Since the segment_size is set to be 4, the size of the sliding window 720 may be 4 lines of text so that each segment 710 may includes 4 lines of text from the message 700.

When applying one or more syntax rules of the programming language to each segment 710 to determine if there is source code, the context of the segment 710 may be taken into account, by examining lines of text immediately before and after the segment 710.

Figure 7C shows an example text segment 730 of the message 700 and the respective context lines of text 740. The context of the segment 730 may be referred as the lines of text 740 before the segment 730 and/or after the segment 730 in the message 700. The context of the segment 730 may include a predetermined number of context
lines of text 740. The predetermined number of context lines may be represented by a configurable parameter "context_size". In this example, the context_size is set to be 2.

[0096]When applying one or more syntax rules of the programming language to each segment, the one or more syntax rules are applied to the selected segment 230 together with its context lines of text 240, which may help to increase the accuracy of the determination of the existence of source code.

[0097]If source code is determined to be included in a text segment 730 and its context lines 740, contents of the text segment 730 may be stored in a memory together with the previous text segments 730 which are determined to include source code. The total size of source code detected in the message 700 may be determined based on the total size of the detected source code in the respective text segments 730.

[0098]When applying one or more syntax rules of the programming language to each segment, the syntax rules may be applied to the selected segment 730 together with its context lines of text 740, which may help to increase the accuracy of the determination of the existence of source code.

[0099]A probability value P may be used to represent the probability of an inspected text segment of a message containing source code. P may be a value between 0 and 1. The probability of the inspected text segment containing source code may increase as P approaches 1, and may reduce as P approaches 0.

[0100]A contributing factor to the value of P may be the extent to which the inspected text segment matches one or more syntax rules of a particular programming language. A coefficient may be provided to each syntax rule of each programming language. The
coefficient may indicate the weightage of the matched syntax rule in determining the value of $P$. The coefficient may be a whole number greater than 0.

[0101] The coefficient for a syntax rule may be provided based at least in part on the uniqueness of that syntax rule in its programming language. For example, the more unique the syntax rule is to the particular programming language, the higher the coefficient value is. In different embodiments, the coefficient for a syntax rule may be provided based on other factors, such as the importance of the syntax rule in its programming language, etc. This may help to increase the speed and accuracy of detecting source code in a particular programming language.

[0102] The one or more syntax rules may be applied to each segment in an order based on their coefficients. The syntax rules may be applied to each segment in an order from highest coefficient to lowest coefficient. The highest coefficient may represent the high importance or uniqueness of the corresponding syntax rule, so that that syntax rule may be applied first. The syntax rules may be applied to each segment in other orders, which may be defined by an administrator in different embodiments.

[0103] Taking into consideration the importance of speed for source code detection, the administrator may select the preferred programming languages so that the relevant syntax rules for the preferred programming languages can be checked against the message first. Syntax rules for unselected programming languages may only be checked against the message after the selected programming languages. The order at which the unselected programming languages are inspected may be pre-defined in the system from the market surveys on the popularity of each programming language. Similarly, the order at which the selected programming languages are inspected may also be pre-defined in the system
from the market surveys on the popularity of each programming language. The pre-defined orders can be updated through regular system updates.

[0104]For each programming language, a program thread may be created to inspect all the segments of the message based on the relevant syntax rules. Within the thread, the inspected message may be checked against the syntax rules of a particular programming language in a descending order of coefficient values of the syntax rules. Accordingly, syntax rules which provide the highest confidence level that the inspected message contains source codes of the particular programming language may be applied and checked against the segments of the message first.

[0105]When a syntax rule is matched with a segment, a product of the coefficient for the syntax rule and the number of characters of text in the segment that match this syntax rule may be determined. This product may be referred to as a weighted size of the matched characters in the segment and may be denoted as "weighted_size". The weighted_size may be determined using the following formula:

\[
\text{weighted_size} = \text{weighted_text_length} \times \text{coefficient}
\]

wherein \( \text{weighted_text_length} \) is the number of characters that match the syntax rule and \( \text{coefficient} \) is the corresponding coefficient value of the matched syntax rule.

[0106]The value of \( P \) may be determined based on the product for each syntax rule described above and the number of characters of text in each segment. Accordingly, for each matched syntax rule, a corresponding product value "weighted_size" is determined. The product values "weighted_size" for the matched syntax rules may be summed up to

27
determine a cumulative weighted size of the matched characters in the segment. The cumulative weighted size may be denoted as "cumulative_weighted_size" and may be determined by summing up all the "weighted_size" values of the current text segment. The "cumulative_weighted_size" may represent a scaled value of the cumulative size of the matched characters in the segment, based on the coefficients of the matched syntax rules.

[0107] The cumulative size of the matched characters in the segment, which represents the cumulative number of the matched characters in the segment, may also be determined. The cumulative size of matched size of the matched characters may be denoted as "cumulative_weighted_text_length" and may be determined by summing up all the "weighted_text_length" values of the current text segment.

[0108] The value P for each segment may be determined using the following formula:

\[
P = \frac{\text{cumulative_weighted_size}}{\text{total_number_of_characters} + \text{cumulative_weighted_size} - \text{cumulative_weighted_text_length}}
\]

where

cumulative_weighted_size represents the cumulative value of the product of each matched syntax;

cumulative_weighted_size represents the cumulative value of the number of matched characters of text of each syntax; and

total_number_of_characters represents the total number of characters of text in the segment.
For example, in a text segment including 300 characters, 100 characters match with a first syntax rule having a coefficient 20, and 50 characters match with a second syntax rule having a coefficient 10. Then,

\[
\text{cumulative \_weighted \_size} = (100 \times 20) + (50 \times 10)
\]

\[
\text{cumulative \_weighted \_text \_length} = 100 + 50
\]

\[
\text{total \_number \_of \_characters} = 300,
\]

According, the value of \( P \) is computed as follows:

\[
P = \frac{(100 \times 20) + (50 \times 10)}{300 + ((100 \times 20) + (50 \times 10) - (100 + 50))} = 0.943396226
\]

Based on the value of \( P \), it may be determined whether the segment and accordingly the message include source code or not. The value of \( P \) may be compared with a predetermined threshold value \( T \) of a programming language to determine the existence of source code. The threshold value \( T \) may be configurable and may range between 0 and 1. If the ratio value exceeds the threshold value, it may be determined that the source code for a particular programming language is present in the message, and the relevant action, such as sending an alert message to a device on the network, may be taken.

Each programming language may have an independent threshold value \( T \). The threshold value \( T \) may be updated through regular system updates or manual modification. The threshold value \( T \), which is configured manually, may not be subjected to automatic updates. In this embodiment, the ratio value may be determined after all the syntax rules of a programming language have been applied to the segment.
In another embodiment, only some of the syntax rules may be applied to the segment before the value of $P$ is determined. A text segment may not be checked against all the syntax rules of a particular programming language. When the value of $P$ exceeds the threshold value $T$ for the particular programming language, the inspection of current text segment may be stopped. The next text segment of the message may then be inspected.

The detected size of source code for each programming language may be computed by summing up the number of characters identified as belonging to the particular programming language from all collected text segments. In the event that multiple programming languages are detected, the source code detection module 304 may inform the correlation engine 302 and the correlation engine may apply the relevant correlation rules as described above.

Figure 8 shows a flowchart 800 of a process for determining whether a message includes source code. In 802, a message may be intercepted on a network device and may be placed into a memory on the network device. In 804, the message may be divided into one or more segments, wherein each segment may include a predetermined number of lines of text from the message. In 806, a coefficient may be provided for each of one or more syntax rules of a programming language, wherein each coefficient may be based at least in part on the uniqueness of the syntax rule. In 808, for each segment, the one or more syntax rules may be applied in an order based on their coefficients to determine whether the segment matches the syntax rule. In an embodiment, the order may be from highest coefficient to lowest coefficient.
In 810, a product of the coefficient for the syntax rule and the number of characters of text in the segment that matches the syntax rule may be determined. In 812, a ratio may be determined based on the product for each syntax rule and the number of characters of text in each segment. In 814, it may be determined whether the ratio exceeds a threshold value. If the ratio does not exceed the threshold value, the next syntax rule of the programming language may be applied to the segment as shown in 808. If the ratio exceeds the threshold value, the application of syntax rules may be stopped and a determination of whether the message includes source code may be provided in 816.

[Crawler Server]

As discussed above with reference to Fig. 2, the crawler server 208 of the system 200 may provide active monitoring and detection of leakages to the external network. The crawler server 208 may operate by automatically logging into one or more of network-accessible sites and performing search-and-filter activities. These network-accessible sites may not be accessible to popular search engines. These network-accessible sites may be designated by a user of the system 200.

The search-and-filter activities performed by the crawler server 208 may be broken down into a plurality of phases (e.g. two phases). An initial search phase may be performed to list out a summary of results ranked in order of relevance. Users can then review the summary results and instruct the crawler server 208 to perform a more in-depth search of the selected initial results. Wherever possible, multiple search functions offered by the designated Internet sites may be utilized by the crawler server 208 to
provide more accurate and comprehensive searches. The above activities can be performed on demand by the administrators or as scheduled.

[0119] Inputs to the online search can be manually entered or automatically derived by the crawler server 208 after accessing protected information repositories and evaluating the protected content. For example, the crawler server 208 can automatically access a source code repository of an organization, extract the source codes, obtain unique identifying elements of the extracted source codes and perform searches using the unique identifying elements.

[0120] An exemplary piece of source code 900 named `GeneralUtil.java` is shown in Figure 9. The exemplary source code 900 is used for illustrating the detailed process of obtaining unique identifying elements.

[0121] Initially, elements may be extracted from the source code 900. The elements extracted from the source code 900 may be categorized into a plurality of element types. The element types may include:

One-line comments;
Declared Package names (for programming languages which support this);
Method names;
Class names; and
File names.

Different element types may be used for categorizing the elements extracted from the source code in different embodiments. The number of element types may also be different in other embodiments.
Next, each of the elements extracted from the source code 900 may be checked, to determine whether it is an unique identifying element, using uniqueness rules. The uniqueness rules may include:

a) Length of the element; and

b) Whether the element is included in a blacklist of common/generic words.

Different uniqueness rules may be used in different embodiments. The number of uniqueness rules may also be different in other embodiments.

Either one uniqueness rule or a combination of uniqueness rules may be applied to each element type. For example,

1. The uniqueness rule "Length of the element" may be applied to the element type "One-line Comments".

2. The uniqueness rule "Length of the element" may be applied to the element type "Declared Package Names", starting (in some embodiments) with a hierarchy of 2 levels, e.g. "com.mycompany". An example element extracted from the source code 900 is "insightcommon".

3. The uniqueness rule "Length of the element" may be applied to the element type "Method Names". The elements categorized under the element type "Method Names" may also be compared to the blacklist of common/generic words.

4. The uniqueness rule "Length of the element" may be applied to the element type "Classes Names". The elements categorized under the element type "Classes Names" may also be compared to the blacklist of common/generic words.

5. The uniqueness rule "Length of the element" may be applied to the element type "File Name". The elements categorized under the element type "File Name" may also be
compared to the blacklist of common/generic words.

Figure 10 shows a table 1000 of elements extracted from the source code 900 classified as unique identifying elements or generic elements. Column 1002 shows the various element types, column 1004 shows the elements determined as generic, and column 1006 shows the elements determined as unique identifying elements.

Row 1008 shows elements, e.g. "this is my comment for the interestingMethodAction" and "Gets today's date", categorized the element type "One-line Comments" determined as unique identifying elements. Row 1010 shows elements, e.g. "insight.common" and "insight.common.util", categorized the element type "Declared Package Names" determined as unique identifying elements. Row 1012 shows an element, e.g. InterestingMethodAction, categorized the element type "Method Names" determined as an unique identifying element. These elements may have a length above a predetermined length threshold if the uniqueness rule "Length of the element" is applied.

By applying the uniqueness rule "Length of the element", Elements such as "getID" and "setID", having a length below a predetermined length threshold may not be determined as an unique identifying element. Elements having a length below a predetermined length threshold may be excluded to improve the accuracy of the search and to reduce false positives.

Row 1012 also shows an element, e.g. GetCurrentDate, categorized the element type "Method Names" determined as a generic element. Row 1014 shows an element, e.g. GeneralUtil, categorized the element type "Classes Names" determined as a generic element. Row 1016 shows an element, e.g. GeneralUtil, categorized the element type "File Name" determined as a generic element. These elements may be found in the
blacklist of common/generic words, and will therefore not be determined to be "unique" if the uniqueness rule applying the blacklist is applied.

[0128] When all the unique identifying elements are obtained, the crawler server 208 may proceed to perform searches with a plurality of combinations of the unique identifying elements. Searches may be performed in a descending order of relevance, starting with the highest relevance, i.e. matches to all unique identifying elements. The crawler server 208 may perform searches starting from the more relevant element type "One-line comments" to the less relevant element type "File names". There can be e.g. thirty-one types of combination searches from the e.g. five elements types that the crawler server 208 analyzes.

[0129] The thirty-one types of combination searches are listed in the following:

*Types of Combinations:*

1st: All One-line Comments +All Packages +All Methods + All Classes + File name

= Highest relevance

2nd: 0 One-line Comments +AH Packages +All Methods +All Classes + File name

3rd: 0 One-line Comments + 0 Packages +All Methods +All Classes + File name

....

91st: 0 One-line Comments + 0 Packages + 0 Methods + 0 Classes + File name =

Least relevance

[0130] After a specific combination search is completed, the next unique identifying element in the same element type may be used for the subsequent combination search. To reduce the number of results, the user may configure a limit to the maximum number of results returned from each combination search.
After the search results are obtained, they may be ranked in a descending order of relevancy. Relevancy may be computed using the following formula:

Relevancy value = CombinationPoints / TotalSearchResults

where

CombinationPoints = (One-Line Comment * Points per comment) + (Declared Package Name * Points per package) + (Method Name * Points per method) + (Class Name * Points per class) + (File Name * Points per filename)

and

TotalSearchResults = the number of results retrieved when searching using that combination.

CombinationPoints may be divided by TotalSearchResults to provide higher weightage to combinations that result fewer results, i.e. more unique. For example:

Case 1: Calculation for a combination search using one Class Name which returns 100 records

Relevancy value = [(O * 25) + (0 * 18) + (0 * 19) + (1 * 10) + (0 * 2)] / 100 = 0.1

Case 2: Calculation for a combination search using one File Name which returns 1 record

Relevancy value = [(O * 25) + (0 * 18) + (0 * 19) + (0 * 10) + (1 * 2)] / 1 = 2

In this example, the result of Case 2 is ranked higher in terms of relevancy than the result of Case 1 although Case 1 uses a more relevant element type.
Figure 11 shows a flowchart of a process for detecting leakage of sensitive source code on network-accessible sites. In 1102, a set of unique identifying elements that identify a sensitive source code module accessed from a source code repository may be determined. In 1104, a crawler server connected to an external network to automatically search a list of one or more network-accessible sites for text that matches one or more of the unique identifying elements, may be used to provide search results. In 1106, the search results may be collected in a memory of the crawler server, in 1108, a relevancy for each of the search results may be determined based at least in part on a number of the unique identifying elements that were matched and on a number of search results. In 1110, the results may be sorted according to the relevancy. In 1112, the results may be provided to a user, to indicate whether sensitive source code was found on the network-accessible sites.

Identity-Relationship Database

The system 200, as described above in Fig. 2, may also provide traceability of source of leakages of information by building a web of relationships between online identities that leak the confidential information to internal user identities (e.g. building an identity-relationship database). The identity-relationship database may be built by integrating identities collected by the network traffic analyzer 308 (as shown in Fig. 3) of the network gateway device 204 and identities collected by the crawler server 208. The network traffic analyzer 308 of the network gateway device 204 may collect identities captured from within the organization (e.g. the internal network) while the crawler server 208 may collect identities captured from the external network (e.g. Internet sites). This
capability may be used for identifying sources of leakages to popular Internet sites such as blogs, social networking sites and forums.

[0136] Internal network identities may be collected from internal sources within the organization. For example, network identities which are of interest to be extracted may include the identities used by employees for accessing instant messaging, personal email, forums, social networking sites, blogs and other Web-based services (including, for example, Web 2.0 services). The collected internal network identities may be linked to their respective users, i.e. the employees. The collected internal network identities may also be linked to the internal corporate identities by resolving against the organization's directory and network servers.

[0137] Counter-part identities of intercepted traffic, e.g. intercepted message, may also be captured. These counter-part identities may be network identities of intended recipient(s) of the message, which may be identities within the internal network or identities in the external network. The network identities of the intended recipient(s) of the message may be considered as the first layer of friends of the internal user in the identity-relationship database. Building of the identity-relationship database may include recording the frequency of communication between the network identity of the message sender and the intended recipient(s).

[0138] When the crawler server 208 detects leakages of protected information, of which the process is described above, the crawler server 208 may capture a profile of the poster of the digital information and profiles of all parties related to the poster. The profiles may include network identities of the users, and may be matched partially or completely when determining the source of the leaked information.
The online identity of the person leaking the information may be captured. If the online identity is present in the identity-relationship database, the possible source(s) of leakage can be traced immediately. If the online identity is not present in the database, the crawler server may attempt to build the second and subsequent "layers of friends" for the online identity in question. Sources of information to build the "layers of friends" may include but may not be limited to social networking sites, blog sites, discussion forum sites, other sites that permit posting of messages and contents, external e-mail and instant messaging sites. When there is a match between the "layers of friends" for the online identity who leaked the protected information and those for internal users, the "layers" may be merged and the online identity may be linked to the relevant internal users. If there is still no match beyond a threshold number of "layers", the closest yet not linked layers may be shown to the administrator for manual evaluation and judgement.

Figure 12 shows an exemplary identity-relationship graph. A target unknown identity 1202 may be linked to its first layer of friends 1204 and second layer of friends 1206. The target unknown identity 1202 may be an internal identity or an external identity. In this example, the target unknown identity 1202 is the network identity of a poster of the leaked information on the external network. An internal network identifier 1208 may be linked to the target unknown identity 1202. By identifying the link from the target unknown identity 1202 to the internal network identifier 1208, the internal network identifier 1208 may be identified as a possible source of the leaked information.

Building of the identity-relationship database may further include determining a closeness of a connection between a first network identity and a second network identity. Determining a closeness of a connection may include determining a type of a detected
relationship, wherein each type of detected relationship may be associated with a proximity value that is used to determine the closeness. The proximity value may represent the distance between the two identities, as shown in Figure 12. Examples of the types of detected relationships that can be used to determine a closeness of a connection may include but may not be limited to:

1) the first network identity and the second network identity are declared friends on a social networking site;

2) the first network identity and the second network identity send personal communications to each other via instant messaging;

3) the first network identity and the second network identity send personal communications to each other via email;

4) hyperlinks exist between a blog of the first network identity and a blog of the second network identity;

5) the first network identity has posted a comment on a blog of the second network identity;

6) the first network identity and the second network identity have communicated via corporate email; and

7) the first and second network identities have both posted messages in the same thread on a blog and/or discussion forum.

The proximity values for the above types of detected relationships may be arranged in a descending order to determine the closeness of two identities. For example, a pair of identities, who had communicated via personal email (i.e. type 3), may have a higher proximity value and thus may be closer than a pair of identities who had
communicated via corporate email (i.e., type 6). In this example, the highest possible proximity may be a declared friend as gathered from the social networking sites (e.g., type 1), whereas the lowest possible proximity is shared postings to a common message thread in online forums (i.e., type 7).

[0143] Other types of detected relationships may also be used to determine the closeness of a connection in different embodiments.

[0144] Determining a closeness of a connection may include determining a frequency of communication between the first network identity and the second network identity. As shown in Figure 12, the frequency of communications may determine the thickness of the relationship links 1210.

[0145] The degree of closeness between various identities may be ranked based on the type of detected relationships and the frequency of communications. The closeness of a connection between two network identities may be used in the identification of a possible source of the leaked information. Identifying an employee as a possible source of the leaked information may include using the closeness of the connections between the employee and the network identity of the poster of the leaked information to determine a likelihood that the employee is the source of the leaked information.

[0146] Figure 13 shows a flowchart 1300 of a process for tracing a source of leaked information owned by an organization after the information has been leaked on an external network. In 1302, an identity-relationship database may be built, wherein the identity-relationship database may contain information linking, either directly or indirectly, an employee of the organization to one or more network identities, and to network identities of others with whom the employee communicates. In 1304, the leaked
information may be located on a site on the external network. In 1306, a network identity of the poster of the leaked information may be determined on the external network. In 1308, it may be determined whether one or more links in the identity-relationship database connect the network identity of the poster of the leaked information to the employee. In 1310, if the one or more links connect the network identity of the poster of the leaked information to the employee, the employee may be identified as a possible source of the leaked information.

[0147] Figure 14 shows a schematic diagram of a computer system 1400. In some embodiments, the network gateway device 204, the management device 206 and crawler server 208 may be implemented as a computer system similar to the computer system 1400. In some embodiments, the correlation engine 302, the source code detection module 304 and the network traffic analyzer 306 may also be implemented as modules executing on a computer system similar to the computer system 1400.

[0148] The computer system 1400 may include a CPU 1452 (central processing unit), and a memory 1454. The memory 1454 may be used for storing and/or collecting search results. The memory 1454 may include more than one memory, such as Random Access Memory (RAM), Read-Only Memory (ROM), Erasable Programmable Read-Only Memory (EPROM), hard disk, etc. wherein some of the memories are used for storing data and programs and other memories are used as working memories. The computer system 1400 may include an input/output (I/O) device such as a network interface 1456. The network interface 1456 may be used to access an external network such as the Internet, and an internal network such as Local Area Network (LAN) or Wide Area Network (WAN). The computer system 1400 may also include a clock 1458, an output
device such as a display 1462 and an input device such as a keyboard 1464. All the components (752, 1454, 1456, 1458, 1462, 1464) of the computer system 1400 are connected and communicating with each other through a bus 1460.

[0149] In some embodiments, the memory 1454 may be configured to store instructions for preventing leakage of sensitive digital information on a digital communication network. The instructions, when executed by the CPU 1452, may cause the processor 1452 to intercept at a network gateway device a digital communication being sent from an internal network to an external network, to extract one or more context information items from the digital communication on the network gateway device, to determine a security risk associated with the digital communication based on the one or more context information items, and to send an alert based on the security risk to at least one device connected to the internal network.

[0150] In some embodiments, the memory 1454 may be configured to store instructions for detecting source code in a message being sent over a digital communication network to secure against unauthorized leakage of source code. The instructions, when executed by the CPU 1452, may cause the processor 1452 to intercept the message on a network device, to place the message into the memory 1454 on the network device, to divide the message in the memory into one or more segments, each segment including a predetermined number of lines of text from the message. For each segment, the processor 1452 may apply one or more syntax rules of a programming language to the segment together with a predetermined number of context lines of text before the segment and/or after the segment, to determine which of the syntax rules of the programming language are matched in the segment. The processor 1452 may also provide a determination of
whether the text message includes source code based on the syntax rules that were matched.

[0151]In some embodiments, the memory 1454 may be configured to store instructions for detecting leakage of sensitive source code on network-accessible sites. The instructions, when executed by the CPU 1452, may cause the processor 1452 to determine a set of unique identifying elements that identify a sensitive source code module accessed from the source code repository, to provide search results by searching a list of one or more network-accessible sites for text that matches one or more of the unique identifying elements in the set of unique identifying elements, to collect the search results in the memory 1454, to determine a relevancy for each of the search results based at least in part on a number of the unique identifying elements that were matched and on a number of search results, to sort the results according to the relevancy, and to send the results to the management device to indicate to a user whether sensitive source code was found on the network-accessible sites.

[0152]In some embodiments, the memory 1454 may be configured to store instructions for tracing a source of leaked information owned by an organization, after the information has been leaked on an external network. The instructions, when executed by the CPU 1452, may cause the processor 1452 to build an identity-relationship database containing information linking, either directly or indirectly, an employee of the organization to one or more network identities, and to network identities of others with whom the employee communicates, to locate the leaked information on a site on the external network, to determine a network identity of the poster of the leaked information on the external network, to determine whether one or more links in the identity
relationship database connect the network identity of the poster of the leaked information to the employee, and to identify the employee as a possible source of the leaked information if one or more links in the identity relationship database connect the network identity of the poster of the leaked information to the employee.

[0153] While embodiments of the invention have been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.
Claims

What is claimed is:

1. A method for preventing leakage of sensitive digital information on a digital communication network, the method comprising:
   - intercepting at a network gateway device a digital communication being sent from an internal network to an external network;
   - extracting one or more context information items from the digital communication on the network gateway device;
   - determining a security risk associated with the digital communication based on the one or more context information items; and
   - sending an alert based on the security risk to at least one device connected to the internal network.

2. The method of claim 1, wherein determining a security risk comprises:
   - matching the context information items against one or more predetermined rules on the network gateway device; and
   - sending an alert comprises sending the alert if the context information items match at least one of the predetermined rules.

3. The method of claim 2, wherein matching the context information items against one or more predetermined rules comprises matching the context information against a
rule having one or more conditions, wherein the rule is matched when any of the one or
more conditions are met by the context information items.

4. The method of claim 2, wherein matching the context information items against
one or more predetermined rules comprises matching the context information against a
rule having one or more conditions, wherein the rule is matched when all of the one or
more conditions are met by the context information items.

5. The method of claim 3 or 4, wherein matching the context information items
against a rule having one or more conditions further comprises matching against the one
or more conditions a predetermined number of times greater than one during a
predetermined time window.

6. The method of any of claims 3 to 5, wherein matching the context information
against one or more predetermined rules comprises matching the context information
against a sequence rule comprising a plurality of sub-rules, wherein the sequence rule is
matched when the plurality of sub-rules have been matched in a predetermined sequence.

7. The method of any of claims 1 to 6, further comprising:

storing at least a subset of the one or more context information items in a
historical data store; and
wherein determining a security risk comprises determining the security risk based
at least in part on the data from previous communications stored in the historical data store.

8. The method of claim 7, further comprising:

determining a plurality of mode values based on the information in the historical data store, each mode value representing a frequency with which a predetermined condition occurs in the data from previous communications in the historical data store;

determining a plurality of risk coefficient values based on the plurality of mode values; and

determining a risk score based on the plurality of risk coefficient values and the context information.

9. The method of any of claims 1 to 8 wherein determining a security risk comprises
determining a security risk based on past recorded context information associated with the sender of the digital communication.

10. The method of any of claims 1 to 9, wherein extracting one or more context information items comprises determining one or more of:

   a time at which the digital communication was sent;

   a size of the information in the digital communication;

   a type of information contained in the digital communication;

   whether the digital communication is encrypted;
whether the digital communication contains digital-rights protected content;
an intended destination for the digital communication; and
an identity of an intended recipient of the digital communication.

11. The method of any of claims 1 to 10, wherein extracting one or more context
information items further comprises determining one or more of:
a source of the digital communication;
an identity of a sender of the digital communication; and
a sensitivity of the sender of the digital communication.

12. The method of claim 11, wherein determining a sensitivity of the sender of the
digital communication comprises accessing a data store to determine at least one of:
the involvement of the sender in sensitive projects within the organization;
a last date of work for the sender; and
a preference of a supervisor of the sender.

13. The method of claim 10, wherein determining a type of information contained in
the digital communication comprises determining whether the digital communication
contains source code.

14. The method of claim 1, wherein sending an alert further comprises one or more
of:
sending an alert message that causes a device on the internal network to log the
sending of the digital communication;
sending an alert message and information on the content of the digital
communication to a management device on the internal network; and
quarantining the digital communication by storing it on a device on the internal
network.

15. The method of claim 14, wherein sending an alert message and information on the
content of the digital communication to a management device comprises encrypting the
information on the content of the digital communication so that the information cannot be
viewed by a system manager without authorization from management.

16. The method of claim 15, wherein encrypting the information comprises using a
public key encryption algorithm to encrypt the information.

17. The method of claim 1 to 16, further comprising blocking sending the message to
the external network.

18. A system for preventing leakage of sensitive digital information on a digital
communication network, the system comprising:
a network gateway device that intercepts a digital communication being sent from
an internal network to an external network, the network gateway device comprising
a network connection to the internal network;
a message store, that stores the digital communication; and

a processor for evaluating a security risk of the digital communication based on context information associated with the digital communication the processor configured to:

extract one or more context information items from the digital communication; and
determine the security risk associated with the digital communication based on the one or more context information items;

wherein the network gateway device is configured to send an alert to at least one device connected to the internal network, depending on the determined security risk.

19. The system of claim 18, wherein:

the processor is configured to determines the security risk by matching the context information items against one or more predetermined rules stored in the network gateway device; and

the network gateway device is configured to send the alert if the context information items match at least one of the predetermined rules.

20. The system of claim 18 or 19, wherein the system further comprises:

a historical data store for storing at least a subset of the one or more context information items; and

wherein the processor is configured to determine the security risk based at least in part on the data from previous communications stored in the historical data store.
21. The system of claim 20, wherein the processor is further configured to:

determine a plurality of mode values based on the information in the historical data store, each mode value representing a frequency with which a predetermined condition occurs in the data from previous communications in the historical data store;

determine a plurality of risk coefficient values based on the plurality of mode values;

determine a risk score based on the plurality of risk coefficient values and the context information; and

determine the security risk based on whether the risk score exceeds a predetermined threshold.

22. The system of any of claims 18 to 21, wherein the processor is configured to determine the security risk based on past recorded context information associated with the sender of the digital communication.

23. The system of any of claims 18 to 22, wherein the context information items comprise one or more of:

- a time at which the digital communication was sent;
- a size of the information in the digital communication;
- a type of information contained in the digital communication;
- whether the digital communication is encrypted;
- whether the digital communication contains digital-rights protected content;
an intended destination for the digital communication; and
an identity of an intended recipient of the digital communication.

24. The system of any of claims 18 to 23, wherein the context information items

5 further comprise one or more of:

a source of the digital communication;
an identity of a sender of the digital communication; and

a sensitivity of the sender of the digital communication.

25. The system of claim 24, further comprising an employee data store, wherein the

10 network gateway device is configured to determine the sensitivity of the sender of the
digital communication by accessing the employee data store to determine at least one of:

the involvement of the sender in sensitive projects within the organization;

a last date of work for the sender; and

15 a preference of a supervisor of the sender.

26. The system of claim 23, wherein the system comprises a source code detection

module, and wherein the network gateway device is configured to determine the type of

information contained in the digital communication by using the source code detection

module to determine whether the digital communication contains source code.

27. The system of claim 18, wherein the alert comprises one or more of:
an alert message that causes a device on the internal network to log the sending of
the digital communication;

an alert message sent to a management device on the internal network, the alert
message comprising information on the content of the digital communication; and

a message instruction that the digital message be quarantined by storing it on a
device on the internal network.

28. The system of any of claim 18 to 27, wherein the network gateway device is
further configured to block sending the message to the external network.

29. The system of any of claims 18 to 28, further comprising a management device
connected to the internal network.

30. The system of claim 29, wherein the management device is configured to interact
with a user to construct context-based rules that are sent to the network gateway device
for determining the security risk.

31. The system of claims 29 or 30, wherein the management device is configured to
receive an alert message, and to display the alert message to a user.

32. The system of claim 31, wherein the alert message includes information on the
content of the digital communication, and wherein the information on the content of the
digital communication is encrypted so that it cannot be viewed by a system manager without authorization from management.

33. The system of claim 32, wherein the information on the content of the digital communication is encrypted using public key encryption.
Intercepting at a network gateway device a digital communication being sent from an internal network to an external network.

Extracting one or more context information items from the digital communication on the network gateway device.

Determining a security risk associated with the digital communication based on the one or more context information items.

Matching the context information items against one or more predetermined rules on the network gateway device.

Sending an alert based on the security risk to at least one device connected to the internal network.

Figure 1
Figure 2
Figure 3
Type 1 Correlation Rules

Simple

Aggregate

1) Duration Window
2) Event Count
3) "Group by" parameter

OR

Simple: Total Amount of Source Code

1) Duration Window
2) Event Count
3) "Group by" parameter

Sequence

Fre Form / Custom

Order at which each simple / aggregate / composite rule is matched

Entering full programming script codes based on pre-defined script syntax to systematically create a simple / aggregate / composite / sequence rule

Figure 4
Storing at least a subset of the one or more context information items in a historical data store

Determining a plurality of mode values based on the information in the historical data store

Determining a plurality of risk coefficient values based on the plurality of mode values

Determining a risk score based on the plurality of risk coefficient values and the context information

Risk Score ≥ Risk Threshold Value

No 510 Low Risk

Yes 514 High Risk

Figure 5
Figure 6
public class GeneralUtil {
    protected static final SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    
    /**
     * Creates a new instance of GeneralUtil.
     */
    public GeneralUtil() {
    }

    /** Gets today's date. */
    public static Date getCurrentDate() {
        Date date = new Date();
        try {
            Calendar cal = Calendar.getInstance();
            date = cal.getTime();
        } catch (Exception e) {
            System.err.println("insight.common.util.GeneralUtil: " + e);
            e.printStackTrace();
        }
        return date;
    }
}

Figure 7A
public class GeneralUtil {

  protected static final SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");

  /**
   * Creates a new instance of GeneralUtil.
   */
  public GeneralUtil() {
  }

  /**
   * Gets today's date.
   */
  public static Date getCurrentDate() {
    Date date = new Date();
    try {
      Calendar cal = Calendar.getInstance();
      date = cal.getTime();
    }
    catch (Exception e) {
      System.err.println("Insight.common.util.GeneralUtil": " + e);
      e.printStackTrace();
    }
    return date;
  }
}

Figure 7C
Intercept a message on a network device and place the message into a memory on the network device

Divide the message into one or more segments, wherein each segment includes a predetermined number of lines of text from the message

Providing a coefficient for each of one or more syntax rules of a programming language, wherein each coefficient is based at least in part on the uniqueness of the syntax rule

for each segment, apply the one or more syntax rules in an order based on their coefficients, to determine whether the segment matches the syntax rule

determine a product of the coefficient for the syntax rule and the number of characters of text in the segment that matches the syntax rule

Determine a ratio based on the product for each syntax rule and the number of characters of text in each segment

ratio > threshold?

No

Yes

Stop the application of syntax rules, and provide a determination of whether the message includes source code

Figure 8
```java
package insight.common.util;

import java.text.SimpleDateFormat;

public class GeneralUtil {
    protected static final SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");

    public static Date interestingMethodAction() {
        //this is my comment for the interestingMethodAction
    }

    //Gets today's date
    public static Date getCurrentDate() {
        Date date = new Date();
        try {
            Calendar cal = Calendar.getInstance();
            date = cal.getTime();
        }
        catch (Exception e) {
            System.err.println("insight.common.util.GeneralUtil. " + e);
            e.printStackTrace();
        } return date;
    }
    etc

    900

Figure 9
```
<table>
<thead>
<tr>
<th>Type of Element</th>
<th>Too Generic</th>
<th>Unique</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-line comments</td>
<td></td>
<td>&quot;this is my comment for the interestingMethodAction&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Gets today's date&quot;</td>
</tr>
<tr>
<td>Declared package names</td>
<td></td>
<td>&quot;insight.common&quot;</td>
</tr>
<tr>
<td></td>
<td>Method names</td>
<td>&quot;insight.common.util&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>InterestingMethodAction</td>
</tr>
<tr>
<td></td>
<td>GetCurrentDate (discarded by blacklist)</td>
<td></td>
</tr>
<tr>
<td>Classes names</td>
<td>GeneralUtil (discarded by blacklist)</td>
<td></td>
</tr>
<tr>
<td>File name</td>
<td>GeneralUtil (discarded by blacklist)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 10
Determining a set of unique identifying elements that identify a sensitive source code module accessed from an internal source code repository

Using a crawler server connected to an external network to automatically search a list of one or more network-accessible sites for text that matches one or more of the unique identifying elements in the set of unique identifying elements, to provide search results

Collecting the search results in a memory of the crawler server

Determining a relevancy for each of the search results based at least in part on a number of the unique identifying elements that were matched and on a number of search results

Sorting the results according to the relevancy

Providing the results to a user, to indicate whether sensitive source code was found on the network-accessible sites

Figure 11
Figure 12

LEGENDS:

- Target Unknown identity
- First level friends
- Second level friends
- Width of connecting lines to indicate the frequency of communication between the 2 connected identities
- Critical path indicating link from target unknown identity to an internal staff
Build an identity-relationship database, which includes information linking, either directly or indirectly, an employee of the organization to one or more network identities, and to network identities of others with whom the employee communicates.

Locate the leaked information on a site on the external network.

Determine a network identity of the poster of the leaked information on the external network.

Determine whether one or more links in the identity-relationship database connect the network identity of the poster of the leaked information to the employee.

If yes, identify the employee as a possible source of the leaked information.
Figure 14
INTERNATIONAL SEARCH REPORT

International application No.
PCT/SG2008/000270

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl.
G06F 21/00 (2006.01)    H04L 9/00 (2006.01)

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

B. MINIMUM DOCUMENTATION SEARCHED

Minimum documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

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>GTB Technologies: &quot;ESSENTIAL REQUIREMENTS FOR A COMPREHENSIVE DATA LEAK PREVENTION SYSTEM&quot;, April 2007. See whole document, especially sections 1 (&quot;Appliance, not Software&quot;) and 4 (&quot;Precision&quot;)</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>US 2007/079365 (ITO ET AL) 5 April 2007 See whole document, especially abstract</td>
<td></td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C

X 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

Date of the actual completion of the international search
09 September 2008

Date of mailing of the international search report
12 SEP 2008

Name and mailing address of the ISA/AU
AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
E-mail address: pct@ipaustralia.gov.au
Facsimile No. +61 2 6283 7999

Authorized officer
BAYER MITROVIC
AUSTRALIAN PATENT OFFICE
(ISO 9001 Quality Certified Service)
Telephone No: +61 2 6283 2164

Form PCT/ISA/210 (second sheet) (July 2008)
This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2007079365</td>
<td>JP 2007096666</td>
</tr>
</tbody>
</table>

Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.

END OF ANNEX