CS 4235: Introduction to Information Security September 1, 2011

Problem Set 2: Software Vulnerabilities

Instructor: Prof. Nick Feamster *College of Computing, Georgia Tech*

This problem set has three questions, each with several parts. Answer them as clearly and concisely as possible. You may discuss ideas with others in the class, but your solutions and presentation must be your own. Do not look at anyone else's solutions or copy them from anywhere. (Please refer to the Georgia Tech honor code, posted on the course Web site).

Turn in your writeup and talk on **September 15, 2011** by 11:59pm. *Please upload your solutions to T-Square. Other forms of submission will not be accepted!* We will be providing more information about how to turn in your assignment as the due date approaches.

1. **15 points** Pfleger and Pfleeger, Section 1.11, Exercise 14.
2. **15 points** Pfleeger and Pfleeger, Section 1.11, Exercise 20.
3. **20 points** The Ware report from 1970 emphasizes defenses for physical attacks and leakage points. In many of todays computing systems, however, physical attacks are relatively uncommon compared with other kinds of attacks.
   - Give three distinct reasons, with appropriate justications, why this is so.
   - In the Checkoway *et al.* paper on automotive security, the authors point out that it is possible to compromise the automotive system even without physical access to the car itself. From the paper identify at least one flaw of each type that creates a vulnerability: (1) design; (2) implementation; (3) process
4. **20 points** Discuss Ken Thompson's *Reflections on Trusting Trust* article in the context of Java applets. What, if anything, would need to be changed? What about byte-code verifiers, interpreted languages, disassemblers, etc. might you have to consider?
5. **30 points**

   print_display.c:

```
#include <stdlib.h>
#include <stdio.h>

int main(){
  char display[512];

  strcpy(display, getenv("DISPLAY"));
  printf("Your display environment variable is %s\n", display);

  return(0);

}
```

The system administrator accidentally set the suid (set user id) bit on this program, which is owned by root and lives in /usr/bin.

- Describe the vulnerability present in this program.
- Write a program to exploit this vulnerability and obtain the root shell. Be clear and concise. If you use a language other than C or Perl, please be extra clear.
- Argue/demonstrate that your exploit works.
- Rewrite print_display.c to be safe.