Assignment :: Common Vulnerability

Name: Weghanasa Cendana

Student ID: 12520716

**1.1 Savegames**

1. Set the character’s gold or health to a number greater than 9000 by utilising a buffer overflow. How did you achieve this? Explain using reference to bytes and ASCII as to what the exact value was that you achieved. [4 marks]

After I do some research on trying different number of character to the c program, I found that using the character from the range 1 to 11 does not affect the gold changes which is still represent 8 and still normal. However, after I try to increase the character into 12 characters, the gold turns into zero (0) because the scanner looks into the data and finds out that strings are terminated by a zero byte character. Furthermore, when adding more characters into 13 characters, the gold number is increasing by 1 value/byte for each character that I change into the higher character such as from a to b. Because of the string values was at 12 character, when having more character then the scanner will look up to the ASCII table and determine the value of the character. Moreover, by having 14 characters, the length of bytes is differentiated by 256 bytes for each character that I add or decrease based on the ASCII table, for example aaaaaaaaaaaaaa to aaaaaaaaaaaaab. Furthermore, by using 15 characters, the result between aaaaaaaaaaaaaaa to aaaaaaaaaaaaaaaab (for example) is differentiate by 65536 bytes. Thus, shows the result of 256 \*\* 2.

2. How could this exploit be prevented? [2 marks]

Buffer overflow can be prevented by avoiding the uses of library files, which these are used by the programming language and not secure. These are the target of the hackers during the application attack. Moreover, the uses of the managed language such as C#, Java, Python, etc are also important which these languages never allow any overflows, and reducing attack surface when implementing languages. Using the static analysis code is also one to mitigate the buffer overflows however this method is often expensive. In addition, mark the stack and the pages as a non-executable memory so that the CPU will give a notification to the kernel that the program cannot be executable then the kernel will stop the program. Furthermore, there are also automated tools to find the buffer overflow which is Fuzzing. Fuzzing is a testing program that can throw massive random data to the program, after that we launch the program which make it runs the testing and the bugs in the applications are more likely to appear. Then using the debugger and investigation to the code is next step to address the buffer overflow. The other method that we can apply is Stack canaries. Stack canaries is putting the canary using 4 or 8 bytes number before returning the address and then checking if the canary has been or has not been modified before it returns. Address Space Layout Randomisation (ASLR) also one of the method, so moving the program to the new random address every time the new start up operates. However it needs some compiler program to runs this method, also the dynamically linked libraries complicated the uses of this mode. Lastly, bound checking which check and detect the variable also one of the method to prevent buffer overflow. It works ensuring that the variable is fits and within the range of the given variable before executing the program.

3. Could this exploit be useful for more than just the game? Could it be used to gain access to a system? If not, why not? If so, where might it be used? [4 marks]

Yes buffer overflow can also be used to gain access a system as well because it is widespread in the operating system code and can perform the privilege escalation and gain an unlimited access to the computer’s resources not only for just cheating the game. The use of buffer overflow is to overwrite the existing data and take a control of it and somehow corrupts it. For example, using the ‘stack smashing buffer overflow’ which can bug the program and writes more data to buffer located on the stack than the actual data and this action result in the corruption of the data on the stack. However, when the overflow was making some mistakes, it leads to crashing and incorrectly operating of the system. These attack is so useful and strong for C and C++ programming languages because those are not performing any bound checking while executing the array.

**1.2 General Questions**

1. Why is it necessary for us to provide the flag *-fno-stack-protector* to GCC? What is a canary in terms of a buffer overflow and how can a canary prevent a buffer overflow exploit? [4 marks]

It is necessary to provide the flag –fno-stack-protector to GCC because in the default, the stack protector is off by default. Therefore, using the –fno-stack-protector commands to GCC can provide the code to compile and adjust with the current code line. Without the use of –fno-stack-protector commands, the system will run the last code that we have not compile. Therefore it is important to use –fno-stack-protector commands before running the current code.

On the other hand, canary is one of the most effective way to prevent and mitigating buffer overflow. Canary is a values (some 4 or 8 byte number) that are placed between a buffer and control data on the stack that act for observing the buffer overflow. Buffer overflow is exploiting by overwriting the memory from the lower to higher memory. Therefore, canary will also be overwritten. Then, the canary value is checked to make sure that anything was not changed or modified. If something was changed, thus buffer overflow is ongoing.

2. If the game above was written in Java instead of C, would the savegame still be exploitable? [2 marks]

No, it would be able to exploit because that savegame program was written in C languages which unnecessarily using bound checking to scan their code program. Therefore buffer overflow will still running on C program when bound checking was not apply to the program. On the other hand, Java languages is using bound checking every time the code run, therefore there will be none of buffer overflow that can attack Java languages. Moreover, the system on Java languages is if the value of the object was not fits or in between the give variable, thus immediately stop the program and cause an error.

3. Imagine you were exploiting a program that was running with escalated privileges (i.e. could read sensitive \_les, modify other users settings and so on) is it possible to obtain a BASH shell using buffer overflows? Be sure to explain what shellcode is and how the shellcode is executed.

[5 marks]

When the program is running with escalated privileges such as it can read sensitive\_les, modify users setting, etc, using buffer overflow to obtain their bash shell is a possible thing. BASH shell is a command shell which has the highest authority. Although it is more likely harder to gain access into the bash shell because it can run the escalated privileges. However, buffer overflow can still take control of the BASH shell.

Shellcode is a small piece machine code that created and used by the hacker for exploiting the vulnerability of the software. Therefore, the attacker can manipulate and take control the machine that runs in the command shell. There are two kinds of shellcode, it can be local or remote. Local shellcode is used when the hacker has a limited access to a machine but can exploit the vulnerability of the system, when this method was succeed, the attacker then has the same privilege as the targeted user. Moreover, remote shellcode is another type of the shellcode. By using this type of shellcode, the attacker can communicate and gain control to the target machine through the local network, intranet or remote network if it is successful. For the local shellcode, the shellcode will execute the same as buffer overflow did. Thus, using the overwritten to the data as it will not be checked by the system and then after the overflowing a buffer is done, then it changes the return address of the system execution in the shellcode address, as now it controllable. On the other hand, by using the remote shellcode, the user will download a malware program on the internet which they thought it was a real software. After they downloaded it, they will execute it and at that time remote shellcode will run. Therefore, the attacker can send a virus through the machine which can cause corrupt files, or slowing the machine speed.

**2 SQL Exploits**

1. Show how it is possible to log in as any user by performing an SQL

injection attack on the username/password login page. [2 marks]

In order to perform the SQL injection attack on username/password login, instead of using the actual username or password, using the written comments below to login is possible.

password' OR 1=1/\*

password' OR 1=1({

password' OR 1=1--

2. The website has been clued in on their major security problem and pre-vented the previous attack. Is it possible to use the status query to work

out the password of one of the administrators *Bobby*? [4 marks]

Yes, it is possible to know the password of administrators Bobby2. First, just type the username which is Bobby and the password is password’ OR 1=1/\*, then after the system says **>>> Find out the status for which user?** then, we type in this command **'UNION SELECT password from Users WHERE username='Bobby';--** we can now find out what is the password of Bobby. This command works by selecting the password from the specific user and display it to the screen. The output of the process will be **User ' UNION select password from Users WHERE username='Bobby';-- is lolcats**. For instance, we now know that the password is **lolcats**.

3. How can these attacks be prevented? Is it a difficult security problem

to fix? Why is it so common? [4 marks]

For protecting the code attacks by the hackers/ attacker. Therefore, using the good python code is essential. For example, the bad python code will use (“””SELECT 1 FROM Users WHERE username = ‘%s’ AND password = ‘%s’ “”” % (username, password)), by changing the code into a good python code (“””SELECT 1 FROM Users WHERE username = ? AND password = ?””” , (username, password)) then the attacker will not be able to login using the password’ OR 1=1/\* anymore. The different of using bad and good python code is: by using the bad python code, the code formats the provided string directly into an SQL query. However, by using the good python code, it allows the SQL library to take care of the converting Python datatypes into an SQL format.