

ECE 568: Assignment 1

Introduction

The following four questions will ask you to have a more in-depth look at a number of the topics that we've been discussing thus far in lecture. Answers are to be submitted in hard-copy in lecture (see below), and work must be done *individually*.

Formal Code Verification

Please read the article *Hacker-Proof Code Confirmed* available from:

<http://ece568.ca/a1>

While formal code verification holds many possible benefits, the author provides a number of examples of hurdles that stand in the way of widespread adoption.

1. Briefly describe the idea behind formal code verification. (2 marks)
2. Briefly describe two challenges that have prevented it from being widely adopted for commercial software. (4 marks)

Return-Oriented Programming

Please download and read the *Return Oriented Programming* paper from the ACM website (via the UofT Library): <http://dl.acm.org.myaccess.library.utoronto.ca/citation.cfm?id=2133377>

In this paper, the authors discuss a style of exploit that bypasses many of the protections that we have discussed in class: Return-Oriented Programming.

1. Briefly describe " $W \oplus X$ " protection, and explain why it would prevent stack-smashing attacks. (2 marks)
2. Briefly explain what "gadgets" are, and how they are used in ROP. (2 marks)

ECE 568: Assignment 1

Computer Virus-Antivirus Coevolution

Please download and read the *Computer Virus-Antivirus Coevolution* paper from the ACM website (via the UofT Library):

<http://dl.acm.org.myaccess.library.utoronto.ca/citation.cfm?id=242869>

In this paper, Carey Nachenberg discusses the evolution of obfuscation techniques used by computer viruses in the 1990's. Polymorphism, the most advanced obfuscation technique at the time, is increasingly used by today's computer worms to evade detection. One example is the notorious Conficker worm, which infected millions of computers in 2009.

1. Describe the challenge(s) does polymorphic virus pose for signature-based detection. (2 marks)
2. Describe two different techniques that a polymorphic virus can utilize to make a GD antivirus program less effective. (3 marks)

Programming Error

Here is an example of a program with a vulnerability in it. Assume that input comes from an untrusted source that could be malicious.

```
1: void bar(char *arg, char *targ, int len)
2: {
3:     int i;
4:     for (i = 0; i < len; i++) {
5:         *targ++ = *arg++;
6:     }
7: }
8:
9: int foo(char *arg)
10: {
11:     char buf[80];
12:     char outbuf[64];
13:
14:     fgets(buf, 80, stdin);
15:     printf("%s", buf);
16:     bar(arg, buf, 63);
17:     strcpy(outbuf, buf);
18:     printf("%s", outbuf);
19:     return(0);
20: }
```

ECE 568: Assignment 1

Considering the above code:

1. What is the vulnerability? What does it allow the attacker to do? (2 marks)
2. What line(s) would you change and what would you change them to correct the vulnerability? (Other than correcting the vulnerability, you must keep the functionality of the program the same.) (3 marks)

Submission

Please treat this as you would a business report: your answers should be written in with full sentences. (We reserve the right to deduct marks for poor English, unintelligible answers or illegible handwriting.) All answers should be written in your own words - no copy-and-pasting! The completed assignments should be submitted in hardcopy during class during the week of October 17, 2016.

Please note that all written assignments are to be done individually.

Total: 20 marks.