CS6542 - Graduate –Computer Network Defense – Spring 2017

Lab 12-1 page 291
Lab 12-3 page 299
Due 7st March 2017, 6:00 pm.
100 points

Policy on collaboration: All examinations, papers, and other graded work products and assignments are to be completed in conformance with The George Washington University Code of Academic Integrity. Each student is expected to write his or her own HW out independently; you may not copy one another's assignments, even in part. You may not collaborate with others on the test and final.
You are expected to cite all your sources in any written work that is not closed book: papers, books, web sites, discussions with others - faculty, friends, students. For example, if, in a group, one student has a major idea that leads to a solution to a HW problem, all other students in the group should cite this student.
You may not refer to solutions to previous years' problem sets, or ask for help students from previous years. Any violations will be treated as violations of the Code of Academic Integrity.

Please work on each lab and capture screenshot of tasks along with your words and analysis of each slide. PLEASE submit all Labs on Blackboard only. Name your files:

PLEASE submit all Project on Blackboard only

Late submission
Please note that, there is a %10 penalty for late submission until next project due date, and also there is NO grade for project submission after the next project due date.

1. Lab 12-1 Compiling on Windows

In this lab, we focus on working with Windows 7 VM. Therefore, all the system requirements and software installation processes need to be install on your Windows 7 VM.
The Microsoft C/C++ Optimizing Compiler and Linker are available for free from www.microsoft.com/express/download/. Select the Express 2013 for Windows or Express 2013 for Windows Desktop option. After the download and a straightforward installation, you'll have a Start menu link to the Visual Studio 2013 Express edition. Click the Windows Start button, followed by All Programs | Visual Studio 2013 | Visual Studio Tools. This will bring up a window showing various command prompt shortcuts. Double-click the one titled "Developer Command Prompt for VS2013." This is a special command prompt with the environment set up for compiling your code.

1.1. System Requirements:

1.1.1. Download Visual Studio 2013 Express:

Go to:

http://www.microsoft.com/express/download/

To download the free version of Microsoft C/C++ Optimizing Compiler Express 2013 for Windows. Follow the default installation instructions.

1.1.2. Download CodeBlocks

You have already install this software for previous labs, however, if you removed the software or re-image your Windows 7 VM please go ahead and install it again. You will use CodeBlocks software for creating C program files.

1.2. Tasks

1. Create hello.c file

```
//hello.c
#include <stdio.h>
main ( ) {
printf("Hello haxor");
}
```

2. Compile hello.c file with windows compiler and get execution file as output:
   a. For compiling a C file, you would need to navigate to Start\Visual Studio 2013\Visual Studio Tools\Developer Command Prompt for VS2013
   b. After Development command prompt windows open, please navigate to your hello.c file path
   c. Now you can use the compiler command:

```
C:\Users\hadimoh\Lab>cl hello.c
```

3. Create meet.c file:

```
//meet.c
#include <stdio.h>
greeting(char *temp1, char *temp2) {
char name[400];
strcpy(name, temp2);
printf("Hello %s %s\n", temp1, name);
}
main(int argc, char *argv[]){
greeting(argv[1], argv[2]);
printf("Bye %s %s\n", argv[1], argv[2]);
}
```

4. Compile meet.c file with windows compiler and get execution file as output.
5. Test your meet.exe file and take the screenshot of the output.
6. Compile meet.c with full debugging information and disable the stack canary function.

2. Lab 12-3 Exploiting ProSSHD Server:

You will need to download and install Immunity Debugger onto your Windows 7 VM from the aforementioned link. Immunity Debugger has a dependency on Python 2.7.1, which will install automatically during installation if not already on your system.

2.1. System Requirements:

You will be debugging the meet.exe program you previously compiled. Using Python IDLE on your Windows 7 system, type in the following:

2.1.1. Download Python 2.7.1:

Go to:

https://www.python.org/downloads/

Follow the default installation instructions, allowing Python 2.7.1 to be installed if necessary and install is on your c:\Python path.

2.1.2. Download Immunity Debugger version 1.85:

Go to:

https://github.com/kbandla/ImmunityDebugger

Also you can download it from the following link by registering process (recommended):

http://www.immunityinc.com/products/debugger/index.html

to download the free Immunity Debugger from Immunity Inc.

2.1.3. Download ProSSHD

The ProSSHD server is a network SSH server that allows users to connect “securely” and provides shell access over an encrypted channel. The server runs on port 22. A couple of years back, an advisory was released that warned of a buffer overflow for a post-authentication action. This means the user must already have an account on the server to exploit the vulnerability. **The vulnerability may be exploited by sending more than 500 bytes to the path string of an SCP GET command.**

Go to:

http://www.labtam-inc.com/articles/prosshd-1-2.html

to download the free trial version of ProSSHD v1.2. Follow the instructions to sign up for the free trial key.

Follow the default installation instructions, allowing Python 2.7 to be installed if necessary. Please ensure you are using Windows 7 SP1.

2.1.4. Download scpclient module for Python:

Go to:

https://pypi.python.org/packages/source/s/scpclient/scpclient-0.4.tar.gz to download the scpclient module for Python. Follow the default installation instructions.

2.2. Tasks

1) Download and install the ProSSHD application. Set up the vulnerable ProSSHD v1.2 server on a VMware guest virtual machine running Windows 7 SP1.
   a) **CAUTION** Because we are running a vulnerable program, the safest way to conduct testing is to place the virtual NIC of VMware in host-only networking mode. This will ensure that no outside machines can connect to our vulnerable virtual machine. See the VMware documentation (www.vmware.com) for more information.
2) Run the program from the installation path and setup an exception for ProSSHD if excited on your VM.
   a) **NOTE** If Data Execution Prevention (DEP) is running for all programs and services on your target virtual machine, you will need to set up an exception for ProSSHD for the time being. We will turn DEP back on in a later example to show you the process of using ROP to modify permissions when DEP is enabled. The fastest way to check is by holding the Windows key and pressing break from your keyboard to bring up the System Control Panel. From the left, click "Advanced system settings". From the pop-up, click Settings from the Performance area. Click the right pane, titled "Data Execution Prevention." If the option "Turn on DEP for all programs and services except those I select" is the one already selected, you will need to put in an exception for the wsshd.exe and xwpsshd.exe programs. Simply click Add, select those two EXEs from the ProSSHD folder, and you are done!
3) Determine the IP address of the vulnerable server and ping the vulnerable virtual machine from your Kali Linux machine.
   a) You may need to allow the pings to reach the Windows virtual machine in its firewall settings.
4) Next, inside the virtual machine, open Immunity Debugger.
   a) You may wish to adjust the color scheme by right clicking in any window and selecting Appearance | Colors (All) and then choosing from the list. Scheme 4 is used for the examples in this section (white background). We have also selected the "No highlighting" option.
   b) At this point (the vulnerable application and the debugger are running on a vulnerable server but not attached yet), it is suggested that you save the state of the VMware virtual machine by saving a snapshot. After the snapshot is complete, you may return to this point by simply reverting to the snapshot. This trick will save you valuable testing time because you may skip all of the previous setup and reboots on subsequent iterations of testing.
5) Open up your favorite editor in your Kali Linux virtual machine and create a new file, saving it as prosshd1.py to verify the vulnerability of the server:

```
#prosshd1.py
# Based on original Exploit by S2 Crew [Hungary]
import paramiko
paramiko.util.log_to_file("filename.log")
from scpclient import *
from contextlib import closing
from time import sleep
import struct
hostname = "192.168.10.104"
username = "test1"
password = "asdf"
req = "A" * 500
ssh_client = paramiko.SSHClient()
ssh_client.load_system_host_keys()
ssh_client.connect(hostname, username=username,
key_filename=None, password=password)
sleep(15)
with closing(Read(ssh_client.get_transport(), req)) as
scp: scp.receive("foo.txt")
```

a) **NOTE** Remember to change the IP address in the script to match your vulnerable server.
b) **NOTE** The **paramiko** and **scpclient** modules are required for this script. The **paramiko** module should already be installed, but you will need to download and run setup.py for the scpclient module from https://pypi.python.org/packages/source/s/scpclient/scpclient-0.4.tar.gz.

```
root@kali# python setup.py install
```

c) You will also need to connect once with the default SSH client from a command shell on Kali Linux so that the vulnerable target server is in the known SSH hosts list.
d) Also, you may want to create a user account on the target virtual machine running ProSSHD that you will use in your exploit. We are using the username "test1" with a password of "asdf."

6) Launch the attack script from Kali and then quickly switch to the VMware target and attach Immunity Debugger to wsshd.exe.
7) Run Debugger on the Immunity Debugger software and check the EIP register.
   a) You should get the message like "Access violation when executing 0X41414141".