(19) IPOS INTELLECTUAL PROPERTY OFFICE OF SINGAPORE

(11) Publication number: **SG 177015 A1**
(43) Publication date: **30.01.2012**
(51) Int. Cl: ;

(12) Patent Application

(21) Application number: **2010039816**
(22) Date of filing: **07.06.2010**

(71) Applicant: **BOXSENTRY PTE LTD 3 PHILLIP STREET, #13-03 COMMERCE POINT, SINGAPORE 048693 SG**

(72) Inventor: **GOEL, MANISH KUMAR 27 MOUNT FABER ROAD, "MOUNT FABER LODGE" #07-04, SINGAPORE 099200 SG**
TURNER, ROLAND JOHN 26 KIM TIAN ROAD "TWIN REGENCY" #07-03, TIONG BAHRU, SINGAPORE 168732 SG

(54) **Title:**
IN SITU CORRECTION OF FALSE-POSITIVE ERRORS IN MESSAGING SECURITY SYSTEMS (LAGOTTO)

(57) **Abstract:**
IN SITU CORRECTION OF FALSE-POSITIVE ERRORS IN MESSAGING SECURITY SYSTEMS (LAGOTTO) Abstract The invention provides the means to perform reporting and/or correction of false- positive errors caused by the filters in already-deployed messaging-security systems, and to do so in most cases without requiring alterations to those systems which minimises barriers to entry thereby making false-positive error correction more readily available. This is achieved by making use of quarantine and log access interfaces already present on many widely-deployed messaging security systems and performing after-the-fact error correction, rather than by focussing solely on preventing errors from being made in the first place. Figure 1

IN SITU CORRECTION OF FALSE-POSITIVE ERRORS IN MESSAGING SECURITY SYSTEMS (LAGOTTO)

Abstract

The invention provides the means to perform reporting and/or correction of false-positive errors caused by the filters in already-deployed messaging-security systems, and to do so in most cases without requiring alterations to those systems which minimises barriers to entry thereby making false-positive error correction more readily available. This is achieved by making use of quarantine and log access interfaces already present on many widely-deployed messaging security systems and performing after-the-fact error correction, rather than by focussing solely on preventing errors from being made in the first place.

Figure 1

Title

IN SITU CORRECTION OF FALSE-POSITIVE ERRORS IN MESSAGING SECURITY SYSTEMS (LAGOTTO)

Cross Reference

Incorporated herein by reference is PCT/AU2009/001614 entitled "Electronic Messaging Integrity Engine".

Technical Field

This invention relates to ensuring reliable delivery of electronic messages, such as email messages, despite errors in recipient's existing email messaging security system, without alteration to that system.

Summary of the prior art

Organisations seeking to avail themselves of the benefits of false positive error avoidance systems frequently encounter cost and/or organisational resistance barriers to adoption because of the need to alter complex existing systems in order to do so.

Shifting to after-the-fact false-positive error correction makes it possible to work with existing messaging security systems as-is which in turn minimises those barriers to adoption thereby enabling more organisations to reap the benefits of improved reliability in message delivery.

Summary of the Invention

The invention provides the means to perform reporting and/or correction of false-positive errors caused by the filters in already-deployed messaging-security systems, and to do so in most cases without requiring alterations to those systems which minimises barriers to entry thereby making false-positive error correction more readily available. This is achieved by making use of quarantine and log access interfaces already present on many widely-deployed messaging security systems and performing

after-the-fact error correction, rather than by focussing solely on preventing errors from being made in the first place.

Brief description of Drawings

Figure 1 illustrating an overview of the system.

Detailed Description

Almost all widely-deployed messaging security systems provide access to logs of messages received and delivered. In conjunction with the previous invention cross referenced above, this information is sufficient to detect a large fraction of the messages lost through false-positive errors made by those systems' filters.

The majority of widely-deployed messaging security systems also provide access to some/all messages that were classified as [probably-]spam and quarantined. In combination with the detection made possible by the above, this makes possible the automated correction of these false-positive errors through the mechanisms already in place in those deployed systems.

Lagotto is a system which receives and processes message logs from existing messaging security systems, detects and reports false-positive errors and – where possible – corrects them.

A Lagotto installation as shown in Figure 1 will consist of some or all of:

- A configuration database
- An administrative API and user-interface to manage the configuration of the system
- A variety of modules for receiving and processing logs from existing systems
- A message-queue to feed processed log entries to a LogiQ client module
- A LogiQ client module to train and query a LogiQ instance and detect false-positives
- A database of detected false-positives for reporting

- A message-queue to feed detected false-positive information to a message releasing module
- A message-releasing module
- A reporting module

Basic Operation

- A log of messages sent and received is transferred from the existing messaging security system to Lagotto.
- A log receiver communicates in whatever protocol is appropriate to receive the transferred log. A log converter passes the received log entries and converts them into a single internal format. The resulting entries are queued on the LogiQ client.
- The LogiQ client turns each of the entries into a LogiQ query, sends it and waits for the result.
- Where the message was inbound, classified as spam by the existing messaging security system but recognised as legitimate by LogiQ, the details are logged in the reporting database and queued on the releasing module.
- The releasing module turns each of the detected false-positive errors into a message release instruction to the messaging security system.
- Periodically, the reporting module generates reports for the corrections on specific systems to the relevant administrators.

Configuration Information

(Super-administrator account creation/resetting is handled out of band)

Create/Retrieve/Update/Delete:

- Users (username, password/openID, roles/groups)
- LogiQ instances (hostname, which roles/groups have access, key)
- Messaging security systems (name, which roles/groups have access, type, which LogiQ instance to use, log retrieval schedule, log-receiving parameters, reporting schedule, report retention time, whether to release detected false positives automatically)

Receiving Logs

In the simplest case, all of the logs which Lagotto uses to perform its function come from a dedicated messaging security system (e.g. an anti-spam system). In more complicated cases, logs may also come from a dedicated message store (e.g. a mail-server which does not have anti-spam capabilities built it) or from a combined messaging security and store (e.g. a mail-server which does have anti-spam capabilities built in). Either or both logs may also come from a log analysis and consolidation system that gets its logs from the messaging components by existing means. A typical example of a more complicated case is that of a dedicated messaging security system used only for inbound message handling connected to a dedicated message store which performs its own outbound delivery directly. In this case, the logs of inbound messages would need to come from the messaging security system – in order to include information about messages received but classified as spam and therefore not delivered – while the logs of outbound messages would need to come from the message store. For simplicity's sake, and without loss of generality, the rest of this document refers to all logs as though they came from a messaging security system, regardless of the actual source of the logs and arrangement of components.

There are several paths that logs may take to get from the messaging security system to Lagotto:

- Lagotto can periodically download logs directly from the messaging security system.
- The messaging security system can periodically upload logs to Lagotto.
- The messaging security system can periodically upload logs to a storage facility to which both the messaging security system and Lagotto have access. Lagotto can then periodically download from the storage facility.
- The messaging security system can send log entries to Lagotto in real time via an appropriate protocol (e.g. the syslog protocol as described in IETF RFC 3164).

In the situations where Lagotto is downloading logs - either directly or via a storage facility - it can do so on a configured schedule, or in response to an API call from an external piece of software to trigger immediate commencement of a download, or both. Protocols appropriate for uploading and downloading of logs include, but are not limited to, POST operations via HTTP, FTP, SMB, etc.

In some cases it may make sense to dispense with log transfer entirely and instead to have the messaging security system deliver a copy of the inbound-classified-as-spam and/or outbound mail streams to Lagotto via SMTP. In this case, Lagotto would pass out internal format logs for queuing on the LogiQ client as other log receivers and converters do, but also to maintain a circular buffer of several minutes' inbound-classified-as-spam email as an internal "quarantine" from which detected false positives can be released.

Reporting

On a configured schedule, the report generation module creates summary reports of messages that were detected as false-positive errors and then corrected. For messaging security systems with very large numbers of errors, only summary statistics are reported. For each messaging security system that a Lagotto installation is monitoring, different reporting intervals, report retention periods and notification settings (whether reports are simply generated and stored, or also emailed to specified addresses) may be specified.

A user interface for browsing the detected false positives and manually releasing them is also provided for users who would prefer to have correction not performed automatically.

Releasing Messages

In most situations an interface present on the existing messaging security system will be used to release [a copy of] the quarantined message to the mail-server (examples include a quarantine management API, IMAP access to the quarantine or a "screen-

scraping" tool which releases a message from the quarantine in a way which looks to the existing system like a user logging in and then selecting and releasing the message).

In some cases the user will elect not to have corrections performed automatically – or Lagotto will not have the means to use available interfaces - but prefer to review the list of apparent false-positives and release them manually. As mentioned above, a user-interface is provided for this purpose.

In some cases it will not be possible, even in principle, for errors to be corrected. This will usually be the case where the existing messaging security system has refused a connection from a peer MTA or has refused or dropped a message that Lagotto was able to authenticate without a copy ever having been stored. In this situation, Lagotto will simply report what it knows (that a message from a known good sender was refused/dropped, or that a connection from an IP address known to send some legitimate email was refused), and no release is possible.

Operating without all required facilities

- No log access: Lagotto is built on the processing of logs, if the relevant logs are not available at all then there is very little scope for using Lagotto to correct false-positive errors, but note that in some cases it may still be possible to have the inbound-classified-as-spam and outbound message streams copies to Lagotto via SMTP as described earlier, in which case the same information can be extracted and correction can proceed as usual.
- No access to information about outbound email: Much of LogiQ's – and therefore Lagotto's - operation depends upon observing email communication in both directions. Situations in which a service provider secures a customer's inbound email stream but has no contact with the corresponding outbound stream arise frequently. In such cases, LogiQ's ability to use BoxSentry's global reputation system, TrustCloud, provides Lagotto with the ability to perform a large subset of the error correction that could otherwise be performed.
- No quarantine access: Some messaging security systems provide no means to release messages from quarantine, or customers with bespoke systems may elect

not to support integration of their quarantine with Lagotto. In these cases Lagotto can still report on the extent of the problem, which is useful in some situations for SLA compliance monitoring. There may also be the option of a copy of the inbound-classified-as-spam message stream being delivered to Lagotto via SMTP to function as an internal "quarantine" from which misclassified-as-spam messages can be released.

Scaling Considerations

As Lagotto operates slightly after the fact, its performance is rarely critical, however for large installations the workload will readily exceed what can be performed by a single server. Fortunately Lagotto retains very little persistent state and what little it retains is slow-changing, rarely-aggregated, or both, making scaling straight-forward:

- Even for very large installations, the rate of change of the configuration database is negligible. At worst it will be necessary to make a change each time a domain is added to or removed from [one of] the messaging system[s] being monitored, although in most cases even this won't be necessary. Consequently the configuration database can trivially be replicated to read-only copies that are used by components on different servers. For the same reason the admin UI/API need only be on a very small number of servers, typically one (or two where high-availability is required and virtualisation is not in use).
- The log receivers and converters operate statelessly between sessions and can therefore be horizontally scaled as required.
- The two "queues" can be parallelised and therefore scaled using readily available message queuing systems.
- The LogiQ client also operates statelessly and can therefore be horizontally scaled as required.
- The releasing modules also operate statelessly and can therefore be horizontally scaled as required.
- The reporting module does need to aggregate all data related to a particular messaging security system collected over a period of time and, therefore, needs to work with data that may have originated from any of multiple servers in a large installation, however detected false positives typically number three

orders of magnitude below the total number of messages processed by a messaging security system. At a first approximation if all of Lagotto apart from the reporting module and database are deployed on fewer than a thousand servers, then the reporting and module and database can probably be deployed on a single server (or two where high-availability is required and virtualisation is not in use).

It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

CLAIMS:

1. The steps, features, integers, compositions and/or compounds disclosed herein or indicated in the specification of this application individually or collectively, and any and all combinations of two or more of said steps or features.

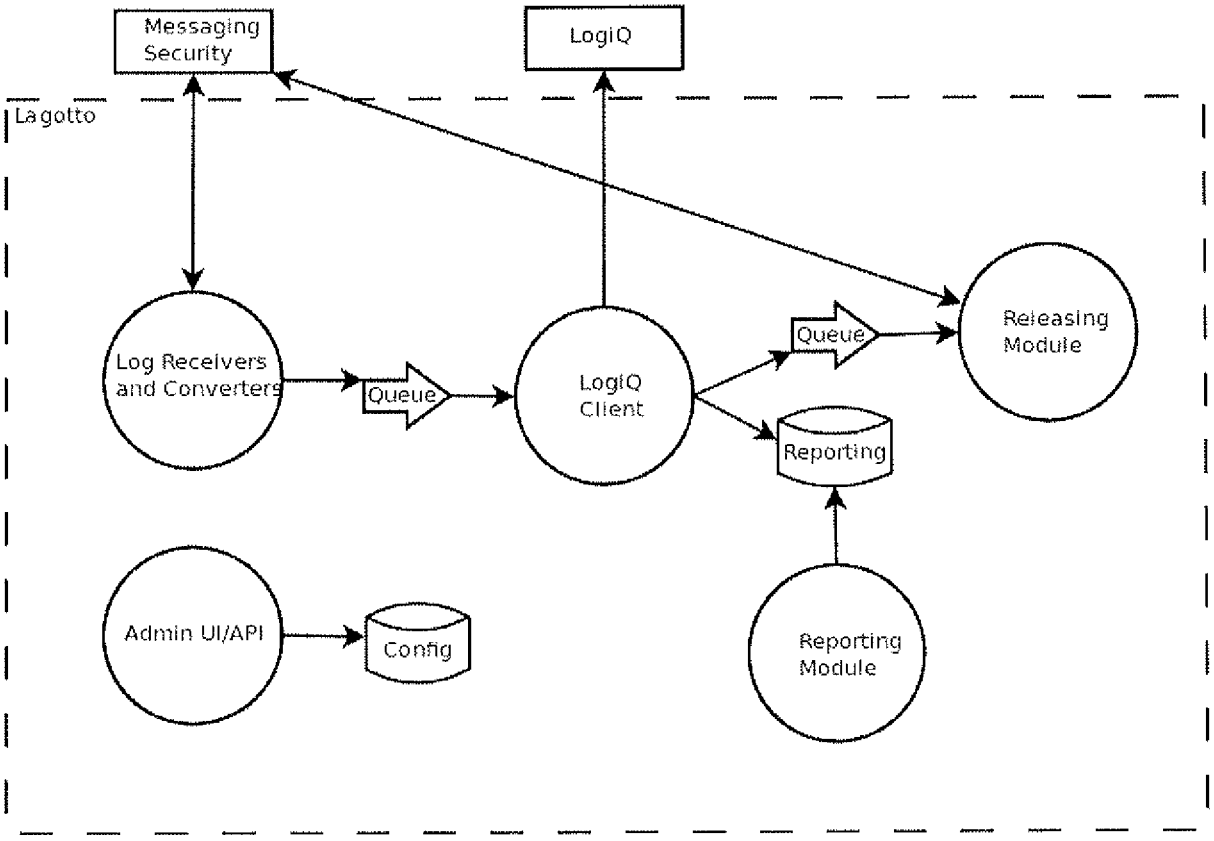
Messaging Security
LogiQ
Lagotto
Log Receivers and Converters
Queue
LogIQ Client
Queue
Releasing Module
Reporting
Admin UI/API
Config
Reporting Module

Figure 1