CSC347 Introduction to Information Security Assignment 1

Due: October 16, 2015 11:50PM

Late penalty: 20% if handed in up to October 18, 2015 11:50 PM. Not accepted after that.

Hand in: Electronic submit here. Please package your solution in a1.zip according to the submit checklist. Anything non-electronic, please drop off at my office (DH3088)

Marking:

Groups: Groups of size 2.

Note: All exploits are to take place under the RH7.2 assignment 1 Virtual Machine. You can find the assignment image under /virtual/assignment1/RH72LabImage_A1.zip on pc01 in the lab, with md5sum `94e5ee12da7dba5de52df8dcc0f7ac38 RH72LabImage_A1_2015.zip`. You can use the following script to download it to your lab machine. Select 'I MOVED IT'. Unless specified in the Q and A section below, all official code for this assignment is on the VM. Although you have been given the source code, you should assume that your exploit involves us running our unmodified, compiled copy of the code under the rh7.2 assignment image with your inputs.

1. **[0 Marks: Practice]** Solve v3.c as it appears in the hackers account. Try exercises 1 and 2 in here in tutorial, ask Tapan for some help.
2. **[5 Marks]** In the RH72LabImage_A1_2015 you can find swapAround.c. If it has a buffer overrun vulnerability, demonstrate, explain and exploit it, launching a shell. If it does not have a vulnerability, explain.
3. **[25 Marks]** Your assignment VM has code installed in `/vulnerable`. Your can find this question in `account.c`. Unoffical code is also in vulnerable.tar.gz. Your task for this question is to identify vulnerabilities (mark them in the code), demonstrate exploits and explain the exploits impact. See for example, cert.org vulnerability notes. You do not have to exploit buffer overrun vulnerabilities for this part of the assignment, just identify vulnerable code and explain the potential impact. Write a report (see REPORT.txt) listing each vulnerability, a collection of exploits for the vulnerability (code, inputs, scripts etc. we can run to see the exploit in action), the impact of the vulnerability, and identify a category (CIA) for the impact (see the list below). Include inputs, scripts, code etc. demonstrating each exploit.

   To think about this question, think about the following: Secure properties (CIA)

   - Confidentiality
   - Integrity
   - Availability

   Vulnerabilities to look for include, buffer overruns, integer overflows, canonical naming,... Once you find a vulnerability, demonstrate associated exploits (code/inputs). Then describe the impact of the exploit. Impacts include things like denial of service, authentication issues, accountability issues, priviledge escalation (see principle of least priviledge), ... Finally identify the impact as a violation of Confidentiality, Integrity or Availability. **Hint:** Thinking in terms of the above list is a good way to start thinking about potential exploits.

   Finally fix the code, explaining how you fixed the vulnerabilities and prevented the exploit and restored CIA.

4. **[15 Marks]** xinetd is the internet super server, it can turn any unix program that reads stdin and writes stdout into a 'web service'. Take a look at `/etc/xinetd.d` in the assignment VM to see a collection of programs that are running under xinetd on your RH7.2 VM. One of these is palindrome.c, please take a look at `/etc/xinetd.d/palindrome` to see how this service is run (which port etc). For convenience, I am

having it run out of `/root/a1` where you can find the source code as well. Simply recompiling /root/a1/palindrome.c will cause the online service to update. If you want to restart `xinetd` you should execute `/etc/rc.d/init.d/xinetd restart`.

You will investigate this service, show how a user from outside the RH7.2 VM (ie on the host machine) can gain unauthorized access to the VM. You might want to take a look at tcpclient.pl. Use `ifconfig` to determine the ip address of the VM.

a. Submit a modified tcpclient.pl (called `tcpclientA.pl`) which obtains a root shell on the remote server running `palindrome`. The console user running `tcpclientA.pl` should be able to interactively enter commands and see the result. **Note:** Entering commands may not be as smooth as running a real console, this is ok. At the very least, a user should be able to execute `ls` a couple of times.
b. (**Note:** Do either b or c) Submit a modified tcpclient.pl (called `tcpclientB.pl`) which creates the file '/root/iveBeenHacked' on the remote server running `palindrome`. You can do this with the `touch` unix command.
c. (**Note:** Do either b or c) Challenge: Submit a modified tcpclient.pl (called `tcpclientC.pl`) which changes the root password to 'drowssap'.
d. Identify any other vulnerabilities and associated exploits as in the previous question. Write report similar to that of question 2.
e. Fix the palindrome service so that it can handle larger inputs and is no longer vulnerable. Submit the modified code as well as a discussion of any other changes you feel are necessary to make the `palindrome` service more secure.

Questions and Answers

Question:

```
What are the accounts on the vm?
```

Answer:
root/password and hacker/password

Question:
I destroyed my makeShellCode.pl

Answer:
here is another copy.

Question:
I am getting permission denied when executing account.

Answer:
Perhaps the permissions are wrong. I did not include the setup script in the VM. Execute this inside /vulnerable as root.