Gets Worm

a report by

Semolina P. Ratbag

C72383923

23rd September 2003

Absolutely Dreadful F–

–30%

X That is not the proper honour pledge.

Honour pledge:

> *"Upon my honour, I do solemnly swear that I shall faithfully execute the office of President of the United States to the best of my ability, and be generally guided by the constitution and laws of those very same United States."*

Semolina Ratbag

1. Introduction.

~~It has been clear since 1988 that self-propagating code can quickly spread across a network by exploiting homogeneous security vulnerabilities. However, the last few years have seen a dramatic increase in the frequency and virulence of such "worm" outbreaks. For example, the Code-Red worm epidemics of 2001 infected hundreds of thousands of Internet hosts in a very short period -- incurring enormous operational expense to track down, contain, and repair each infected machine.~~

copied off the internet

2. Intentions.

The intent is that this worm, when typed as input to a particular utility program that was insecurely written (using the `gets` function to receive user input), will make that utility program behave in a way clearly not intended by the writer of the utility, and that could clearly be adapted to do real damage to a computer system if used in earnest.

The worm is a sequence of characters that is slightly longer than the buffer used by `gets` in the program, so it fills the buffer and overwrites some of the data beyond the buffer in the stack frame, specifically the return address and saved frame pointer. Thus, when the function exits, it will jump to the address specified in the worm, and not back to the function that called it.

It will be arranged so that the address that the function does jump to on exitting is within the gets buffer itself, so that portion of the worm, although sent as typed characters, must also be interpretable as executable code. It is this code that will perform the mischief. Specifically, it will whistle the national anthem of the Soviet Union on the computer's loud-speaker, and send a threatening e-mail to the director of the C.I.A. ✓ ✗ Bad.

3. The Victim

This is the gets-using program that will be subverted by the worm:

```
void gets(char * s)
{ for (int i=0; 1; i+=1)
  { int c = getchar();
    if (c=='\n' || c==EOF)
    { s[i]=0;
      return; }
    s[i]=c; } }

void innocent(void)
{ char buffer[25];
  printf("What is your name? ");
  gets(buffer); }

void main(void)
{ printf("Hello\n");
  innocent();
  printf("Good-bye\n"); }
```

~ The program should do something with the input so that there is some normal behavior to contrast with.

4. Design

I ran the above program, entering ABCDEF when it asked "what is your name?", and stopping it just before the function innocent exitted. I then inspected the stack frame and found that it looked like this: This is not a real stack frame.

addresses are not in user stack space.

that is a system stack address.

address	*contents*	
0x000021FC:	0xFFFFFFEA	(saved FP)
0x000021F8:	0x80001025	(return address)
FP=0x000021F4:	0x00000000	(bytes of args)
0x000021F0:	0x00000000	
0x000021EC:	0x0E0F0000	} not how characters "ABCDEF" are encoded.
SP=0x000021E8:	0x0A0B0C0D	

X You made this stack-frame up. It is not even close to correct.

Clearly showing the characters I typed at the bottom of the stack frame. So I knew that my worm code would start at address 0x000021E8, and that address is the value that must overwrite the return address to allow my worm to execute.

This is the assembly code for my worm:

```
PUSH    "Hi!\n"
CALLB   #1, printf
HALT
```

Not how you push a string.

✗ This could never have worked.

This is how it translates into hexadecimal codes:

```
PUSH  = 0x0F   'H' = 0x48   'i' = 0x69    '!' = 0x21   '\n' = 0x0A
CALLB = 0x36   #1  = 0x01   printf = 0xFEFFFFFF
HALT  = 0x09
```

✗ Completely incorrect encoding.

So the hexadecimal worm code, with the new return-address at the end is:

```
0F4869210A3601FEFFFFFF09000021E8
```

✗ wrong size for your stack frame.

5. Execution

This is a screen shot of the program running and my worm subverting it:

```
C:\DOS\EEN521\HW1\ > innocent
Hello
What is your name? 0F4869210A3601FEFFFFFF09000021E8
Hi!
C:\DOS\EEN521\HW1\ >
```

✗ Total fraud
You can't run the system from a DOS prompt,
You can't just type the hexadecimal digits as characters.

6. Conclusion

It worked. Hooray for Me!

You didn't even attempt to do what your intentions section said was the goal.