The Futility of Common Firewall Policies:
An Experimental Demonstration

James E. Ries[1,2], M.S., Phillip V. Asaro[1], M.D.,
Arturo Guillen[2], B.S., Jordanka Ivanova[2], M.S.
[1]Department of Health Management and Informatics
[2]Department of Computer Engineering and Computer Science
University of Missouri, Columbia, Missouri

Many healthcare organizations utilize network "firewalls" to protect their networks from being accessed by unauthorized external entities. These same firewalls are also often configured to deny access to certain external services from within the internal network. The latter policy can be subverted through a "protocol tunneling" strategy, which has been implemented as a set of programs called "Firehole." Organizations should be aware of this potential weakness in their network security designs. Policies that deny external services to users should be carefully evaluated in light of clearly defined organizational goals.

Introduction

Corporate internal networks today are often connected to the Internet to provide employees with the ability to do research using the World Wide Web (WWW). These networks are typically protected from external access or attack via network "firewalls". Firewalls are special packet routers that allow or deny traffic (typically TCP/IP traffic, but possibly other kinds of network traffic) based on a variety of criteria. Increasingly, firewalls are used not only to regulate external access to intranets, but also to control internal access to the external Internet.

Healthcare organizations are particularly concerned with controlling access to their networks due to the variety of potentially sensitive information contained within computers connected to these networks. Private patient health information as well as financial information is likely to be present within typical healthcare intranets. While such information is usually protected by server-based security mechanisms, additional security is needed at the network level. The Technical Security section of the proposed Security and Electronic Signature Standards[1] under HIPAA addresses network security, stating that:

> *"Each organization that uses communications or networks would be required to protect communications containing health information that are transmitted electronically over open networks so that they cannot be easily intercepted and interpreted by parties other than the intended recipient, and to protect their information systems from intruders trying to access systems through external communication points."*

However, many organizations utilize network firewalls to control internal employee access to external Internet resources as well. We show that such attempts are largely futile given simple protocol tunneling techniques. In addition, we suggest that organizations clearly define and state the motivation behind policies that serve only to limit access to services without protecting patient health information or private organizational data. Clarification of the goals behind security policy will avoid unnecessary impediments to reasonable use and more importantly, help to assure that limited information-system resources are better focused on threats to information confidentiality.

Firewalls

Firewalls are barriers between a secure intranet and the open Internet[2]. A firewall may range from impermeable (allowing little or no traffic in or out) to porous (allowing most or all traffic in or out). To be truly useful, however, a firewall must allow *some* traffic in and/or out of an intranet.

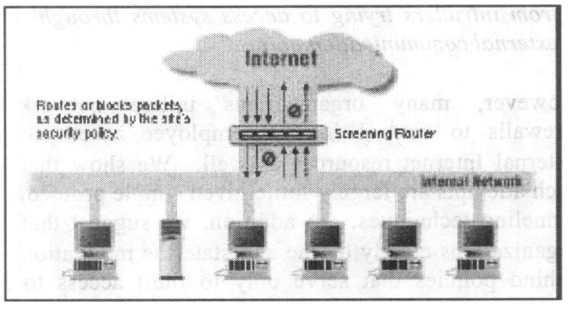
There is a proportional trade-off between level of service offered and level of security provided by firewalls. Thus, a completely secure intranet is one that is not connected to the Internet at all (i.e., connected to an impermeable firewall). Clearly, this extreme is severely limiting in that it provides no access to Internet services whatsoever. Conversely, a completely open intranet (or an intranet connected to an entirely porous firewall) provides easy and free access to Internet services such as WWW and email. This extreme is also clearly undesirable since internal

1067—5027/00/$5.00

network resources are subject to access and possible abuse by external entities.

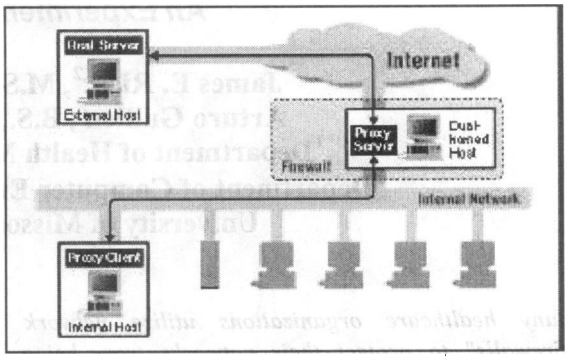
Firewalls are typically implemented as "screening routers". A screening router is a protocol router or gateway that selectively routes traffic based on various criteria. For example, a typical screening router may block inbound traffic traveling on TCP/IP port 23 (a port generally used for the Telnet service), but may allow both inbound and outbound traffic on TCP/IP port 80 (generally used for WWW service). Figure 1 depicts a screening router blocking outgoing traffic on one port and incoming traffic on another port. All other traffic is allowed to pass through.

Figure 1: Screening Router[2]



In addition to screening routers, firewalls often include "proxy servers". A proxy server is a device that conceptually straddles the firewall. It selectively allows traffic that normally could not penetrate a firewall to be allowed through the firewall. A proxy server is typically configured to operate on a particular protocol (e.g., the WWW's HyperText Transfer Protocol or HTTP). The proxy is able to access particular characteristics of the specific protocol that are hidden (transparent) from the screening router[3]. Thus, for example, a proxy server might allow most HTTP requests, but block requests for a Uniform Resource Locator (URL) that contains predetermined offensive words. Figure 2 illustrates a proxy server that is bypassing a firewall to allow access to an external server.

Figure 2: Proxy Server[2]



Common Firewall Policies

While no formal survey has been conducted to our knowledge, our own experiences in a variety of organizations indicate that three basic firewall security policies are common. The least restrictive utilizes only a screening router and allows any connection-oriented traffic which is initiated inside the firewall. This policy prevents external connections into the intranet from being created, while still allowing users inside the firewall to access most Internet services.

A second common policy builds on the first, by additionally restricting internally generated traffic to only a specified set of TCP/IP ports. For example, an administrator may allow HTTP traffic (which typically travels on port 80), but deny Telnet traffic (normally, port 23). This policy prevents users from connecting to unknown services, in that a user must request that the administrator open a particular port if a new application requires it. This may be useful in that it gives the administrator an opportunity to evaluate possible risks associated with the new service.

Finally, many network administrators require all Internet traffic to pass through a proxy server. Typically, in this configuration, only WWW traffic is allowed. This traffic must also pass through a proxy server for logging, and possibly for filtering as well. Such a configuration gives a network administrator seemingly great control over resources accessed by users. However, it is our assertion that this scheme can be overcome by technologies such as our own Firehole. Furthermore, such a tightly controlled Internet access policy may be viewed negatively by users, and may lead users to seek ways to subvert the system.

What is Firehole?

In order to demonstrate that allowing internal access to the WWW (through HTTP) is equivalent, with respect to security, to allowing internal access to all protocols, we developed a system called "Firehole". Firehole consists of a client application deployed inside a firewall (on the internal network), and a server application deployed outside the firewall in the open Internet.

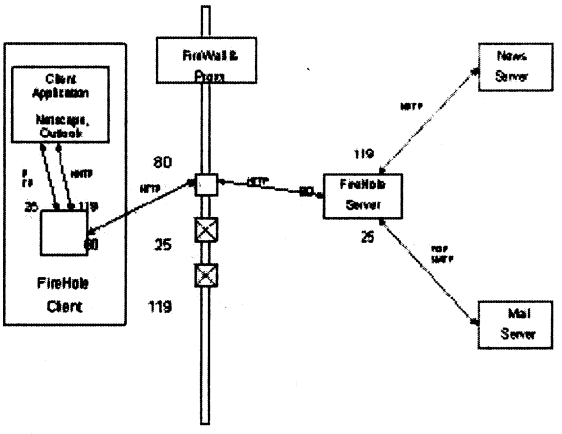
Firehole is essentially an "HTTP tunnel". That is, Firehole creates a "tunnel" through a firewall by taking a request for any Internet service and making it appear to be a request for a web page. It is arguable that Firehole does not compromise intranet security any more than general access to the WWW. That is, most any Internet service *could* be made available on the WWW directly. For example, email access can be (and often is) provided through a WWW interface. Firehole simply adds access to services that have not been explicitly web-enabled.

An application configured to use Firehole, sends native requests (e.g., POP or SMTP email type requests) to the Firehole Client. The Firehole Client encapsulates these requests in the trappings of the HTTP protocol and forwards them on to the proxy server, with the Firehole Server set as the destination.

The proxy server treats Firehole requests just as any request for a particular WWW resource, and forwards them to the Firehole Server (since it was specified as the destination). The Firehole Server then, de-encapsulates the message, contacts the real server, retrieves the result, encapsulates the result in HTTP, and returns it to the proxy server. The proxy server completes the circuit, by returning the result to the Firehole Client, who de-encapsulates the result and provides it back to the calling application.

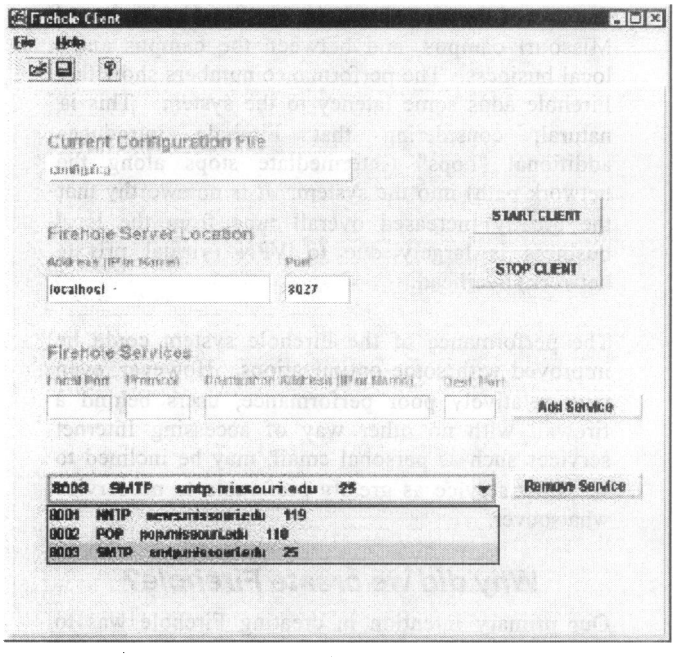
For example, Figure 3 shows two applications: an email and network news package interacting with the Firehole system. The application client (e.g., Outlook) talks to the Firehole Client as though it were the true destination server. The Firehole system simply passes requests through to the eventual server and sends back results. No special support for Firehole is required on either the application client or server.

Figure 3: Firehole Overview



Firehole is largely hidden from its users, in that the code doing encapsulation and de-encapsulation has little in the way of user interface. However, it is necessary for a user to initially configure Firehole to support the various servers he or she wishes to use. Firehole includes a Java Swing based configuration component show in Figure 4.

Figure 4: Firehole User Interface



The configuration component specifies the IP address (or mnemonic name) of the Firehole client, the ports on the Firehole Client which map to given real destination servers, their given ports, and the corresponding real protocol. Thus, the Firehole Client "knows" that a request received on port 8003

should eventually be routed to smtp.missouri.edu on port 25 as an SMTP type request.

The Firehole system requires a "plug-in" module in order to deal with new protocols. A protocol plug-in is a small Java module that is able to parse a given protocol. This approach allows Firehole to be extended to support any application-level protocol. Firehole currently supports POP3 (email receive), SMTP (email send), and NNTP (network news).

Firehole Performance

Table 1

Messages	Configuration	With Firehole	Without Firehole
0	Campus	8 seconds	2 seconds
2	Campus	12 seconds	6 seconds
0	Local Bus.	50 seconds	N/A
2	Local Bus.	70 seconds	N/A

Table 1 shows a few simple performance measures that we did using Firehole on the University of Missouri campus, and between the campus and a local business. The performance numbers show that Firehole adds some latency to the system. This is natural, considering that Firehole introduces additional "hops" (intermediate stops along the network path) into the system. It is noteworthy that the greatly increased overall time from the local business is largely due to VPN (virtual private network) overhead.

The performance of the Firehole system could be improved with some optimizations. However, even with relatively poor performance, users behind a firewall with no other way of accessing Internet services such as personal email, may be inclined to see slow service as greatly preferable to no service whatsoever.

Why did we create Firehole?

Our primary intention in creating Firehole was to demonstrate that restrictions based on Internet protocol can be easily circumvented. It should be pointed out that a group of graduate students created Firehole as a class semester project, and others could potentially create similar products (and may have done so already) in a short timeframe.

We hope to focus attention on the goals of information-system security policy. If communications through a firewall using Internet protocols other than HTTP truly represent a threat to an organization's information, then the possibility of protocol tunneling using an HTTP tunnel should be considered. We recognize that an organization may have other reasons for limiting access to Internet services, but any limiting policy should be aligned with clearly defined organizational goals.

Future Directions

An important step in the future development of the Firehole project will be to survey network administrators concerning their security policies. It will be important to understand motivations for limiting internal access, and to consider the ramifications of proliferation of Firehole-like technology. Conversely, a survey may expose a lack of understanding of some security issues by administrators. That is, it may be that administrators are taking the "block everything" approach because they do not fully understand firewall capabilities.

Another consideration (or perhaps concern) for future development of Firehole is encryption. Currently, all data including Firehole's proprietary protocol data is sent in clear text. However, administrators should be aware that Firehole-like technology could employ encryption to block proxy server logging techniques for user privacy reasons. This could be quite harmful, in that it would, as a by-product, also block automatic virus scanning and other potentially beneficial services that a proxy server may provide.

Conclusions

Firewalls can be used to prevent external access to internal network resources. This is a critical and appropriate use of firewall technology. However, firewalls are also often used to regulate the use of external Internet resources by those who reside inside the firewall. Such restriction may be viewed negatively by network users, and may even encourage users to attempt to defeat security schemes. Moreover, given the simple protocol tunneling techniques that Firehole demonstrates, restrictions based solely on the type of Internet protocol do little to enhance protection of patient health information or private organizational data.

With the current trend toward increasing attention to health information privacy, including the pending regulatory requirements of HIPAA, it is necessary for healthcare organizations to carefully allocate resources dedicated to information-system security.

Unless other organizational goals are served by such policies, they should be reconsidered in light of the information we have presented.

Acknowledgements

This work was partially supported by the National Library of Medicine, grant LM-07089-08.

References

[1] Department of Health and Human Services, Proposed Security and Electronic Signature Standards. Federal Register: August 12, 1998 (Volume 63, Number 155): 43241-43280.

[2] Chapman D., Zwicky E. "Firewall Design"; 1996. Available at: URL: http://www.sunworld.com/swol-01-1996/swol-01-firewall.html. Used by permission.

[3] Hare C., Siyan K. Internet Firewalls and Network Security. New Riders Publishing; 1996.