

# TOPICS IN COMPUTER SECURITY

*for the undergraduate student*

Jim Griffin

*Cabrillo College*

**Abstract:** Cabrillo College has just adopted a new program in Computer Network and System Administration for the purpose of preparing students for industry certification in fields relating to Information Technology and for admission into higher education at four-year institutions. This report describes how Information Assurance topics, exercises and labs were integrated into the existing courses of this program. Specifically, examples in the five areas of *passwords*, *encryption*, *viruses*, *spoofing*, and *steganography* are given.

**Key words:** security, passwords, virus, encryption, spoofing, steganography

## 1. INTRODUCTION

With the goal of integrating Information Assurance concepts into existing Computer and Information Systems classes, four target classes were identified:

- An Introduction to UNIX/Linux
- UNIX/Linux Shell Programming
- Perl Programming in UNIX
- UNIX/Linux System Administration

Information Assurance topics from the areas of Authentication, Encryption and Threats were chosen to incorporate into these classes. The Introductory course was ideal for inclusion of a password security exercise, since students obtain accounts on campus computers in this class. The two programming classes were appropriate for exercises in handling viruses, spoofing, and steganography. The MD5 Message-Digest Algorithm allowed a way to bring an encryption topic into the account administration portion of the administration course. The exercises and labs developed for these three topics in Information Assurance will be outlined and described in the following sections.

### AUTHENTICATION:

This exercise in password security addresses one of the student learner outcomes for the Introduction to UNIX/Linux class:

*Access a UNIX computer in a secure manner*

The exercise described below acts as both a learning activity in password security and authentication as well as a means of assessing the success of the outcome. The password cracking tool can be used to determine if students are selecting adequately secure passwords.

Students in this class are given accounts to a remote UNIX server for their class work. The accounts are initially created in a locked state with their passwords being the same as their login names. The accounts are activated on the first day, and the students are instructed on how to log in. When all students are logged in, the instructor runs the **John the Ripper**<sup>1</sup> program, demonstrating how quickly the passwords are *cracked*. Students are then instructed on how to change their password, and after doing so, the cracking program is run again. This serves as a discussion for the concept of good vs. bad passwords. The WECS5 material provided below and discussed in the **Lab 1** for CS3600 is used to instruct students how to select secure passwords that are memorable.

Table 1. The Brute Force Attack

Alphabet Size / Password Length	26	52	93
4	$4.57 \times 10^5$	$7.31 \times 10^6$	$7.48 \times 10^7$
6	$3.09 \times 10^8$	$1.98 \times 10^{10}$	$6.49 \times 10^{11}$
7	$8.03 \times 10^9$	$1.02 \times 10^{12}$	$6.01 \times 10^{13}$
8	$2.09 \times 10^{11}$	$5.34 \times 10^{13}$	$5.60 \times 10^{15}$
10	$1.41 \times 10^{14}$	$1.44 \times 10^{17}$	$4.84 \times 10^{19}$

## THREATS

Threats to computer security come in all shapes and sizes, but the two programming classes using the UNIX shell language and Perl lend themselves to handling a variety of software threats.

### Viruses

One of the student learner outcomes for the course, UNIX/Linux Shell Programming is:

*Format the output and process the input for a shell script using the UNIX commands of sed and awk*

The lab exercise described below serves as an excellent way to give the students practice in performing string manipulation and file processing within a shell script. At the same time, the student is discovering the principles of virus replication, detection and eradication.

I provided a shell-scripted virus that, when run, would infect all shell scripts in the current directory and subdirectories one level below the current directory. The text of the script is included below. The spreading nature of the virus was demonstrated to all students in class, and it was pointed out that the virus caused no damage other than replicating itself.

Students were then assigned the task of writing a scanner shell script that would identify any infected files, and, if invoked with a specified option, would fix the script by removing the virulent code. Issues that came up in this lab were ones relating to the importance of correctly identifying and isolating the undesirable code. We discuss false positive and false negative identifications.

```
-----
#!/bin/bash
VIRUS=`sed -n "2,18p" < $0`
FILES=`find . -maxdepth 2 2>/dev/null`
for i in $FILES
do
```

```

set `file $i` > /dev/null 2>&1
if [ "$3" = "shell" ]      # Infect only shell scripts
then FILE=`echo $1 | sed "s/:$//"`
  HDR=`sed -n "2,18p" < $FILE`
  if [ "$HDR" != "$VIRUS" ] # Don't infect twice!
  then
    ORIG=`tail +2 $FILE` # Get all lines except first
    echo "#!/bin/bash" > $FILE
    echo "$VIRUS" >> $FILE
    echo "$ORIG" >> $FILE
  fi
fi
done
echo Achoooo, I've got a cold.
exit 0

```

---

## 1.1 Spoofing

Another student learner outcome for the UNIX/Linux Shell Programming course is:

*Write shell scripts that interact with the user as well as read and write files on the system.*

The lab exercise described below gives students the opportunity to practice reading the keyboard and writing to the terminal screen in a variety of ways. At the same time the student learns how login spoofing programs can be used to capture a user's password. Not only is the student challenged in trying to reproduce the activity of a user login as accurately as possible, but practices in detecting spoofing attempts are learned.

### Login Spoof Program

---

One way that system security is compromised is by one program pretending to be, (spoofing), another program. Spoofing the login program is one way to catch a user's password. System Administrators need to be aware of these types of programs and have procedures for mitigating against this kind of attack. To see what is involved in this kind of spoof, your task is to write a shell script program that mimicks a user logon session from the initial logon message through to the display of the shell prompt in the user's home directory. You will mail the captured password to yourself, and otherwise try not to let the user know the login was faked.

The purpose of this exercise is not to capture unsuspecting users' passwords, but to serve as a discussion as to what it takes to successfully spoof a login and how you as a system administrator can guard against this type of attack.

---

## 1.2 Steganography

A Perl Programming in UNIX course includes in its content: arrays, data transformation and the language's use within the World Wide Web. The instructor for this course incorporated as a final lab project, a lab on steganography. An image was hidden in the low-order bytes of a bitmapped file, (bmp image). The bmp image was used so students didn't also have to deal with compression and complex structures. A 24-bit color model was chosen so that students could process the input byte at a time. The low-order bits of pixels that did not carry hidden information were zeroed out.



Although we have not done anything specifically to accommodate underrepresented groups in Computer Science, we have a separate program titled, *The Watsonville Digital Bridge Academy*<sup>3</sup>, which is helping underrepresented groups to make a successful transition to college through such themes as support services, learning-to-learn, time management and self confidence.

The success of integrating IA concepts into our existing courses was manifested by an increase in student interest in performing the labs and exercises that related to real-world and sometimes “taboo” subjects. The relevancy of their work instilled a greater desire to succeed and was therefore motivating. In choosing a lab exercise for the students, it is important that the difficulty of the assignment match the objectives of the course as well as the capability of the students.

## NOTES

1. John the Ripper is a password cracker, currently available for UNIX, DOS, WinNT/Win95. Its primary purpose is to detect weak UNIX passwords. It has been tested with Linux x86/Alpha/SPARC, FreeBSD x86, OpenBSD x86, Solaris2.x SPARC and x86, Digital UNIX, AIX, HP-UX, and IRIX. <http://www.openwall.com/john/>
2. Network Working Group, R. Rivest, Request for Comments: 1321  
MIT Laboratory for Computer Science and RSA Data Security, Inc. April 1992
3. Watsonville Digital Bridge Academy: <http://www.cabrillo.cc.ca.us/~wdba>