

CSAW '08 Embedded Systems Challenge

Team "Tropicana" Submission

Sean Cassidy, Robert Ghilduta, Daniel Liu, Jon Szymaniak
Rochester Institute of Technology

I. INTRODUCTION

MULTIPLE trojans have been implemented to address each of the three general categories of attacks, which include functional specification modifications, information leakage and denial of service. Section II first provides the team's analysis of the encryption device and possible attack vectors. Section III details a trojan which transmits unencrypted message text in real-time as it is typed. The second trojan provides an adversary the ability to obtain the master encryption key, and is described in Section IV. Section V details a Denial Of Service (DoS) attack that disables the ability to transmit encrypted text over the RS232 port. Section VI addresses the last trojan, a method of strobing key presses as they occur over LEDs.

II. ANALYSIS AND ATTACK VECTORS

In designing trojans for the *Alpha* encryption device, contest restrictions, in addition to the standard device operation, were both taken into account. Although some attacks theorized by the team may have been less trivial to detect, they were not implemented if difficult to verify without additional hardware (per contest rules), or unlikely to be carried out successfully by an adversary in the setting in which the device was intended to be used.

The *Alpha* device requires that an attacker have physical access to the device and either a PC, or a device capable of entering keystrokes via the PS/2 port, and retrieving data over the serial port. Therefore, it is desirable that the trojans provide an attacker the means to perform actions quickly, without arousing suspicion through the use of extraneous hardware. As a result, the trojans implemented and presented in this document utilize serial communications to leak information, and are enabled via keyboard input.

The aforementioned restrictions eliminated some of the trojans theorized by the team. For example, simple observation of the device's functionality suggests that the red and blue pins of the RGB connector were left unused. This provides the opportunity to leak sensitive information over one of these unused pins. However, this would require that additional hardware be constructed to monitor these pins. Additionally, this attack would include the challenge of leaking information while not interfering with the information displayed on the monitor. Carrying out this attack would be highly valuable if an attacker were able to place a malicious

monitor, which would include the hardware necessary to read and store information leaked on the unused pins of the RGB connector, in the location where the *Alpha* device was being operated. Security-conscious users may suspect the use key-loggers connected to a keyboard, but would be far less likely to suspect that a monitor would be logging information. Due to time restrictions, this attack was not further investigated.

Since standard operation of the device requires that information transmitted over the RS232 connection be validated during testing, the trojans described in sections III and IV were designed to output sensitive information via a second "hidden" serial port. As a proof of concept, this port is connected to the JD 6-pin header on the Basys board, as shown in Figure 1. However, this "hidden" serial transmitter could be connected to other unused pins to avoid detection during testing.

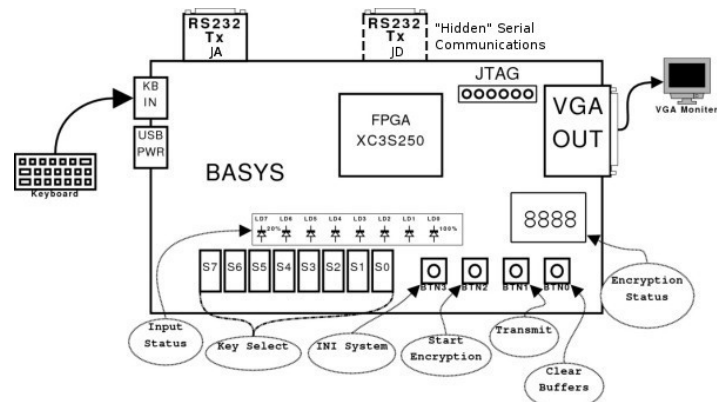


Figure 1: Alpha Setup With "Hidden" Transmitter

Upon reviewing the code for the *Alpha* project, the team found that the master key was hard-coded in `alphatop.v`. Furthermore, it was noted that the eight "key" bits set by the user prior to performing encryption are included in the encrypted message. The packet format includes the 8-bit modifier for the 128-bit master key, enabling the provided `enc_verifier` program to automatically decrypt any message with a matching master key. That is, any 8-bit value specified does not need to be given to the `enc_verifier` program; it is given already. Therefore, by obtaining the master key, an adversary could easily decrypt any intercepted messages. This observation has motivated the development of the trojan described in Section IV.

Although the *Alpha* device provides a means to encrypt sensitive data prior to an exchange, the level of security the device provides can only be obtained through correct

VI. STROBING ASCII VALUES

To allow the attacker to know what is being typed into Alpha remotely a different approach had to be taken to implementing a transmission mechanism. To achieve this, a series of five LEDs were used to represent ASCII letter values. Five bits is less than what is required to transmit ASCII values in full, however due to implementation constraints and making it so that the backdoor is not as obvious letters are capitalized (to save one bit) and numeric values are omitted, which brings the character set to a total of 26. To output the five bit vector the four dots on the four-seven-segment display are used in conjunction with the middle bar in the right most seven segment display. The middle bar was chosen in order to avoid easy detection by making it seem that the number changing from zero to eight is part of the operation of *Alpha*. Values are strobed to the output mechanism as keys get pressed. The time that the LEDs remain on is relatively short for the human eye to pick up but enough for a fast camera to detect. If the “under handing” operator is willing to risk detection he may enable the strobed values to last all the way until a new value is pressed and received from the keyboard. The output is simple to understand being that it’s just a character’s ASCII value without the leading three bits from its eight bit (or seven regarding ANSI) being displayed. The values being strobed to the dots on the seven segment displays represent the four MSBs of the five bit value and the bar going across the middle of the last seven segment display represents the LSB.

VII. CONCLUSION

As described above, separate trojans have been implemented to cover a range of possible attacks. In order to attempt to maintain verifiability, in addition to reducing their footprint, these trojans have been designed to be fairly simple. These trojans are included in separate builds, with the folder/zip file name denoting the contents of the build. Based upon these trojans, far more elaborate and elusive attacks could be derived, given that an attacker has sufficient time to minimize the footprint of the trojans.

Appendix A - Modified enc_verifier program used to obtain leaked master key

```

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>

#define BAUDRATE B9600
#define MODEMDEVICE "/dev/ttyS0"
#define _POSIX_SOURCE 1 /* POSIX compliant source */
#define FALSE 0
#define TRUE 1

volatile int STOP=FALSE;

int main()
{
    int fd,c,rx,i;
    struct termios oldtio,newtio;

    fd = open(MODEMDEVICE, O_RDONLY | O_NOCTTY );
    if (fd <0) {perror(MODEMDEVICE); exit(-1); }

    tcgetattr(fd,&oldtio); /* save current port settings */

    bzero(&newtio,sizeof(newtio));

    newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
    newtio.c_iflag = IGNPAR;
    newtio.c_oflag = 0;

    /* set input mode (non-canonical, no echo,...) */
    newtio.c_lflag = 0;

    newtio.c_cc[VTIME]      = 0; /* inter-character timer unused */
    newtio.c_cc[VMIN]      = 1; /* blocking read until 5 chars received */

    tcflush(fd, TCIFLUSH);
    tcsetattr(fd,TCSANOW,&newtio);

    printf("Waiting for transmission to begin.....\n");
    while (1) {/* loop for input */
        unsigned char buf[8];
        int i=0;
        int e=0;
        rx = read(fd,buf,8);
        if(rx<0) {printf("Error %d", rx);exit(-1);}
        fflush(stdout);
        for(i=rx;i > 0;i--) {
            printf("%02x",buf[rx - i]);
            fflush(stdout);
        }
        bzero(buf,sizeof(buf));
    }
}

```

```
tcsetattr(fd, TCSANOW, &oldtio);  
close(fd);  
}
```