Network Working Group
Request for Comments: 5383
BCP: 143
Category: Best Current Practice

R. Gellens
Qualcomm

October 2008

Deployment Considerations for Lemonade-Compliant Mobile Email

Status of This Memo

Abstract

This document discusses deployment issues and describes requirements for successful deployment of mobile email that are implicit in the IETF lemonade documents.

Table of Contents

1. Introduction

The IETF lemonade group has developed a set of extensions to IMAP and Message Submission, along with a profile document that restricts server behavior and describes client usage [PROFILE].

Successful deployment of lemonade-compliant mobile email requires various functionality that is generally assumed and hence not often covered in email RFCs. This document describes some of these additional considerations, with a focus on those that have been reported to be problematic.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

3. Ports

Both IMAP and Message Submission have been assigned well-known ports [IANA] that MUST be available. IMAP uses port 143. Message Submission uses port 587. It is REQUIRED that the client be able to contact the server on these ports. Hence, the client and server systems, as well as any intermediary systems, MUST allow communication on these ports.

Historically, Message User Agents (MUAs) have used port 25 for Message Submission, and [SUBMISSION] does accommodate this. However, it has become increasingly common for ISPs and organizations to restrict outbound port 25. Additionally, hotels and other public accommodations sometimes intercept port 25 connections, regardless of the destination host, resulting in users unexpectedly submitting potentially sensitive communications to unknown and untrusted third-party servers. Typically, users are not aware of such interception. (Such interception violates [FIREWALLS] and has many negative consequences.)

Due to endemic security vulnerabilities in widely deployed SMTP servers, organizations often employ application-level firewalls that intercept SMTP and permit only a limited subset of the protocol. New extensions are therefore more difficult to deploy on port 25. Since lemonade requires support for several [SUBMISSION] extensions, it is extremely important that lemonade clients use, and lemonade servers listen on, port 587 by default.

In addition to communications between the client and server systems, lemonade requires that the Message Submission server be able to establish a TCP connection to the IMAP server (for forward-without-download). This uses port 143 by default.

Messaging clients sometimes use protocols to store, retrieve, and update configuration and preference data. Functionality such as setting a new device to use the configuration and preference data of another device, or having a device inherit default configuration data from a user account, an organization, or other source, is likely to be even more useful with small mobile devices. Various protocols can be used for configuration and preference data; most of these protocols have designated ports. It is important that clients be able to contact such servers on the appropriate ports. As an example, one protocol that can be used for this purpose is [ACAP], in which case port 674 needs to be available.

Note that systems that do not support application use of [TCP] on arbitrary ports are not full Internet clients. As a result, such systems use gateways to the Internet that necessarily result in data integrity problems.

4. TCP Connections

Both IMAP and Message Submission use [TCP]. Hence, the client system MUST be able to establish and maintain TCP connections to these servers. The Message Submission server MUST be able to initiate a connection to the IMAP server. Support for application use of [TCP] is REQUIRED of both client and server systems.

The requirements and advice in [HOST-REQUIREMENTS] SHOULD be followed.

Note that, for environments that do not support application use of [TCP] but do so for HTTP, email can be offered by deploying webmail. Webmail is a common term for email over the web, where a server speaks HTTP to the client and an email protocol (often IMAP) to the mail store. Its functionality is necessarily limited by the capabilities of the web client, the webmail server, the protocols used between the webmail server and the client (HTTP and a markup language such as HTML), and between the webmail server and the mail store. However, if HTTP is all that is available to an application, the environment is by definition limited and thus, functionality offered to the user must also be limited, and can’t be lemonade compliant.

4.1. Lifetime

In this document, "idle" refers to the idle time, as in the "established connection idle-timeout" of [BEHAVE-TCP], while "duration" refers to the total time that a TCP connection has been established.

The duration of the TCP connections between the client and server systems for both IMAP and Message Submission can be arbitrarily long. The client system, the server, as well as all intermediate systems MUST NOT terminate these TCP connections simply because of their duration (that is, just because of how long they have been open).

Lemonade depends on idle timers being enforced only at the application level (IMAP and Message Submission): if no data is received within a period of time, either side MAY terminate the connection as permitted by the protocol (see [SUBMISSION] or [IMAP]). Since IMAP permits unsolicited notifications of state changes, it is reasonable for clients to remain connected for extended periods with no data being exchanged. Being forced to send data just to keep the connection alive can prevent or hinder optimizations such as dormancy mode (see Section 5).

Two hours is a fairly common configuration timeout at middleboxes. That is, there are a number of sites at which TCP connections are torn down by the network two hours after data was last sent in either direction (for example, REQ-5 in [BEHAVE-TCP]). Thus, lemonade clients and servers SHOULD make sure that, in the absence of a specific configuration setting that specifies a longer maximum idle interval, the TCP connection does not remain idle for two hours. This rule ensures that, by default, lemonade clients and servers operate in environments configured with a two-hour maximum for idle TCP connections. Network and server operators can still permit IMAP connections to remain idle in excess of two hours and thus increase the benefits of dormancy, by configuring lemonade clients and servers, and network equipment, to allow this.

It has been reported that some networks impose duration time restrictions of their own on TCP connections other than HTTP. Such behavior is harmful to email and all other TCP-based protocols. It is unclear how widespread such reported behavior is, or if it is an accidental consequence of an attempt at optimizing for HTTP traffic, implementation limitations in firewalls, NATs, or other devices, or a deliberate choice. In any case, such a barrier to TCP connections is a significant risk to the increasing usage of IETF protocols on such networks. Note that TCP is designed to be more efficient when it is

used to transfer data over time. Prohibiting such connections thus imposes hidden costs on an operator's network, forcing clients to use TCP in inefficient ways. One way in which carriers can inadvertently force TCP connections closed, resulting in users wasting packets by reopening them, is described in Section 7.

Note that systems remain able to terminate TCP connections at any time based on local decisions, for example, to prevent overload during a denial-of-service attack. These mechanisms are permitted to take idle time into consideration and are not affected by these requirements.

4.2. Maintenance during Temporary Transport Loss

TCP is designed to withstand temporary loss of lower-level connectivity. Such transient loss is not uncommon in mobile systems (for example, due to handoffs, fade, etc.). The TCP connection SHOULD be able to survive temporary lower-level loss when the IP address of the client does not change (for example, short-duration loss of the mobile device's traffic channel or periods of high packet loss). Thus, the TCP/IP stack on the client, the server, and all intermediate systems SHOULD maintain the TCP connection during transient loss of connectivity.

In general, applications can choose whether or not to enable TCP keep-alives, but in many cases are unable to affect any other aspect of TCP keep-alive operation, such as time between keep-alive packets, number of packets sent before the connection is aborted, etc. In some environments, these are operating system tuning parameters not under application control. In some cases, operational difficulties have been reported with application use of the TCP keep-alive option, which might be the result of TCP implementation differences or defects specific to a platform. Lemonade client and server systems SHOULD NOT set the TCP keep-alive socket option unless operating in environments where this works correctly and such packets will not be sent more frequently than every two hours. Application-level keep-alives (such as IMAP NOOP) MAY be used instead of the TCP keep-alive option.

Client, server, and intermediate systems MUST comply with the "Destination Unreachable -- codes 0, 1, 5" text in Section 4.2.3.9 of [HOST-REQUIREMENTS], which states "Since these Unreachable messages indicate soft error conditions, TCP MUST NOT abort the connection".

5. Dormancy

Cellular data channels are connection-oriented (they are brought up or down to establish or tear down connections); it costs network resources to establish connections. Generally speaking, mobile device battery charges last longer when the traffic channel is used less.

Some mobile devices and networks support dormant mode, in which the traffic channel is brought down during idle periods, yet the PPP or equivalent level remains active, and the mobile retains its IP address.

Maintenance of TCP connections during dormancy SHOULD be supported by the client, server, and any intermediate systems, as described in Sections 4.1 and 4.2.

Sending packets just to keep the session active causes unnecessary channel establishment and timeout; with a long-idle TCP connection, this would periodically bring up the channel and then let it idle until it times out, again and again. However, in the absence of specific configuration information to the contrary, it is necessary to do this to ensure correct operation by default.

6. Firewalls

New services must necessarily have their traffic pass through firewalls in order to be usable by corporate employees or organization members connecting externally, such as when using mobile devices. Firewalls exist to block traffic, yet exceptions must be made for services to be used. There is a body of best practices based on long experience in this area. Numerous techniques exist to help organizations balance protecting themselves and providing services to their members, employees, and/or customers. (Describing, or even enumerating, such techniques and practices is beyond the scope of this document, but Section 8 does mention some.)

It is critical that protocol design and architecture permit such practices, and not constrain them. One key way in which the design of a new service can aid its secure deployment is to maintain the one-to-one association of services and port numbers.

One or more firewalls might exist in the path between the client and server systems, as well as between the Message Submission and IMAP servers. Proper deployment REQUIRES that TCP connections be possible from the client system to the IMAP and Message Submission ports on the servers, as well as from the Message Submission server to the IMAP server. This may require configuring firewalls to permit such usage.

Firewalls deployed in the network path MUST NOT damage protocol traffic. In particular, both Message Submission and IMAP connections from the client MUST be permitted. Firewalls MUST NOT partially block extensions to these protocols, such as by allowing one side of an extension negotiation, as doing so results in the two sides being out of synch, with later failures. See [FIREWALLS] for more discussion.

Application proxies, which are not uncommon mechanisms, are discussed in [PROXIES].

6.1. Firewall Traversal

An often-heard complaint from those attempting to deploy new services within an organization is that the group responsible for maintaining the firewall is unable or unwilling to open the required ports. The group that owns the firewall, being charged with organizational network security, is often reluctant to open firewall ports without an understanding of the benefits and the security implications of the new service.

The group wishing to deploy a new service is often tempted to bypass the procedure and internal politics necessary to open the firewall ports. A tempting kludge is to tunnel the new service over an existing service that is already permitted to pass through the firewall, typically HTTP on port 80 or sometimes SMTP on port 25. Some of the downsides to this are discussed in [KLUDGE].

Such a bypass can appear to be immediately successful, since the new service seems to deploy. However, assuming the network security group is competent, when they become aware of the kludge, their response is generally to block the violation of organizational security policy. It is difficult to design an application-level proxy/firewall that can provide such access control without violating the transparency requirements of firewalls, as described in [FIREWALLS]. Collateral damage is common in these circumstances. The new service (which initially appeared to have been successfully deployed) as well as those existing services that were leveraged to tunnel the new service, become subject to arbitrary and unpredictable

failures. This encourages an adversarial relationship between the two groups, which hinders attempts at resolution.

Even more serious is what happens if a vulnerability is discovered in the new service. Until the vulnerability is corrected, the network security group must disable both the new service and the (typically mission-critical) existing service on which it is layered.

An often-repeated truism is that any computer that is connected to a network is insecure. Security and usefulness are both considerations, with organizations making choices about achieving acceptable measures in both areas. Deploying new services typically requires deciding to permit access to the ports used by the service, with appropriate protections. While the delay necessary to review the implications of a new service may be frustrating, in the long run, it is likely to be less expensive than a kludge.

7. NATs

Any NAT boxes that are deployed between client and server systems MUST comply with REQ-5 in [BEHAVE-TCP], which requires that "the value of the 'established connection idle-timeout' MUST NOT be less than 2 hours 4 minutes".

See Section 5 for additional information on connection lifetimes.

Note that IMAP and Message Submission clients will automatically re-open TCP connections as needed, but it saves time, packets, and processing to avoid the need to do so. Re-opening IMAP and Message Submission connections generally incurs costs for authentication, Transport Layer Security (TLS) negotiation, and server processing, as well as resetting of TCP behavior, such as windows. It is also wasteful to force clients to send NOOP commands just to maintain NAT state, especially since this can defeat dormancy mode.

8. Security Considerations

There are numerous security considerations whenever an organization chooses to make any of its services available via the Internet. This includes email from mobile clients.

Sites concerned about email security should perform a threat analysis, get relevant protections in place, and then make a conscious decision to open up this service. As discussed in Section 6.1, piggybacking email traffic on the HTTP port in an attempt to avoid making a firewall configuration change to explicitly permit mobile email connections would bypass this important step and reduce the overall security of the system.

Organizations deploying a messaging server "on the edge" (that is, accessible from the open Internet) are encouraged to choose one that has been designed to operate in that environment.

This document does not attempt to catalogue either the various risks an organization might face or the numerous techniques that can be used to protect against the risks. However, to help illustrate the deployment considerations, a very small sample of some of the risks and countermeasures appear below.

Some organizations are concerned that permitting direct access to their mail servers via the Internet increases their vulnerability, since a successful exploit against a mail server can potentially expose all mail and authentication credentials stored on that server, and can serve as an injection point for spam. In addition, there are concerns over eavesdropping or modification of mail data and authentication credentials.

A large number of approaches exist that can mitigate the risks while allowing access to mail services via mobile clients.

Placing servers inside one or more DMZs (demilitarized zones, also called perimeter networks) can protect the rest of the network from a compromised server. An additional way to reduce the risk is to store authentication credentials on a system that is not accessible from the Internet and that the servers within the DMZ can access only by sending the credentials as received from the client and receiving an authorized/not authorized response. Such isolation reduces the ability of a compromised server to serve as a base for attacking other network hosts.

Many additional techniques for further isolation exist, such as having the DMZ IMAP server have no mail store of its own. When a client connects to such a server, the DMZ IMAP server might contact the authentication server and receive a ticket, which it passes to the mail store in order to access the client’s mail. In this way, a compromised IMAP server cannot be used to access the mail or credentials for other users.

It is important to realize that simply throwing an extra box in front of the mail servers, such as a gateway that may use HTTP or any of a number of synchronization protocols to communicate with clients, does not itself change the security aspects. By adding such a gateway, the overall security of the system, and the vulnerability of the mail servers, may remain unchanged or may be significantly worsened. Isolation and indirection can be used to protect against specific risks, but to be effective, such steps need to be done after a threat analysis, and with an understanding of the issues involved.

Organizations SHOULD deploy servers that support the use of TLS for all connections and that can be optionally configured to require TLS. When TLS is used, it SHOULD be via the STARTTLS extensions rather than the alternate port method. TLS can be an effective measure to protect against specific threats, including eavesdropping and alteration, of the traffic between the endpoints. However, just because TLS is deployed does not mean the system is "secure".

Attempts at bypassing current firewall policy when deploying new services have serious risks, as discussed in Section 6.1.

It’s rare for a new service to not have associated security considerations. Making email available to an organization’s members using mobile devices can offer significant benefits.

9. Acknowledgments

Chris Newman and Phil Karn suggested very helpful text. Brian Ross and Dave Cridland reviewed drafts and provided excellent suggestions.

10. Normative References

[BEHAVE-TCP] Guha, S., Ed., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

[HOST-REQUIREMENTS] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[IANA] IANA Port Number Registry, <http://www.iana.org/assignments/port-numbers>

[TCP] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

11. Informative References

[ACAP] Newman, C. and J. Myers, "ACAP -- Application Configuration Access Protocol", RFC 2244, November 1997.

[FIREWALLS] Freed, N., "Behavior of and Requirements for Internet Firewalls", RFC 2979, October 2000.

[IMAP] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[KLUDGE] Moore, K., "On the use of HTTP as a Substrate", BCP 56, RFC 3205, February 2002.

[PROFILE] Maes, S. and A. Melnikov, "Internet Email to Support Diverse Service Environments (Lemonade) Profile", RFC 4550, June 2006.

[PROXIES] Chatel, M., "Classical versus Transparent IP Proxies", RFC 1919, March 1996.

[SUBMISSION] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.

Author's Address

Randall Gellens
QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121

EMail: randy@qualcomm.com

Full Copyright Statement

Intellectual Property