CS 4235: Introduction to Information Security September 15, 2011

Problem Set 3: 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 **September 29, 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** Pfleeger and Pfleeger, Section 3.10, Exercise 15. For the "design requirements", concentrate on specifying a security policy for the processor, in terms of both its abstract functionality and physical properties.

2. **15 points** Consider Automated Teller Machines (ATMs), which use a customers bank card and secret Personal Identication Number (PIN) for common banking tasks like withdrawals, checking account balances, etc.

   Identify at least three distinct input/output paths on an ATM, and name the endpoints of each.

   For each of the above paths, describe the extent to which it qualifies (or doesn't) as a "trusted I/O path". Focus on condentiality and integrity of data as it travels between the two endpoints. For this analysis, you may want to research some of the relevant known attacks on ATMs.

3. **20 points** Find an example of a specic security flaw in a commonly used commercial operating system (e.g., Windows, Linux, Mac OS X) in the past year. Good sources include CERT, the Microsoft Security TechCenter, or `securityfocus.com`.

   Choose a flaw that has very signicant security implications (e.g., favor arbitrary remote code execution over local denial of service). Give a high-level summary of the flaw and its implications, in your own words. Classify the nature and cause of the vulnerability: for example, is it the result of flawed code in an unsafe language? an incomplete or inconsistent specication? a flawed design in terms of modularity and/or encapsulation? Justify your answer.

4. **20 points** In UNIX, the Internet Daemon (now called xinetd on some versions of UNIX) provides the handshaking that occurs when a TCP/IP connection. Xinetd is susceptible to a Denial of Serice attack, where many connections are made to the same service. When too many connections are made within a specified short period of time, xinetd will terminate that service for a short period of time and print error messages of the form:

   ```
   inetd[354]: telnet/fcp server failing (looping), service terminated
   ```

Various attack programs exist to launch thousands of connections on a specific port, overloading the machine. See `http://www.cotse.com/dos.htm` for some examples of source code designed to mount denial of service attacks.

Explain the design alternatives that the designers of an Internet service like xinetd considered when they decided to implement inetd. What are some alternative designs that solve this vulnerability? Do they introduce new vulnerabilities?

5. **30 points** In this problem, you will try to understand how UNIX generates password files, and then try to crack some passwords!

   - Examine the source code or man page for crypt. How does this program take a plaintext password and generate the ciphertext that we see in /etc/password, or /etc/shadow? What cipher is used to generate the cipher from the plaintext?
   - Using crypt(3) on a UNIX machine, generate the ciphertext for security and netsecurity. What do you observe? Why?
   - `passwd` typically uses something called a salt to generate the password for each user. Why?
   - Consider the following password file generated with crypt(3):

     ```
     root:IWpIzqD0jR1.c:100:100:Charlie Root:/home/root:/bin/sh
     cs4251:UNzrFi5aYL9DU:101:101:CS4251:/home/cs4251:/bin/sh
     mysql:WqCBVG36lcuAc:102:102:MySQL:/home/mysql:/bin/sh
     guest:FTQinpjr.VRM.:103:103:Guest:/home/guest:/bin/sh
     test:LF2c9qM5l6X7Q:104:104:Testing:/home/test:/bin/sh
     ```

     Run the default mode of John the Ripper (`http:www.openwall.com/john/`) on the password file. One of the passwords will be cracked. Which one? Why (which rule of John was applied)? One of these passwords will be cracked. Which one? Why (which rule of John was applied)?
   - Try the "wordlist" mode of John. Which password is cracked now? Which rule of John was applied? (*Hint:* John's default wordlist is very small by default. You may have to augment this wordlist with one of your own.)
   - One of the users has a password that is a rotated version of a dictionary word. Modify Johns rule list to incorporate this feature. Which password does this now reveal? Please include the source for your modifications to the rule list.
   - One user is predisposed to using leetspeak (4 for a/A, 1 for i/I and l, 3 for e/E). His password is also a dictionary word. Modify john.conf to incorporate this feature. Which password is revealed?
   - One user likes to swap two adjacent characters of a dictionary word. Can you modify john.conf to do this using existing syntax? If not, how can you incorporate this feature? What is the password that is revealed?