



# Boundary Protection

(Build Number 3111)



## Contents

Overview .....	3
Philosophy .....	4
Security .....	5
Attacks on the messaging server .....	5
Server Control Attacks .....	5
Message Relay Control .....	5
Service Denial Attacks .....	6
Attacks on the mail clients .....	6
Message Content Attacks .....	7
Conclusion .....	8
Appendix A: Boundary Protection Summary .....	9
Mail sent via SMTP - Global level protection .....	9
Mail sent by SMTP - Per domain protection .....	15
Mail sent by SMTP - Per user protection - set by administrator .....	15
Mail sent by SMTP - Per user protection - set by user .....	16
Protection of mail collection services .....	17
Protection using HTTP(S) and FTP proxy services .....	17
System protection .....	17
Appendix B: Glossary .....	19



## **OVERVIEW**

The Gordano Messaging Suite has a wide range of options and controls that allow extremely granular control over access to the Internet via email, web and ftp services.

Electronic mail messages (email) are accepted using the Internet standard SMTP (and SSMTP) protocol and pass through a series of checks before being delivered, redirected, delayed, quarantined or rejected. GMS includes checks that help prevent attacks targeted at known vulnerabilities of particular clients (e.g. iframes, malformed messages, etc.).

Access to the Internet via the web (HTTP and HTTPS) and file transfer services (FTP) is available through the integrated caching proxy service. This service allows the administrator to control the access allowed to the Internet as well as the content delivered.

Appendix B contains a short glossary to expand on some of the technical terms used within this document.



## PHILOSOPHY

The GMS scalable architecture has been designed to manage extremely high volumes of messages on standard commodity systems (e.g. six million messages per day has been recorded on a dual pentium machine). In order to maintain this throughput with a large number of complex analyses the system has been designed to carry out checks as early in the delivery process as possible. Each SMTP session will completely process the message delivery process before appending the message to the destination user's InBox directly.

Some other systems use a "pipelining" approach to delivering messages where as each check is passed, the message is queued for the next check to be carried out. With over a hundred checks being carried out per message (see the appendix A for a full listing) this mechanism may result in an unmanageable number of queues and processes leading to the possible overflow of any queue, loss of messages, temporary bottle necks, delivery delays, etc. Worse, this design requires the complete acceptance of a message before any checks can be carried out. This introduces two more major requirements on such a system:

- The system must return messages that are rejected; and
- A greater bandwidth is required to accept and then return unwanted messages.

Gordano's solution does not suffer from these issues and has additional advantages:

- The administrator can define what protocol response is sent in response to errant messages. For example, if the administrator has specified their own filter (e.g. using MML), they could request the SMTP response "550 Rejected – please see <http://www.mycompany.com/reject1.htm> for more information." to be sent each time the filter is activated. These filters could be applied to the whole system or a single, specific account. This gives additional control and aids continuous improvement of such filters.
- Messages containing viruses are quickly identified and rejected without residing on the server for longer than is necessary.
- There are no queues to manage, so server utilisation is considerably lower. Sizing the server for a given number of accounts is simpler.
- A message delivered for multiple accounts can pass through the system once and have all the checks performed once rather than one time for each destination account. This is particularly important for companies who have employees on external news and discussion forums – in this case a message delivered to 100 internal employees is checked once rather than 100 times. This represents a significantly lower load on the messaging system, yielding higher throughput.

## **SECURITY**

Attacks on messaging systems can take several forms. GMS has been designed to mitigate the risk of all attacks as much as is possible while still supporting the Internet Standards. There are two major areas of attacks – first on the messaging servers themselves and second on the destination mail clients. A further type of attack revolves around obtaining the message content.

### ***Attacks on the messaging server***

Attacks on the messaging server may lead to the complete compromise of the company's IT infrastructure. There are three major forms of attack:

- gaining control of the server as a stepping stone to accessing other systems within an organisation;
- to use the server resources for delivering messages on behalf of someone else (mail-relay); and
- denying access to the server for legitimate purposes.

### **SERVER CONTROL ATTACKS**

Traditionally gaining control of a server is attempted by exploiting buffer over-runs within code that has not been rigorously verified and tested. Once a buffer over-run has occurred, the hacker can cause his own code to be executed rather than the correct code. Once this occurs, the hacker can upload their own programs and gain complete control of the server.

Additional issues have come to light recently where software is downloaded from a site that has mal-functioning code. For example, a version of sendmail became available with "Trojan" code in it.<sup>1</sup> In order to avoid this scenario, it is essential for systems administrators to download code only from reputable sources.

Gordano has both implemented procedures within the software to prevent buffer over-runs and rigorously applied peer review to prevent any instance of these types of issues within any service. Further, each version of software is made available from our web site and old versions are available on our ftp site so that customers always have access to a known level of software.

### **MESSAGE RELAY CONTROL**

In the early days of the Internet, all messaging systems allowed mail to be relayed. This helped to improve the robustness and efficiency of the messaging network. In 1996 some individuals and organisations exploited this facility to send large quantities of unsolicited commercial email (UCE) through networks by delivering one message with multiple recipients to a server which (correctly) proceeded to deliver the message to all the requested destinations. Today, all

---

<sup>1</sup> See <http://www.newscientist.com>



reputable and correctly configured Internet mail servers are now locked down to prevent this unauthorised use. However, there are occasions when such relay facilities are still required, so a high degree of control and flexibility of this relay is essential for any messaging system operating in today's environment.

When first installed, Gordano's Messaging Suite has relay turned off so that no email may be relayed. Administrators have a wide range of choices about how the control can be modified. For example "allowing all relay", "allowing relay for specific domains", or "allowing relay" with a large list of authentication options. As a result, GMS can provide suitably controlled relay for any mail client where company operations require them.

### **SERVICE DENIAL ATTACKS**

Miscreants, disgruntled employees, cyber terrorists, etc. may attempt to flood server resources in order to prevent legitimate use being made of a server. Such attacks may result in multiple connections from servers, starting a connection and issuing "idle" commands (tar-pitting), etc. All of the services resources are thus used by responding to the "non-requests" so no legitimate messages are processed.

GMS has a range of options enabled, by default, to prevent multiple connections, tar-pitting, etc. which are enabled by default. When any filter criteria is met, the system posts alerts for the administrator.

### ***Attacks on the mail clients***

Messaging users' systems often have considerable IPR in the form of address books, documents, spread sheets and presentations. Further, they often have authenticated access to other resources within an organisation – file archives, control systems, etc. This makes their systems a very useful site for a miscreant to cause maximum problems for an organisation. There are several stages to an attack:

- Deliver code to destination system.
- Get the processor to execute the code.
- Carry out damage or copy data to an external site.

Delivering code to a destination is generally carried out by using attachments to mail messages. An attachment may contain a word document, executable, binary, commands, etc. Anti-virus software can be used to identify and remove such code from attachments.

Getting the processor to execute the code used to be a case of asking the user to "double click" the attachment to see it. However, attacks are now much more sophisticated and use bugs and weaknesses within mail clients to execute the code without the control of the user. For example, mail clients that display HTML messages may be vulnerable to the "iframe" issue which causes the miscreant's



code to be run directly. Broken "MIME" boundaries may be used to cause buffer over-flows within clients.

Gordano's Messaging Suite contains message analysis to allow blocking of attachments, virus scanning (and cleaning), message content checking, etc. for just these types of exploits. If a new exploit is discovered, MML can be used to allow the capture and review of messages using it.

### ***Message Content Attacks***

Message content may be compromised while the message is in transit or residing on the message server disk (awaiting download or while queued for on bound delivery). While messages may be encrypted by employees individually, the time taken to carry out such encryption for the majority of messages vastly out-weighs the content. However, an intermediate level of encryption while messages are between servers and on servers may be required.

Gordano's Messaging Suite contains encryption software that will encrypt messages using SSL when they are travelling between suitable configured servers. This allows partner organisations to ensure that their communication is safe even though communication is travelling through multiple public networks.

At the disk level on the Windows platform, Microsoft's disk encryption and access control technology may be used with GMS in order to prevent unauthorised access to message content.



## **CONCLUSION**

Since Gordano first released Internet messaging software in 1994, the Internet has become a much harsher environment – attacks on organisations through messaging, web services, etc. are increasing and will continue to do so. Gordano has been at the forefront of technology and, as a result, has protected its customers from attacks through its messaging systems while retaining the overall performance of the solution.

Those organisations that are not using GMS to provide all messaging services directly can install GMS at the Firewall level as a “boundary protection” system to prevent attacks reaching other, more sensitive internal messaging systems.

Gordano recognises that a “one-size-fits-all” solution cannot possibly provide a good solution for all its customers. Therefore GMS has a highly granular range of additional easy-to-manage security options in order to allow the organisation to choose the solution that fits its environment the best.

## **APPENDIX A: BOUNDARY PROTECTION SUMMARY**

The following section lists 147 methods and facilities that are available to improve the security of a corporate system using GMS.

### ***Mail sent via SMTP - Global level protection***

All messages sent to GMS via the SMTP protocol will pass through the following checks and may be rejected during the "protocol" stage in order to save bandwidth.

1. Block a specific IP or range of IP addresses using IP matching rules with SMTP protocol reject or retry option.
2. Block one or more specific hosts using the host name with SMTP protocol reject or retry option.
3. Limit the number of messages per day from any IP using IP matching rules. Reject or retry at SMTP protocol.
4. Limit the number of messages per day from any sender clause.
5. Set the maximum number of messages per SMTP session for the whole server and/or specific sender IP addresses. Either reject or retry later at SMTP protocol level.
6. A series of RBL checks on remote server connection with providing accept/try next/deny for each RBL services entered. Specify SMTP protocol response for each RBL service tested.
7. A series of RBL checks on MAIL clause with providing accept/try next/deny for each RBL services entered. Specify SMTP protocol response for each RBL service tested.
8. Specify IP addresses using IP matching rules for clients that are considered local and allowed to send email from domains managed by this server. This check prevents those from outside attempting to deliver email claiming to be from a local account.
9. Relay control allows several options: 1. Allow relay, 2. Disallow relay, but allow mail where MAIL clause or more than one RCPT clause is local; and 3. Disallow relay, all RCPT clauses must be local. Enter set of domains allowed to relay to support mutual MX server backups. Reject or retry later using SMTP protocol message.
10. Limit maximum number of simultaneous connections from any remote IP address (default is 5, helps to prevent DoS attacks). Provide over-rides for a specific selection of addresses with a different number of maximum connections per IP address.
11. Check the name of the machine in the "HELO" clause against reverse IP lookup to ensure sender is who they claim to be. Option to terminate connection if they are not the same. Elect to use real IP address in logs or reverse lookup details in logs.
12. Perform MX lookup on sender's address to verify that the domain being sent from exists before allowing message delivery to continue.

13. If address message is being sent from is in the local domain, verify that the account exists before allowing message delivery to continue.
14. Perform MX lookup on recipient's address to verify that the domain being sent from exists before allowing message delivery to continue.
15. Limit the number of bad commands used by remote server (prevents remote server harvesting addresses and prevents web proxy bypass attacks). If maximum reached, define how long to ban further connections from this server.
16. Maximum number of idle commands used by remote sending server or recipient server (prevents a tar-pitting attack from using the resources of the local system). If maximum reached, define how long to ban further connections from this server.
17. Reject messages with specific attachments.
18. Set the maximum outbound message size.
19. Filtering based on any part of the message header and/or message body and be restricted to a given number of lines. Messages matching the filter can be failed with SMTP error code, copied to quarantine and/or returned to sender. Further a bypass list of IP addresses is available.
20. Global filtering based on the number of instances of one or more words within messages. Messages that match the filter may be failed with an SMTP code, delayed with an SMTP code, forwarded to specific address or copied to another address.
21. Options for authenticated POP users allow authenticated users to have all normal GMS Anti-Spam checks, allow relay but perform other checks, or bypass all GMS Anti-Spam checks. Specify how long authenticated user has to use the authenticated IP address before further mail is rejected.
22. Options for authenticated IMAP4 users allow authenticated users to have all normal GMS Anti-Spam checks, allow relay but perform other checks, or bypass all GMS Anti-Spam checks. Specify how long authenticated user has to use the authenticated IP address before further mail is rejected.
23. Reject, ignore, or fix messages with lines not terminated by CRLF.
24. Reject messages with lines length exceeding RFC2822 limits.
25. Reject messages with attachments names that are too long.
26. Reject messages with suspicious attachment names.
27. Reject messages with a CLSID in attachment name.
28. Reject messages with UUEncode begin in subject
29. Reject messages with UUEncode incomplete begin statement.
30. Reject messages with UUEncode data with blank lines.
31. Reject messages with UUEncode data with spaces.
32. Reject messages with UUEncode data line too long.
33. Reject messages with UUEncode data invalid.
34. Reject messages with UUEncode data that does not decode correctly.
35. Reject messages with Base64 encoding of inline text.
36. Reject messages with Base64 data invalid.
37. Reject messages with Base64 data invalid length.
38. Reject messages with Base64 data has leading '=' signs.

39. Reject messages with Base64 data has too many '=' signs.
40. Reject messages with Base64 data after end of decode.
41. Reject messages with Base64 data line too long.
42. Reject messages with Binhex data in text section.
43. Reject messages with Binhex data invalid.
44. Reject messages with no final MIME boundary.
45. Reject messages with 8 bit characters in header of MIME field.
46. Reject messages with partial MIME message fragment.
47. Reject messages with invalid MIME fieldname format.
48. Reject messages with invalid message/rfc822 content type.
49. Reject messages with MIME comment detected.
50. Reject messages with MIME section in prolog or epilog.
51. Reject messages with HTML component with IFrame entities. Note, this protects from attacks on mail clients which understand and display HTML code.
52. Reject messages with HTML component using CID to load file. Note, this protects from attacks on mail clients which understand and display HTML code.
53. Reject messages with HTML component has Object entities. Note, this protects from attacks on mail clients which understand and display HTML code.
54. Run MML script on connection. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection. One or more scripts may be used.
55. Run MML script on HELO clause. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection. One or more scripts may be used.
56. Run MML script on EHLO clause. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection. One or more scripts may be used.
57. Run MML script on MAIL clause. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection. One or more scripts may be used.
58. Run MML script on RCPT clause. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection. One or more scripts may be used.
59. Run MML script on DATA clause. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection. One or more scripts may be used.

60. Run MML script on EOM clause. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection. One or more scripts may be used.
61. Run MML script on message arriving for any mail arriving at any local account. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection. One or more scripts may be used.
62. Run MML script on a message being delivered to a specific Internet domain. Each Internet domain may have one or more scripts. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.
63. Run MML script on delivery of a message to a specific account on the system. Each account may have one or more scripts. The script can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.
64. Run DLL (on Windows) or .so (on unix) library function on connection. The function can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.
65. Run DLL (on Windows) or .so (on unix) library function on HELO clause. The function can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.
66. Run DLL (on Windows) or .so (on unix) library function on EHLO clause. The function can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.
67. Run DLL (on Windows) or .so (on unix) library function on MAIL clause. The function can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.
68. Run DLL (on Windows) or .so (on unix) library function on RCPT clause. The function can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.
69. Run DLL (on Windows) or .so (on unix) library function on DATA clause. The function can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.
70. Run DLL (on Windows) or .so (on unix) library function at the end of receipt of the message. The function can return a status to the SMTP server telling the SMTP server to continue the connection, return a specific SMTP protocol message or drop the connection.

71. Load checking based on average number of messages sent from this remote server. If the server delivery spikes, the additional messages can be rejected or retried later at the SMTP protocol phase.
72. Load checking based on average number of messages sent from this email address. If the server delivery spikes, the additional messages can be rejected or retried later at the SMTP protocol phase.
73. Load checking based on average number of messages sent to this email address. If the server delivery spikes, the additional messages can be rejected or retried later at the SMTP protocol phase.
74. The virus scanner will decode all MIME, uuencode and binhex attachments for checking. When infected email arrives the virus scanner gives options to: return the message with user defined text (optionally disinfecting it), reject the message, redirect the message (optionally disinfecting it), deliver the message as usual (optionally disinfecting it) or place the message into the quarantine folder for further examination. Request updates to signature files as frequently as every 15 minutes.
75. Files discovered for virus scanning are uncompressed (if they are ZIP files) and the content of the archive individually scanned. Recursive decompressing is carried out until the final files are found.
76. If virus scan fails for any reason optionally reject the email and notify the postmaster that an error occurred.
77. Request that the virus scanner scans inline text as well as all attachments.
78. Define maximum idle time for any remote connection.
79. Insert missing "From:" clause (as per RFC822) so that on-bound messaging systems and clients can process the message correctly.
80. Insert missing "To:" clause (as per RFC822) so that on-bound messaging systems and clients can process the message correctly.
81. Always insert missing "To:" clause if not available so that on-bound messaging systems and clients can process the message correctly.
82. Insert "Return-Path:" clause to allow tracking of messages.
83. Fix addresses created by MSMail systems so that on-bound messaging systems and clients can process the message correctly.
84. Set the maximum number of hops that a message can make in order to prevent message looping between one or more systems.
85. Set the maximum length of "Received" line in order to prevent mail clients buffer-over runs.
86. Send bad recipient notifications (always, never, or if there are multiple recipients).
87. Enable or disable all the enhanced SMTP options.
88. Enable or disable the ESMTP option VRFY. VRFY allows external systems to check that an email address does exist before sending email to it.
89. Enable or disable the ESMTP option for delivery status notification (DSN). If enabled, this option will cause GMS to special messages showing the current delivery status to servers that request them.
90. Enable or disable the ESMTP option to display Enhanced Status Codes. These error codes give additional information about why a message has

been rejected by GMS and help in the debugging and tracking of errant messages.

91. Enable or disable the ETRN command which causes the server to start sending messages via SMTP to a specific IP address. Typically used by ASPs, there are several options available, no password (so mail is delivered to the MX record), plaintext password (allows delivery to either an MX record and static IP address) and MD5 password (allows delivery to either an MX record or any IP address).
92. Enable or disable the ESMTP option SIZE. This option allows a sending server to reserve disk space on the destination. GMS can respond with an error if enough disk space is not available so saving bandwidth.
93. Enable or disable the ESMTP option to accept 8BitMIME messages. This reduces the bandwidth for message delivery. However, on bound servers may not accept such messages. In this case, GMS will convert the messages to 7-bit MIME.
94. Enable or disable the ESMTP option to allow AUTHenticated sending or relaying of email messages. Password options include Login, MD5, CRAM-MD5 or plain text.
95. Enable or disable the ESMTP option to allow Pipelining which reduces the latency in sending messages.
96. Enable or disable the ESMTP option to allow Restart and Recovery which allows partial messages to be sent thus reducing bandwidth requirements.
97. Enable or disable the ESMTP option to allow SSL over SMTP. This results in the entire contents of the message being encrypted.
98. Set the maximum inbound SMTP bandwidth so utilisation of network resources is automatically limited.
99. Set the maximum outbound SMTP bandwidth so utilisation of network resources is automatically limited.
100. Set an "X-Mailer" clause to aid identification of servers that have processed the message.
101. Perform a special action on a rule match based on RegEx matching with MAIL and RCPT clauses and IP matching on the source and destination IP address. If a match is found, then either refused a protocol level, delay at protocol level, run an MML script or redirect to a different email address.
102. Matching MAIL, From and Sender clauses to ensure full compliance before message delivery.
103. An X-Defects clause may be added to the header of a message indicating any specific issues with MIME headers within a message. Given the results of checks, messages may be rejected, returned, etc. on the basis of this analysis.
104. POP3 dynamic IP Deny List prevents logon from given IP if repeated logon failures occur.
105. IMAP4 dynamic IP Deny List prevents logon from given IP if repeated logon failures occur.



106. Set the maximum number of recipients per message. Set different number of maximum recipients on a per-incoming IP address basis. Define message given to sending server when maximum reached.
107. Enable or disable the use of the ESMTP XTRN command.
108. Define maximum CNAME depth to prevent incorrect DNS configuration causing server instability.

### ***Mail sent by SMTP - Per domain protection***

GMS supports multiple internet domains – for example, the domains gordano.com and nmail.com can be managed as completely separate entities without respective users being aware that the software is supporting two domains. Each domain may have specific protection facilities that may be defined independently from any other domain on the system.

1. Each domain has filtering based on the number of instances of one or more words within messages. Messages that match the filter may be failed with an SMTP code, delayed with an SMTP code, forwarded to specific address or copied to another address.
2. Set the maximum outbound message size.
3. Addition of custom footer to messages allows administrator to include details of administrative contact in case of accusations of spam.
4. On a per-domain setting, the virus scanner will decode all MIME, uuencode and binhex attachments for checking. When infected email arrives the virus scanner gives options to: return the message with user defined text (optionally disinfecting it), reject the message, redirect the message (optionally disinfecting it), deliver the message as usual (optionally disinfecting it) or place the message into the quarantine folder for further examination.
5. If virus scan fails for any reason optionally reject the email and notify the postmaster that an error occurred.
6. Request that the virus scanner scans inline text as well as all attachments.
7. Define the maximum size of messages allowed to be delivered to any account in the domain (with the option of a specific account over-ride).

### ***Mail sent by SMTP - Per user protection - set by administrator***

1. Set the maximum outbound message size.
2. On a per-user setting, the virus scanner will decode all MIME, uuencode and binhex attachments for checking. When infected email arrives the virus scanner gives options to: return the message with user defined text (optionally disinfecting it), reject the message, redirect the message (optionally disinfecting it), deliver the message as usual (optionally disinfecting it) or place the message into the quarantine folder for further examination.
3. If virus scan fails for any reason optionally reject the email and notify the postmaster that an error occurred.



4. Request that the virus scanner scans inline text as well as all attachments.
5. Enable or disable an account (or accounts).
6. Define the maximum size of messages allowed to be delivered to a specific account (or accounts).

### ***Mail sent by SMTP - Per user protection - set by user***

1. Check if sender's address is in any of the address books, if not found then if any of the following checks fail then 1. Deliver the message as usual; 2. Transfer the message to any folder; 3. Redirect the message to any other account (local or remote); 4. Delete the message; 5. Copy the message to another folder; 6. Forward the message to another account (local or remote); 7. Reply with a user defined message; 8. Delivery a copy via SMS; or 9. Deliver a copy via Instant Messaging. The specific "Spam" checks review messages that: 1. Are addressed to me (or any of my personalities); 2. Have no reply address specified; 3. Have a reply address does not match from address; 4. Have a subject is all capitals; and/or 5. Have no subject .
2. Check if sender's address is in any of the address books, if not found then if any of the following checks fail then 1. Deliver the message as usual; 2. Transfer the message to any folder; 3. Redirect the message to any other account (local or remote); 4. Delete the message; 5. Copy the message to another folder; 6. Forward the message to another account (local or remote); 7. Reply with a user defined message; 8. Delivery a copy via SMS; or 9. Deliver a copy via Instant Messaging. The specific checks can provide regex matching on any message header clause or part of the message body.
3. Check is the sender is in the user's block list. If so, either delete the message or transfer it into the user's quarantine folder.
4. Addition of custom footer to messages allows user to include details of administrative contact in case of accusations of spam.
5. Confirmation Process<sup>2</sup>. This queries an "unknown" sender of an email to acknowledge that they want to send an email *before* it appears in the recipient's inbox. Following confirmation, the sender's details can be added automatically to a specified address book and thus become "known". The next time they send a message they are recognised as a known sender and so no further acknowledgement is required.
6. Transitory Email Addresses<sup>3</sup>. A user can request a transitory email address from the system which they can then use on web sites, posting to news groups, etc. The address will last a specific time and when it expires any email to that address will no longer be accepted.

---

<sup>2</sup> Please see white paper for full details of this patent pending process.

<sup>3</sup> Patent pending.



## ***Protection of mail collection services***

Gordano provides access to messages through the POP3, IMAP4, HTTP and HTTPS protocols. Each of these services is protected in order to prevent attacks from compromising the system. Some of the checks include are:

1. Requirement for user name and password before results of authentication request are processed (this prevents miscreants attempting to identify valid accounts).
2. Increasing delay of logon for each failed authentication attempt.
3. The option to require the use of non-plaintext authentication methods (e.g. APOP).
4. Support for APOP, MD5, CRAM-MD5 password encryption.
5. Automatic password expiry and implementation of password type policy including maximum number of times for any character, minimum length, requirement for special characters, alpha numerics, etc..
6. Control the maximum number of sessions allowed per source IP address.

## ***Protection using HTTP(S) and FTP proxy services***

The proxy services allow both forward and reverse proxy of http and ftp protocols. In addition they will cache pages in order to speed up subsequent requests of the same page. The forward proxy allows a web browser to use the proxy to access any other web server on the Internet. The reverse proxy will accept a connection from any location and forward it to a given, specified web server (with caching this allows GMS to be used as part of a load balancing proxy array).

1. Require user authentication using basic or digest algorithms.
2. Prevent access to specified sites using RegEx matching.
3. Allow access to specified sites using RegEx matching.
4. Prevent specific attachment types from being downloaded.
5. Scan all files downloaded for viruses (using the system settings for actions on finding a virus).
6. Option to allow compression of pages being sent on bound by the server.
7. Allow automatic authentication to external sites.
8. Set the maximum size of the cache for the proxy.

## ***System protection***

GMS includes code to allow the system resource use to be managed and controlled. These are automatic but can be modified if required. Examples include:

1. Set the maximum inbound SMTP bandwidth so utilisation of network resources is automatically limited.
2. Set the maximum outbound SMTP bandwidth so utilisation of network resources is automatically limited.



3. Limit Disk space used on a per-user, per-domain or system basis. When maximum reached, delay email until enough is deleted to allow service to be continued.
4. Delete all messages marked as read when a user disconnects from POP3.
5. Define maximum number of log files and space used and action to take when log files reach the limit.
6. Automatically compress log files (including transaction and message archive logs).



## **APPENDIX B: GLOSSARY**

This glossary explains some of the terms and facilities within each of the options listed in Appendix A.

<b>Address books</b>	There are a large number of address books within GMS – the system, internet domain, user, quarantine, block list, etc. In addition, the user may define any number of their own address books. Filters will work using any or none of the address books.
<b>Address Harvesting</b>	Those wishing to send UCE need to discover email addresses in order to send the messages. A way to generate such email addresses is to log onto a POP3 or IMAP4 server and try different user names. If the server responds with a command showing that the account is active and address identified then the details are captured by the harvester. GMS software always accepts the username and returns the acceptable or non-acceptable status after a password has been entered.
<b>IP Matching Rules</b>	A space separated list of IP addresses or IP address ranges allowing bit masking, class addresses, inversion, and specific ranges of IPs to be defined. For example 1.1.1.* represents a C-Class address space from 1.1.1.0 to 1.1.1.255. 1.2.3.4/17 represents the address space 1.2.0.0 to 1.3.255.255.
<b>MIME</b>	The Internet Mail standard which defines how messages are composed. The specification identifies how attachments are to be added to messages and how messages can contain multiple sections (e.g. HTML and plain text).
<b>Quarantine</b>	There are several quarantine folders within GMS – the system, internet domain and user quarantine. Any messages places in quarantine may be forwarded, deleted, etc. as appropriate. In addition, the user's quarantine folder has a special option where messages older than a given age are automatically deleted when they log out.
<b>Regex</b>	A method of matching strings (e.g. domain names). Thus the string “*abc*” will match “abcdef” and “89sjabcd” but not “cbaks”. In addition to the wildcard (“*”), GMS also supports the wildcard single character, character ranges and “not” function.
<b>Mail Meta Language (MML)</b>	This is a language written by Gordano for processing email messages. As well as providing the full web interface within GMS, it can process and manipulate



email messages very effectively and efficiently. MML provides full control of GMS and allows practically anything to be done to an email message while it passes through the delivery stages. The language is object orientated and based on run-time scripting like C or awk.

**Tar-pitting**

An attack that is designed to exhaust a server's resources completely by continually issuing "OK" or "idle" commands. This type of attack can affect inbound SMTP, POP3, IMAP4 protocols and outbound SMTP if servers are not protected against it.

**Web Proxy Bypass Attack**

Some ISP's prevent their users from sending email other than through their own mail servers. This means that they can log and verify all messages that are sent. A program has been written to encapsulate the SMTP protocol into a standard HTTP proxy request that results in the ISP's proxy server unwittingly connecting to an SMTP server and delivering a message. Users are known to use this method to evade detection by their ISP.