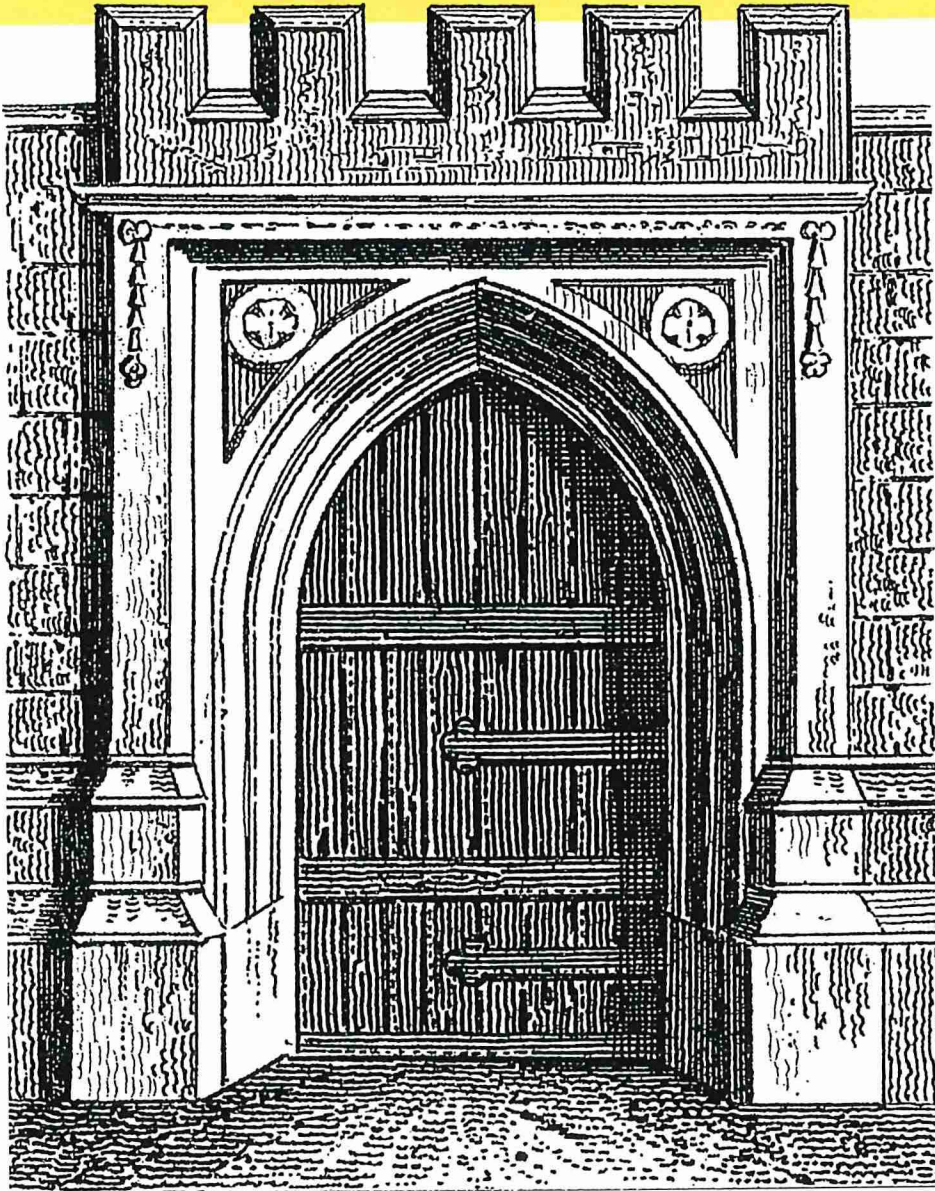


Building Internet

Firewalls



O'REILLY®

*Elizabeth D. Zwicky, Simon Cooper
& D. Brent Chapman*

FreeWheel, Ex. 1087
FreeWheel v. AlmondNet
IPR2024-00419

Building Internet Firewalls

Building Internet Firewalls

Second Edition

Elizabeth D. Zwicky, Simon Cooper &
D. Brent Chapman

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Paris · Sebastopol · Taipei · Tokyo

Building Internet Firewalls, Second Edition

by Elizabeth D. Zwicky, Simon Cooper, and D. Brent Chapman

Copyright © 2000 O'Reilly & Associates, Inc. All rights reserved.
Printed in the United States of America.

Published by O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472.

Editor: Deborah Russell

Production Editor: Nancy Crompton

Production Coordinator: Madeleine Newell

Cover Designer: Edie Freedman

Printing History:

April 1995:	First Edition.
November 1995:	Minor corrections.
June 2000:	Second Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly & Associates, Inc. Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. O'Reilly & Associates, Inc. is independent of Sun Microsystems.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly & Associates, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps. The association between the image of a Gothic doorway and the topic of Internet firewalls is a trademark of O'Reilly & Associates, Inc.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Library of Congress Cataloging-in-Publication Data

Zwicky, Elizabeth D.

Building Internet firewalls / Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman.--

2nd ed. p. cm.

ISBN 1-56592-871-7

1. Computer networks--Security measures. 2. Internet (Computer network)--Security measures. I. Cooper, Simon. II. Chapman, D. Brent. III. Title.

TK5105.59.Z85 2000

005.8--dc21

00-039954

ISBN: 1-56592-871-7

[12/00]

[M]

nal users can use it as a platform to attack internal servers they would not otherwise be able to get to (this is particularly dangerous if they can use CONNECT to get to arbitrary services). Even if the proxy server can't be used this way, it can be used to attack third parties.

People often search actively for open HTTP proxy servers. Some of these people are hostile and want to use the proxy servers as attack platforms, but some of them just want to use the proxy servers to access web sites that would otherwise be unavailable to them because of filtering rules at their site (or in a few cases, filtering imposed by national governments). Either way, it's probably not to your advantage to let them use your site. Being nice to people behind restrictive filters is tempting, but in the long run, it will merely use up your bandwidth and get you added to the list of filtered sites.

Network Address Translation Characteristics of HTTP

HTTP does not use embedded IP addresses as a functional part of the protocol, so network address translation will not interfere with HTTP. Web pages may contain URLs written with IP addresses instead of hostnames, and those embedded IP addresses will not be translated. You should therefore be careful about the content of web pages on servers behind network address translators.

In addition, HTTP clients may provide name and/or IP address information to servers, leaking information about your internal numbering and naming schemes. HTTP clients may provide "From:" headers, telling the server the user's email address (as the user told it to the browser), and proxies may add "Via:" headers indicating the IP addresses of proxies that a request (or response) has passed through.

Securing HTTP

You may hear discussions about secure versions of HTTP and wonder how they relate to firewalls and the configuring of services. Such discussions are mainly focused on the privacy issues of passing information around via HTTP. They don't really help solve the kinds of problems we've been discussing in previous sections.

Two defined protocols actually provide privacy using encryption and strong authentication for HTTP. The one that everyone knows is usually called HTTPS and is denoted by using *https* in the URL. The other, almost unknown protocol, is called Secure HTTP and is denoted by using *shttp* in the URL.

The goal of HTTPS is to protect your communication channel when retrieving or sending data. HTTPS currently uses TLS and SSL to achieve this. Chapter 14, *Intermediary Protocols*, contains more technical information on TLS and SSL.

The goal of Secure HTTP is to protect individual objects rather than the communications channel. This allows, for example, individual pages on a web server to be digitally signed—a web client can check the signature when the page is downloaded. If someone replaces the page without re-signing, then the signature check will fail, causing an alert to be displayed. Similarly, a secure form that is submitted to a web server can be a self-contained digitally signed object. This means that the object can be stored and used later to prove or dispute the transaction.

The use of Secure HTTP could have significant advantages for the consumer in the world of electronic commerce. If a company claims that it has a digitally signed object indicating your desire to purchase 2,000 rubber chickens but the digital signature doesn't match, then you can argue that you did not make the request. If the signature does match, then it can only mean one of two things; either you requested the chickens, or your private key has been stolen. In contrast, when you use HTTPS, your identity is not bound to the transaction but to the communication channel. This means that HTTPS cannot protect you from someone switching your order for rubber chickens to live ones, once it has been made, or just ordering chickens on your behalf.

Packet filtering characteristics of HTTPS and Secure HTTP

HTTPS uses a single TCP connection at port 443. Secure HTTP is designed to operate over port 80 (see the section on HTTP).

Direction	Source Addr.	Dest. Addr.	Protocol	Source Port	Dest. Port	ACK Set	Notes
In	Ext	Int	TCP	>1023	443	^a	Request, external client to internal server
Out	Int	Ext	TCP	443	>1023	Yes	Response, internal server to external client
Out	Int	Ext	TCP	>1023	443	^a	Request, internal client to external server
In	Ext	Int	TCP	443	>1023	Yes	Response, external server to internal client

^a ACK is not set on the first packet of this type (establishing connection) but will be set on the rest.

Proxying characteristics of HTTPS and Secure HTTP

Because HTTPS and Secure HTTP use straightforward TCP connections, they are quite easy to proxy. Most programs that provide HTTP proxying also provide HTTPS and Secure HTTP proxying. However, both HTTPS and Secure HTTP use end-to-end encryption between the client and the server. This means that the data