CS 380S - Theory and Practice of Secure Systems
Fall 2009

Homework #2

Due: 2pm CST (in class), November 5, 2009

NO LATE SUBMISSIONS WILL BE ACCEPTED

YOUR NAME: _____

Collaboration policy

No collaboration is permitted on this assignment. Any cheating (*e.g.*, submitting another person's work as your own, or permitting your work to be copied) will automatically result in a failing grade. The Computer Sciences department code of conduct can be found at `http://www.cs.utexas.edu/academics/conduct/`

Homework #2 (30 points)

Problem 1

Consider the following snippet of C code:

```
static char logfile[]="log.txt";

int checklog() {
  return (access(logfile, O_RDWR));
}
int openlog() {
  return (open(logfile, O_RDWR));
}
int writelog(int fd, int len) {
  if (fd > 0)
     write(fd, "", len);
}
void log() {
  int fd;
  if (checklog()) {
     fd=openlog();
     for(int i=0; i<5; i++)
        writelog(fd,64);
     writelog(fd,32);
  }
}
```

Problem 1a (5 points)

Write the Dyck model of the control flow of this code.

Problem 1b (2 points)

Can you use a reference monitor based on the above Dyck model to prevent TOCTTOU attacks on this code? If yes, explain how; if no, explain why.

Problem 2 (4 points)

Consider extending the technique for detecting SQL injection vulnerabilities described in the PLDI 2007 paper by Wassermann and Su ("Sound and Precise Analysis of Web Applications for Injection Vulnerabilities") to detect cross-site scripting vulnerabilities.

What do you think would be the main difficulties?

Problem 3 (3 points)

Suppose you want to design a client-side protection against XSRF attacks, using an HTTP proxy or a browser plugin. What are the main issues to solve?

Problem 4

Problem 4a (2 points)

Explain why enforcement of the same-origin policy in modern browsers fundamentally relies on the integrity of the DNS (Domain Name System).

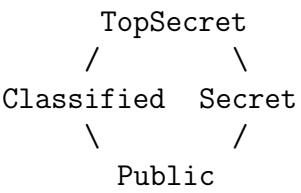
Problem 4b (3 points)

One solution for enforcing the same-origin policy even when DNS may be compromised is to associate the server's public key with each Web object retrieved over HTTPS. When one Web object attempts to access another object, the browser checks whether the keys match.

Does this solution provide complete enforcement of the same-origin policy, or can it still be subverted for certain Web objects by an attacker who controls DNS? Explain.

Problem 5 (3 points)

Consider the following lattice of security labels (the higher the label, the more confidential the information):



and consider the following pseudocode:

```
Public     w
Classified x
Secret     y
TopSecret  z

p = w - x
if (y != 0) then
   q = 1
   if (p == 0) then
      r = 0
   endif
else
   s = 1
endif
t = z - z
```

For each of the variables `p, q, r, s, t`, write the *minimal* security label that it can be given so that the above code is secure according to the Bell-LaPadula model.

Problem 6 (6 points)

What hard computational problem is equivalent to decrypting an ElGamal ciphertext without the private key? Prove your answer.

Problem 7 (2 points)

The Molvanian Institute of Cryptology suggests the following efficiency improvement to ElGamal. After generating a random k as part of creating an ElGamal ciphertext, save its value and re-use it for subsequent encryptions (like many other things, randomness is hard to come by in Molvania).

Would this modification have any impact on the security of ElGamal encryption? Explain.