Security Engineering - Winter 2017

Sambuddho Chakravarty

March 4, 2017

Assignment 2 (Total points: 40)

Due date: March 19, 2017. Time: 23:59 Hrs.

Basic Buffer Overflow Vulnerability

The objective of this assignment is to familiarize you with writing shellcode and exploiting programs. You need to write a shell code using assembly language using any assembler of your choice (GNU AS, NASM *etc.*).

The shellcode should do the following:

- Delete all files in the directory in which it is running (Hint: you may want to read up the manpages for `unlink(2)` system call)
- Reboot the system.

Once you have the shellcode ready you need to devise a way to pass it as an input to program (binary) so that at termination the input code is executed – thereby the files of the directory are deleted and the system reboots.

The (victim) program is written for X86_64 architecture. Following stack smashing attack protections turned off:

- Address Space Layout Randomization (ASLR).
- Stack smashing protection (SSP).
- Protections to turn off executable stacks (executable stack allowed).

The victim program binary is uploaded on backpack.

Grading Rubric

- Successful compilation of the shellcode using Makefile – 5 points.
- Working standalone shellcode that uses system calls to implement the two operations – 10 points.
- Correctly passing the shellcode to the program forcing it to correctly execute it – 10 points.
- Description of the systems, commands to execute and test the program and the assumptions that you made – 5 points.

Late Submission Policy

- Submitted on or before March 19, 2017 (23:59 hrs) – No points deducted.
- Submitted after March 19, 2017 but on or before March 21, 2017 (23:59 hrs) – 5 points deducted.
- Submitted after March 21, 2017 but on or before March 23, 2017 (23:59 hrs) – 10 points deducted.
- No points for assignments submitted after March 24 (0000 hrs)