Homework1 for CS165 (Fall 2017)

Due: Monday Oct 27

Instructions:

• Be brief. You will be graded for correctness, not on the length of your answers.
• Make sure to write legibly. Incomprehensible writing will be assumed to be incorrect.

Q1. A special form of self-reproducing code is called “quine”: a quine is a non-empty computer program which takes no input and produces a copy of its own source code as its only output. Following is a one-line quine in C, with some parts of the program missing.

```c
main(){char *c=s0;printf(c,p0);}
```

a. What should be the missing parts (s0, p0, p1)? (3 points)
b. Explain how the program works when compiled and executed (i.e., how it prints itself). (5 points)

Q2. Both CodeRed and Slammar are uses random scan to probe and propagate themselves. Their infection rate fits into a model called random constant spread (RCS) model. However, Slammer spread nearly two orders of magnitude faster than CodeRed. Read the Slammer paper and answer following questions.

a. In the RCS model, why the infection rate will saturate eventually? (2 points)
b. The spread rate of Slammer saturated much earlier than the RCS model, why? (2 points)
c. What contributes to the much faster spread rate of Slammer? (4 points)

Q3. Malware analysis is an important step to understand malware (e.g., its infection method, possible malicious behavior, and the criminal infrastructure it uses). To impede malware analysis, malware authors have developed many anti-analysis techniques.

a. What is the most popular technique to prevent static analysis? How it works? (1+2 points)
b. We’ve discussed many ways to detect virtual machine based analysis environment, choose one method and explains how it works. (3 points)

Q4. In a slightly modified “example2.c”:

```c
void function(char *str) {
    char buffer[16];
    strcpy(buffer, str);
    printf("%s", buffer);
}
```

1
void main() {
    char large_string[256];
    int i;
    for (i = 0; i < 255; i++)
        large_string[i] = 'A' + i;
    function(large_string);
}

a. When function returns, a segment fault will be generated at which address? Why? (1+1 points)
b. How would you fix the vulnerability using strncpy? (2 points)
c. How many character are in the printed out string? Why? (2 points)

Q5. Read the StackShield and StackGuard bypassing paper and answer following questions.
   a. StackGuard v2.0.1 uses a constant 0x000aff0d as canary, why this would work? Why can’t attackers overwrite the canary with the same value? (2 points)
   b. What is StackShield? How it differs from stack canary? (2 points)
   c. How exactly does Microsoft’s /GS protection works? (2 points)

Q6. Read the Eternal War in Memory paper and answer the following questions.
   a. What are the three solutions that can provide spatial memory safety? Why they’re not adopted in practice? (3 points)
   b. What are the three solutions that can provide temporal memory safety? (3 points)