Does Your Enterprise Have A SMTP Onion?

by Lawrence Kearney

Often overlooked SMTP infrastructure components can easily be compared to onions. Most mature slowly, each layer builds on and obscures the last, and finally, they can make you cry. This often leads to a don't touch it if it ain't broke mentality within Information Technology (IT) departments where SMTP services are concerned. As a result more organisations have SMTP onions than you would think. In order to manage and improve these services you need to "solider on" and start peeling back your onion. Employing new design and management strategies will help most organisations optimise existing infrastructure investments. If done properly, the "tears" you produce may just be tears of joy.

Where is SMTP 2.0 ?

The features provided by most internet service protocols eventually align with technology advancements and usage trends. For example, HTML is just now seeing it's most significant re-tooling in over a decade. Version 5 touts native support for the rich web delivered content absent from version 4. Application developers, multimedia providers and eLearning systems had to develop complimentary technology to provide that content, and they started doing it almost a decade ago.

Frustratingly, SMTP has followed an even slower development curve than HTML. Even though remarkable feature enhancements were defined in 1995 (referring to extended SMTP [ESMTP]) those features were not widely implemented by SMTP software providers until 2003. Today, the rapid consumerism of IT has SMTP struggling to stay viable. Whether we like it or not consumer experiences shape their expectations of business systems. Subsequently, those expectations impact how providers of messaging services deal with slow development curves.

LAWRENCE KEARNEY joined Georgia Health Sciences University in 1998 and is currently a System Support Specialist. He previously worked in the business sector in New York.

He has experience of many Novell and SUSE Linux solutions and co-authored the Novell GroupWise 8 Best Practice Wiki.

Current messaging service needs are out-distancing the SMTP protocol in complexity. As a result we need to improve the tools we use to satisfy those complexities. Other solutions are already being used to accomplish this. For example, Research in Motion® (RIM) uses it's Blackberry Internet Service® (BIS) product to marry POP, IMAP and SMTP with it's service layers and features to satisfy the "on demand" messaging services it's customer's want. Satisfying the messaging needs of your organisation's users will require a similar effort. This article describes how you can implement your own on premises "SMTP 2.0" architecture to better fulfil your organisation's messaging needs.

"Layers" that can be used to implement a solution

Organisations with on-premises messaging systems are comparing their needs with hosted or other off-premises offerings when considering upgrade or migration options. Most understand that calculating the cost effectiveness of a solution goes well beyond the cost of deployment. The administration and usability of those options by staff and users are always major decision points. This is where relatively low cost changes to your existing infrastructure can really shine.

Most enterprises have some or all of the items in the following list. The redeployment or addition of these items whilst making clever architecture design choices can make a world of difference in your messaging environment that will be noticed by all.

Relay servers offer an additional layer of security and I/O abstraction from your messaging system and often augment it with additional features.

Often this layer incorporates unsolicited bulk email filtering and anti-virus components. Handling this I/O before it reaches your internal mail handling processes is key. The goal is to optimise your organisation's computational resources to its benefit. In most cases over 90% of mail sent to your system(s) can be filtered out. That's' a lot of processor and storage controller work saved.

Also consider the additional features these appliances or servers may bring to the table. Enhanced logging, policy based message management, and other auditing and analysis tools to name a few. Ensuring they have enterprise features built-in such as fail over, clustering or load balancing capabilities will increase their usefulness significantly.

GroupWise Internet Agents (GWIA) can make use of

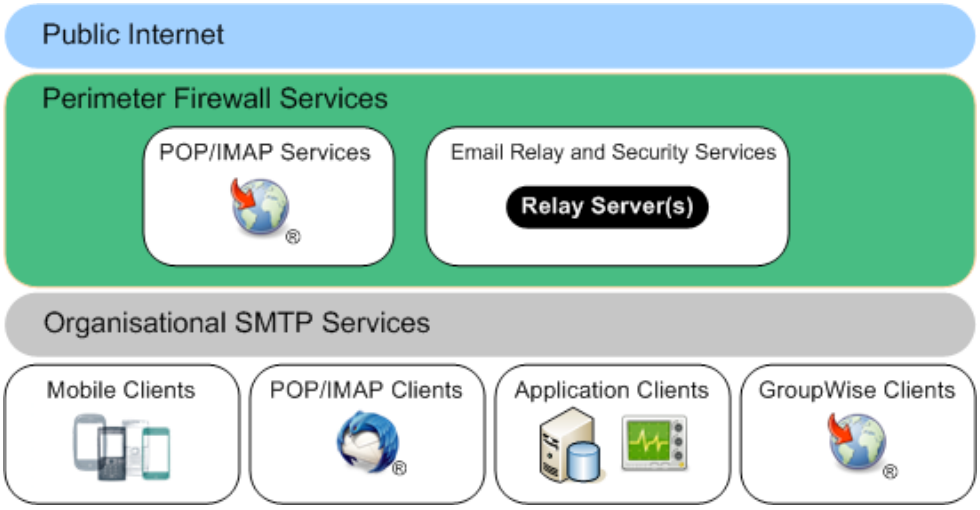

Figure 1: A typical enterprise SMTP onion

multiple relay servers. If the primary becomes unavailable it will communicate with a configured secondary or tertiary relay server.

Access Control Lists (ACL) prevent unauthorised access to services. They can also be used to implement service demarcations for Quality of Service (QoS) purposes.

ACLs can be very helpful for optimising services. You need to plan service demarcation placement and manage the resources within them. For example, implementing separate SMTP resources for native GroupWise clients, application clients and third party POP and IMAP clients are typical and realistic demarcations for most organisations. Ensuring that the resources allocated to a particular service silo are dedicated to that service is the goal.

The primary challenge here will be harvesting and validating the internal (and likely undocumented) application network addresses using your current services. Once this information is known it can be used to build service specific ACLs. Fortunately, GWIA's support the necessary ACL features to accomplish this type of QoS implementation.

Load Balancers optimise network traffic and distribute computational workloads. They can also be used as aggregation points for ACL configurations and Secure Socket Layer (SSL) certificates.

Hardware load balancers offer flexible architecture design choices within the enterprise, especially when multiple GroupWise agents are dedicated to a specific service silo. Often ACL records and SSL certificates for multiple agents can be managed in one place.

Many load balancers include "SSL offloading" features which allow the appliance to offer the SSL certificate associated with the service, rather than the agent(s), to clients. Using SSL offloading saves on certificate costings, administrative overhead and server workloads. Additionally, the usability of the services by clients that have issues using Transport Layer Security (TLS) security for messaging services will likely improve.

Also, consider using software load balancing where appropriate. Load balancing with DNS and other tools, such as your perimeter appliances or your email software, can compliment hardware load balancing. When used sensibly, solution performance and cost effectiveness increase.

Clustering software provides high availability services. Moreover it allows you to increase your SMTP service instances without requiring additional server and operating system instances.

Clustering gateways like the GWIA make them flexible relative to their host platform and deployment style. If your agents are not currently deployed on Linux you might want to take the opportunity to do so. Frankly, the GWIA simply performs better on Linux. Using a Novell Open Enterprise Server (OES) Linux cluster or a SUSE Linux Enterprise Server (SLES) High Availability (HA) cluster where new GWIA instances can be provisioned are great options for hosting these services. Clustering options make better use of your hardware in addition to providing high availability services.

Internal human resource skills required for a successful deployment range from non-technical to leadership. Non-technical tasks such as inventorying service information, documenting architecture and implementing new business processes must not be overlooked when planning to succeed.

Understanding the usage trends of your services allows you to gauge the resources you will need to improve them not to mention identifying what can be responsibly culled out. Enterprise service management software can help identify network address and use trends within your infrastructure very accurately. Do use it if the option is available to you. However, if that option is not available, agent log files can provide the necessary info. This is a difficult task to perform manually but a successful

deployment may require the effort.

Isolating messaging workloads will almost certainly require ACL implementations, and likely new policies for their use. Working with staff to implement business processes for ACL provisioning and service access policies will ensure the deployment has leadership and organisational support.

Once the needed resources take shape, having clear deployment documentation for key decision makers will help move things forward. Having leadership "buy in" and support will be necessary to approve architecture changes in your environment. Building these relationships doesn't come naturally to most technicians, but you should try. Practice will make perfect.

So, now we peel back the onion …

Phase one: The easiest service silo to optimise is for native GroupWise client services. Even if the existing GWIA's are in use, new ones can be provisioned. These new agents can then be assigned to domains where your users reside transparently. This implements the initial service demarcation with relatively little technical effort. Figure 2 details this approach.

Beyond the network perimeter, the Domain Name Service (DNS) is used to match the Mail Exchanger (MX) record priorities that route inbound mail to the perimeter relay servers. This performs some basic but effective load balancing as mail enters the enterprise. Within the perimeter, the relay servers are configured to distribute inbound mail to specified GWIA's with equal priority as well.

Multiple new GWIA's have been provisioned with ACLs allowing them to only accept inbound SMTP connections from the relay servers in the perimeter. This effectively limits these agents to handling messages transferred to them by their parent domain Message Transfer Agents (MTA) and the relay servers. All inbound messages are eventually handed to a GroupWise MTA by a GWIA to begin routing them to users. Any MTA can usually route a message to it's destination domain within one or two hops. GroupWise MTA's can be used as very effective software load balancers in this respect.

For outbound mail, each domain is assigned to a preferred and an alternate GWIA. This ensures each domain supporting users has a dedicated gateway or that one is always available to a domain should its preferred GWIA fail. Additionally, each GWIA is configured to use multiple relay server addresses in a differing order. This will allow a certain amount of load balancing and redundancy for outbound mail should a relay server fail.

Service benchmarks should improve significantly after the completion of phase one. Transparently, email I/O has been optimised and client feature thresholds can be increased as a result. User and leadership support for additional changes is easier to secure in this light.

Phase two: Since application clients are still using the previous SMTP services without interruption they can be left as is for now. Implementing and publicising a change for users is often an easier sale than upsetting business operations. Changing enterprise application mechanics can do just that. So, the service silo for third party POP and IMAP clients is the next best candidate for change. Initially desktop and mobile clients can share the same silo.

Again, new GWIA's can be provisioned and load balanced with hardware using a single service address. All participating GWIA's should be configured with ACLs that limit communication to the load balancer and perimeter relay servers. Using a single SSL certificate on the load balancer will securely unify all agents from a network identity perspective. Once this silo is provisioned, users can simply be migrated to it using communication tools or hammers. The previous agents and resources can be

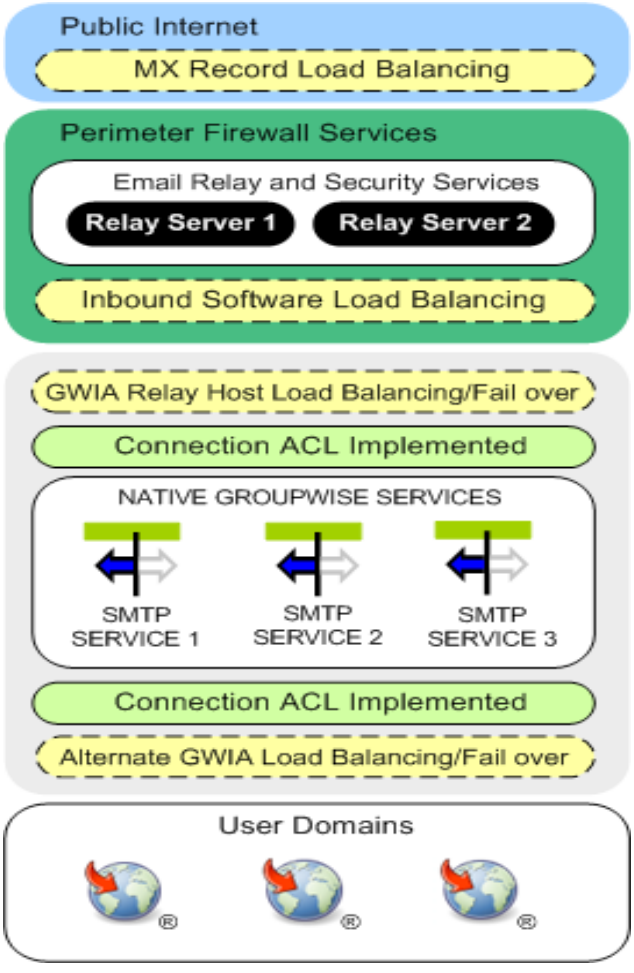

Figure 2: Dedicated GroupWise client SMTP services

reclaimed and provisioned for application client resources.

Phase three: The most cumbersome silo to deal with will be the one that is dedicated to applications. Make no mistake - most of your work will be done here. One of the keystones of this design is the separation of workloads. In this context, separating the resources dedicated to users and applications. These two workloads are very different, and isolating them can resolve the obvious and undiscovered resource contention issues.

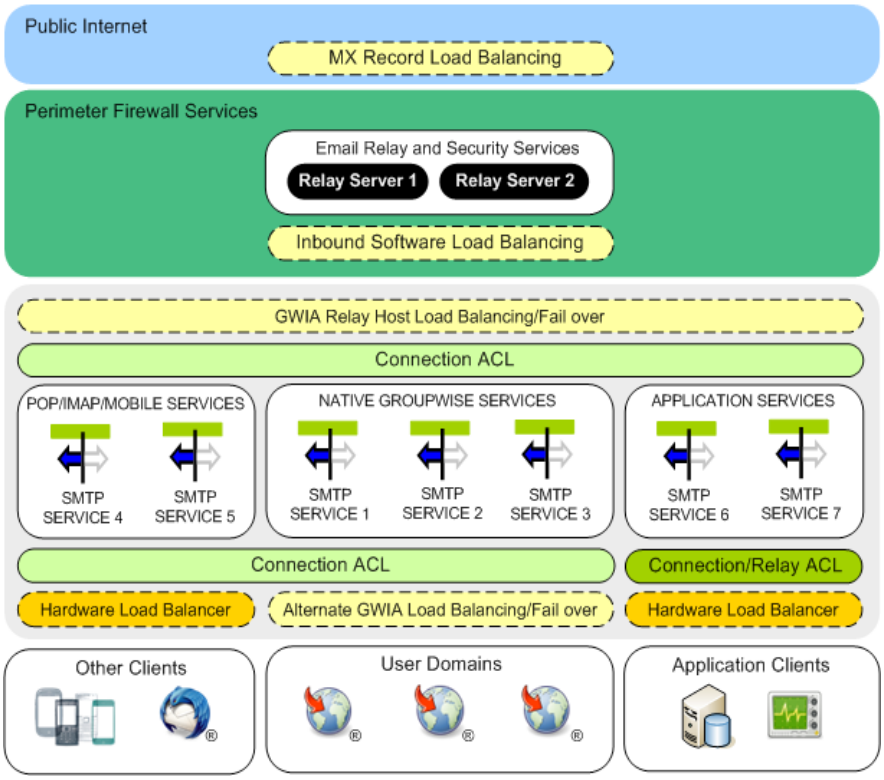

Figure 3: A much improved SMTP service infrastructure

Fortunately, the only GWIA(s) left standing are being used by these application clients already. Using multiple agents in a load balanced configuration is recommended here if your application messaging workload warrants it. So if you need additional agents provision and load balance them as well.

Your application client security model may not require SSL connectivity, but ACLs can be used here to your organisation's benefit. Whether applications can send mail internally, externally or both is important for service capacity, compliance and data leakage concerns. Remember, we're talking about unmanaged thin clients, SQL directives and Power Shell scripts that can send mail. Therefore, managing which automated processes and applications can make SMTP connections, or perform email relay operations using your resources is important. Initial ACLs can be built using the agent log files. It is possible to harvest the network address information and the messaging behaviours of the current application clients from them. Limiting incoming SMTP connections to authorised application clients and the perimeter relay servers should be the operational goal. This ACL list could also reside with the load balancer, so it can be managed in a single location. Email relay logic will most likely need to be maintained on each participating GWIA. Figure 3 details the architecture for Phases 2 and 3.

Where to go from here?

After consuming the information here, we are likely to turn our thoughts to mobile messaging workloads. Workforce demands for office portability and device support have organisations struggling to keep up with user work styles and productivity. However, many simply do not have enough information about their services, or their consumption, to adapt their infrastructures to meet their workforce needs.

A service infrastructure like the one detailed here has benefits beyond performance and usability improvements. Reliable metrics for service auditing, trending, capacity monitoring and capacity management are available as a result. This level of data quality can provide organisational agility, allowing cost effective scaling decisions to be made quickly. Including knowing whether an on-premises upgrade, hosted or a hybrid infrastructure solution fits their needs best.

Obviously infrastructure design is a very complex topic, and many technical details are omitted for brevity. Additional details and technical information for this deployment are available online.

Additional information on GroupWise infrastructure design can be found on the authors web site:
http://www.lawrencekearney.com/files/GW_SMTP_Infrastructure_Design.pdf

Additional information on GWIA ACL and email relay configurations can be found in the "Good and Bad Habits" section of the Novell GroupWise 8 Best Practices wiki:
http://wiki.novell.com/index.php/GroupWise_8_Best_Practices