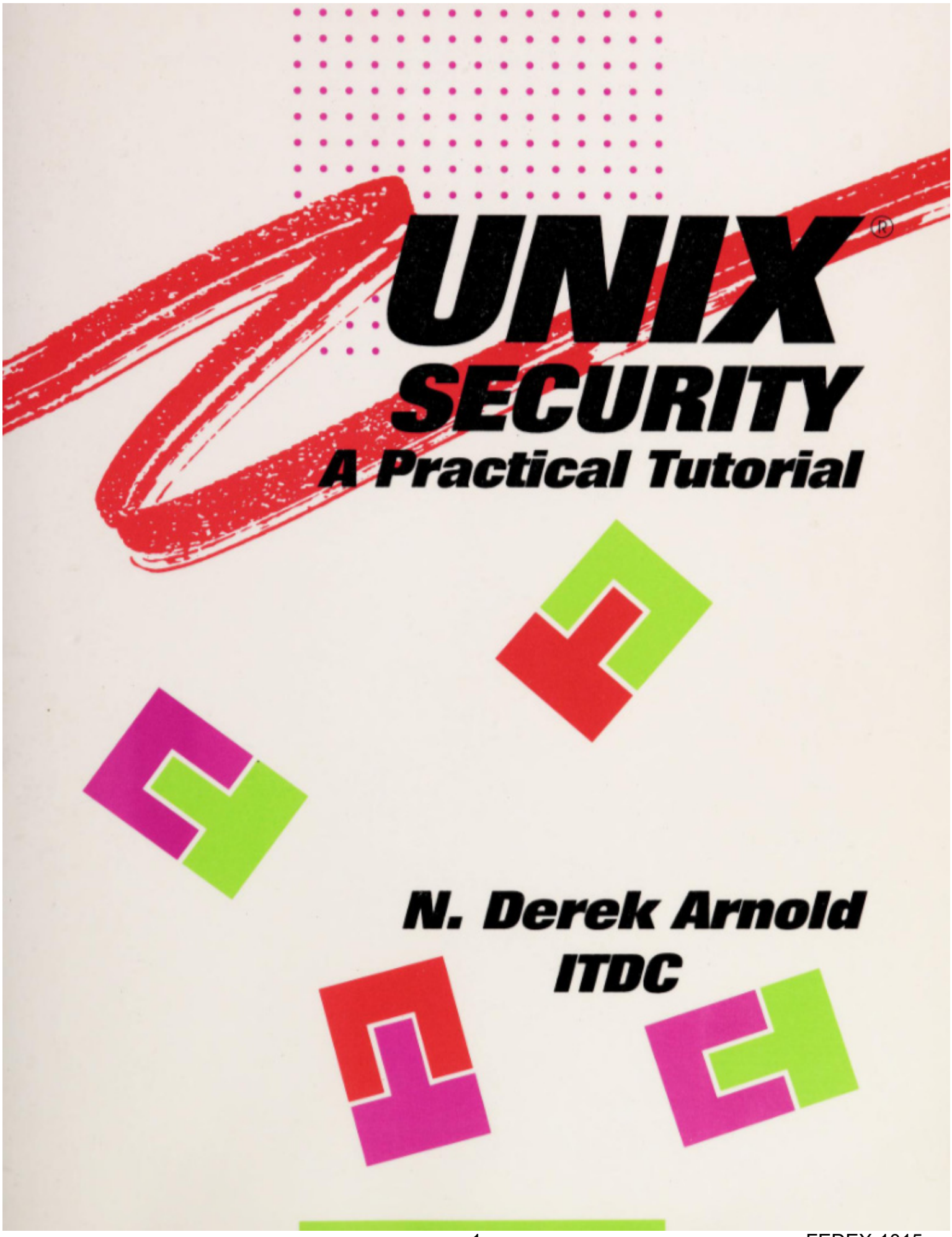

FEDEX-1015

UNIX Security

A Practical Tutorial

N. Derek Arnold
Vice President of Technical Services, ITDC
Cincinnati, Ohio

McGraw-Hill, Inc.

New York San Francisco Washington, D.C. Auckland
Bogotá Caracas Lisbon London Madrid
Mexico City Milan Montreal New Delhi San Juan
Singapore Sydney Tokyo Toronto

Other McGraw-Hill Titles of Interest

BAKER • *C Mathematical Function Handbook* 0-07-911158-0

BAKER • *C Tools for Scientists and Engineers* 0-07-003355-2

BAKER • *More C Tools for Scientists and Engineers* 0-07-003358-7

GROTTOLA • *The UNIX Audit* 0-07-025127-4

HANCOCK ET AL. • *The C Primer* 0-07-026001-X

HOLUB • *C+ C++ : Programming with Objects in C and C++* 0-07-029662-6

MCGILTON, MORGAN • *Introducing the UNIX System* 0-07-045001-3

MORGAN, MCGILTON • *Introducing UNIX System V* 0-07-043152-3

PARRETTE • *UNIX for Application Developers* 0-07-031697-X

PARRETTE • *Motif Programming in the X Window System Environment* 0-07-031723-2

RANADE, ZAMIR • *C++ Primer for C Programmers* 0-07-051216-7

SMITH • *Concepts of Object-Oriented Programming* 0-07-059177-6

SMITH • *C++ Applications Guide* 0-07-039010-X

SMITH • *C++ for Scientists and Engineers* 0-07-059180-6

TARE • *Data Processing in UNIX* 0-07-062885-8

TARE • *UNIX Utilities* 0-07-062879-3 (hard cover); 0-07-062884-X (soft cover)

To order or to receive additional information on these or any other McGraw-Hill titles in the United States, please call 1-800-822-8158. In other countries, please contact your local McGraw-Hill representative.

MH92

Library of Congress Cataloging-in-Publication Data

Arnold, N. Derek.
UNIX security : a practical tutorial / N. Derek Arnold.
p. cm. — (UNIX/C series)
Includes index.
ISBN 0-07-002559-2 — ISBN 0-07-002560-6 (pbk.)
1. Operating systems (Computers). 2. UNIX (Computer file).
3. Computer security. I. Title. II. Series.
QA76.76.063A758 1992
005.8—dc20 92-34992
CIP

Copyright © 1993 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

2 3 4 5 6 7 8 9 0 DOH/DOH 9 8 7 6 5 4 3

ISBN 0-07-002559-2 {HC}
ISBN 0-07-002560-6 {PBK}

The sponsoring editor for this book was Neil Levine, the editing supervisor was Kimberly A. Goff, and the production supervisor was Suzanne W. Babeuf.

Printed and bound by R. R. Donnelley & Sons Company.

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY
The author and publisher have exercised care in preparing this book and the programs contained in it. They make no representation, however, that the programs are error-free or suitable for every application to which the reader may attempt to apply them. The author and publisher make no warranty of any kind, expressed or implied, including the warranties of merchantability or fitness for a particular purpose, with regard to these programs or the documentation or theory contained in this book, all of which are provided "as is." The author and publisher shall not be liable for damages in amount greater than the purchase price of this book, or in any event for incidental or consequential damages in connection with, or arising out of the furnishing, performance, or use of these programs or the associated descriptions or discussions.

Readers should test any program on their own systems and compare results with those presented in this book. They should then construct their own test programs to verify that they fully understand the requisite calling conventions and data formats for each of the programs. Then they should test the specific application thoroughly.

Secure UNIX Variations

- Major computer security violations publicized by the mass media caused the Department of Defense (DoD) to create a set of guidelines and categories to evaluate computer systems for security. The guidelines divide the systems into categories. Each category is defined by a letter and a number. Each successive letter indicates a less trusted system. An **A** system is *more trusted* than a **B** system. To give levels within these categories, the DoD also assigns numerical subdivisions, with 1 being less secure than 2. Each level builds on the previous level by adding more security restrictions to the former level.

  These criteria were published by the DoD as "DOD 5200.28-STD", *"Department of Defense Trusted Computer System Evaluation Criteria."* It was given an orange cover when it was first published in 1985, so that it would not get lost on users' shelves. Today it is usually known simply as ***the orange book*** and has been joined by several other DoD publications on computer security, each with a different color cover. They are known collectively as ***the rainbow books*** and can be ordered from the Government Publications Office.

  Trusted computer base (TCB) rating is based on both hardware and software. A system must meet *all* of the criteria before being certified.

Division D

- The orange book's ***Division D: Minimal Protection*** only has one class. To quote the orange book:

  > *It is reserved for those systems that have been evaluated but that fail to meet the requirements for a higher evaluation class.*

  Most home computers fall into this category. Manufacturers do not submit their products for this classification. Rating by the National Computer Security Center is a lengthy (and therefore expensive) process. It is not worth it to apply for a Division D rating, which is implied by default.

Division C

- ***Discretionary Protection***

  > *Classes in this division provide for discretionary (need-to-know) protection and, through the inclusion of audit capabilities, for accountability of subjects and the actions they initiate.*

Division B

- ***Mandatory Protection***

  The notion of a TCB that preserves the integrity of sensitivity labels and uses them to enforce a set of mandatory access control rules is a major requirement in this division. Systems in this division must carry the sensitivity labels with major data structures in the system. The system developer also provides the security policy model on which the TCB is based and furnishes a specification of the TCB. Evidence must be provided to demonstrate that the reference monitor concept has been implemented.

Division A

- The orange book's ***Division A*** includes all of the standards for ratings C through B, but adds even more qualifications. Quoting the orange book again:

  This division is characterized by the use of formal security verification methods to assure that the mandatory and discretionary security controls employed in the system can effectively protect classified or other sensitive information stored or processed by the system. Extensive documentation is required to demonstrate that the TCB meets the security requirements in all aspects of design, development and implementation.

To date, only one computer has been granted an A1 security classification, the Honeywell SCOMP system. There are one or two being tested.

- Government purchases are the largest market for computers. The government requires highly trusted systems. The UNIX operating system has been augmented by many firms in order to gain better ratings. Some firms that are marketing these products are:

  - Sun Federal, Inc., a wholly owned subsidiary of Sun Microsystems, Inc., which markets a variation called Multi-level Security (MLS) which AT&T has incorporated into its government release of the UNIX operating system. This variation is being evaluated for a **B1** rating.
  - Addamax Corporation of Champaign, Illinois, is also being evaluated for a **B1** rating. They are producing a secured version for a foreign market.
  - SecureWare, Inc. of Atlanta, Georgia, has submitted systems for rating evaluations for **B1, B2** and **C2** ratings.

  AT&T (which means Sun Federal, Inc. also) has submitted UNIX System V/MLS for evaluation. AT&T is making the MLS variation more accessible to the general public. It was formerly restricted to government agencies.

UNIX in the Near Future

- UNIX developers have already announced their intentions of releasing a version of the operating system that would not have a Super User. It will probably have a hierarchical group authority rather than putting all of the Super User powers in one basket. Their powers will still be powerful, "abusable", and mis-usable.
- UNIX V release 4.3 has the shadowed password file, as does the AIX system, along with others. Soon most UNIX Systems will have the password file "hidden", unlike some of today's versions which allow the world to see it.
- Companies are marketing new security software and hardware every day. They come in all prices and capabilities. The following chapters have been designed to help you know your system and design your security. Study your needs carefully and design your security with sensibility and care. Then enforce it!

Review Questions

- Please write down the answers to the following questions.

1. Where was the UNIX operating system developed?
2. Which university was responsible for many user friendly enhancements to the UNIX operating system?
3. What is the name of the variation this university markets?
4. Why did AT&T suddenly start marketing the UNIX operating system?
5. What was the computing industry's response to AT&T marketing?
6. What was AT&T's response to the industry?
7. What is the major motivating force today behind the UNIX operating system?
8. List at least two variations on the UNIX operating system that are available in secure form.
9. Is the MLS UNIX operating system available to the general public?
10. Is the MLS UNIX operating system rated at a B1 level?

end users of the product do not need to have any permissions on the database directory except execute. Since the product calculates the name of user created tables from its fixed name catalog, the permissions on a database directory may be simply 110 with execute for the DBA (creator) and informix group.

- By setting the DBTEMP variable, the user can hold confidential data in a directory to which only they have access. This means that the race condition of creating database temporary files can be defeated. Using a different directory is not without its cost. Imagine administrating a temporary directory for each database user.

  This can be done by using shell scripts and a naming convention. As an example, the DBTEMP variable could be set to a directory called *tmp* in the user's home directory. A shell script could parse the password file for unique home directories of non-administrative identities. The list created could form the basis of a cleanup *find* command.

Encryption of the Data

- All the techniques for security on the data is useless against the Super User, who can read any file and use any UNIX utility. The only way to completely secure a database from the Super User is not to have one. However, it is possible to frustrate a Super User. The use of C-hooks in the INFORMIX (and other products) can allow very confidential data to be encrypted. Without the proper key, the data is useless. This book cannot be an exhaustive authority on encryptions. The reader should get as much help on this subject as possible from authorities on the subject if the encryption of the database is a mandate. The remainder of this chapter is a simple introduction to the techniques used in encryption.

XOR function

- By using C-hooks in a PERFORM screen or an ACE report, it is possible to encrypt and decrypt data. One of the simplest forms of encryption is the eXclusive OR (XOR) function of the C language. The encryption of data may be done as a reversible algorithm. Example 10.1 is a small program that will encrypt a file using an XOR function.

  The XOR function is limited by its key. The larger the key generated, the better the encryption. The more random the key the better. This is the basis for a very secure method of encryption called scratch pad encryption. In ***scratch pad encryption*** the sender and receiver of data have a set of text which is the key. This text is random and known only to the sender and the receiver. The scratch pad is used to encrypt one and only one message. Since it is a large random key which is changed with each message the probability of breaking the message is nil. In our Example 10.1 the key was a string passed in as the first parameter. There is no reason that the key could not have been a file name. This file would be used once and then thrown away.

```
# include <stdio.h>

int main( argc, argv)
        char *argv[];
{
        int     len;
        FILE    *in;
        FILE    *out;
        int     x;
        int     i = 0;

        /* check usage */
        if( argc !=3 ) {
            fprintf( stderr, "USAGE: %s key file\n", *argv );
            exit( 1 );
        }

        /* get length of key */
        len = strlen( argv[1] );

        /* open file */
        if( ( in = fopen( argv[2], "r" ) ) == NULL ) {
            fprintf( stderr, "Could not open %s\n", argv[2] );
            perror( *argv );
            exit( 2 );
        }

        /* open output file */
        if( ( out = fopen( "output", "w" ) ) == NULL ) {
            fprintf( stderr, "Could not open output\n", argv[2] );
            perror( *argv );
            exit( 2 );
        }

        while( (x = fgetc( in ) ) != EOF ) {
            putc( x ^ argv[1][i], out );
            i++;
            i %= len;
        }

        /* close files */
        fclose( in );
        fclose( out );

        /* exit with success */
        exit( 0 );
}
```

Example 10.1 Exclusive or encryption.

Two Pad Encryption

- Two Pad Encryption is a slightly more sophisticated way of doing the XOR encryption using a simpler set of keys. It has the advantage of generating a larger key from a smaller set of keys. Example 10.2 is a two pad encryption. The length of pad1 and pad2 have been carefully constructed at 17 and 11 characters respectively. Since these are prime numbers, the characters are guaranteed to rotate so it will take 11 uses of pad1 and 17 uses of pad2 to repeat a XOR combination. This means the 28 characters in the two pads create a combination of 17 × 11, or 187, scratch pad values. If the values of the pads are chosen with random values that are guaranteed to be non-repeating, relatively small pads can be used to create large combination pads. Of course additional pads could be added to create even larger combinations. The program *scrambler.c* in this volume's Reference Section, with its companion PERFORM screen called test_s and C-hook called test_f is an example of a encryption algorithm based on the two pad method.

The encrypt Library Function

- Along with these techniques there exists an implementation of the DES (Data Encryption Standard) specified by the Federal Government as a part of the Standard C library of UNIX. These functions are well documented and therefore we will not cover them here. These DES functions may be included in a DBMS user program as easily as any of the other functions we have cited.

  There is a well known problem in using these functions. If the text material is large enough, the data may be decrypted using functions available in the public domain. A Cryptographers Workbench exists and is quite good at breaking the UNIX DES algorithm. The Workbench is of dubious use if a column or columns of a database is encrypted using a password. Since the text encrypted is small, it is also harder to break, especially if the password used on different rows is different. This does not mean that the data cannot be decoded. It just means that it would be more difficult to extract the data and decode it.

Encryption and Password Problems

- There are a couple of problems with encrypting data. In password protected methods the password must be remembered for as long as the data is needed. Password distribution becomes a problem. The larger the number of people that know passwords the more likely the password is to be revealed.