Research
Week 10

CIT 288

Chaz Davis

BCTC
Spring 2020

March 7, 2020

Chapter 10

i) What does buffer overflow mean?
A buffer overflow occurs when data written to a buffer also corrupts data values in memory addresses adjacent to the destination buffer due to insufficient bounds checking. This can occur when copying data from one buffer to another without first checking that the data fits within the destination buffer.
ii) Are Windows systems the only vulnerable operating system?
No, I would say the notion that its the most vulnerable comes from the fact that it overwhelmingly olds most of the home computing dektop os market. That in itself makes it the largest target, therefore, the appearance of most vulnerable. No one was really trying to attack macs and/or linux, because they were less than two percent of home computing at one point. The largest advantage linux has, is that its open source, all of the code is out there for free, so anyone can see view it, find holes, and patch holes.
iii) Can you find an outside example of a non-windows overflow attack?
Yes, Linux, due to being C and C++ based, is the most susceptible system to buffer overflows. Simple C programs can be used to create buffer overflows. First you would want to allocate a slot of memory, and then you would want to pass arguments that it doesnt precheck. this is when Python programming becomes very, very handy. Say i create a program in c and allocate 500 bytes of memory to hold the variable. call it something like vuln.c

```c
#include <stdio.h>
#include <string.h>

int main (int argc, char** argv)
{
    char buffer[500];
    strcpy(buffer, argv[1]);

    return 0;
}
```

Now, you could simply pass "Hello" to the command line and all would work well, bu to create a simple overflow with a segfault error you could run something like `python -c 'print "\x41" * 508'` and it would save the 'a' character into memory 508 times, then we would also over write the entirety of the return address. This would have the return address as `0x414141` . So, we could set a virus or exploit at that point of address memory like this in assembler where the system interupt and gives you access to a root shell.
iv) What kind of locations do buffer overflows typically attack in a process address space?
It would attack the Base Pointer for the return space of the memory. they would target the stack, the heap, or the data section to do this.

v) What programming languages are susceptible to buffer overflow?
Compiled and memory managed languages, without bounds. SO, C, Cpp, or Assembler are the best, they provide direct access to root and low-level memory. Although, you can adapt just about any language to be exploited with the correct code and implementations.
vi) What does an off-by-one attack do to a system?
Buffer Overflow: Off-by-One. The program writes just past the bounds of allocated memory, which could corrupt data, crash the program, or lead to the execution of malicious code. ... The result is that information on the call stack is overwritten, including the function's return pointer.
vii) What is a Guard page? How effective is this defense in your opinion – no wrong answer here..
Guard pages are unmapped pages placed between all memory allocations of one page or larger. The guard page causes a segmentation fault upon any access. Thus useful in implementing protection for areas such as network interfacing, virtual machines, and interpreters

Guard pages have a high degree of overhead because they fragment the kernel's memory map and can increase the amount of virtual space considerably. Their effectiveness depends on the size and pattern of allocations; they are often more effective as a debugging facility than an operational security measure.
viii) Shellcode has to be "position independent". What does this mean?
Shellcode is the machine code that is injected into the flow of a program as the result of an exploit. It generally must be position independent as you can't usually control where it will be loaded in memory
ix) What are the possible consequences of a buffer overflow occurring?
The user would gain a login shell with root privileges ad be able to access all resources and all memory access.
x) What are the two key elements that must be identified in order to implement a buffer overflow?

- To identify a buffer vulnerability in some program that can be triggered using externally sourced data ubder the attackers control
- To understand how that buffer will be stored in the process memory, and hence the potential for corrupting adjacent memory locations and potentially altering the flow of execution of the program

xi) List some of the different operations an attacker may design shell code to perform.

- Creating a remote shell to listen to services performed on the user's system.
- Creating a reverse shell to return control of the user's system.
- Create a shell on the user system by using local exploits in the program.
- Break out of the firewall rules in the system to avoid attack prevention.

xii) List and briefly describe some of the defenses against buffer overflows that can be used when compiling new programs.
Developing a program with modern high-level programming language. Using safe code techniques like buffer validation and safe functions. Using safe library to implement the program. Using some stack protection mechanism
xiii) Describe how a global data area overflow attack is implemented.
The buffer which is targeted is located at the global data area of a program. The buffer which is used as global has been overwritten by using unsafe buffer operations; it changes the pointer memory locations used in the buffer. The program which is attacked will make a call to overwritten function; the function will send the program's control to the shellcode which is written by the attacker.
xiv) Do you think Linux or Microsoft web servers are the most vulnerable? This is open ended. Don't get too lengthy - but look into it a bit and give a response.
I believe that Linux is the more secure option overall. I must say that based on bas-configuration out-of-the-box, windows is probably a touch more secure. But as you decide to work on microsoft servers, adding additional programs, users, and the like, thats when microsoft servers get weaker. Those are also the things that make Linux Stronger. The more you tinker in the correct ways, the more secure and less vulnerable they will be. While both are meant to work on an abundance of hardwares, i believe that linux works better on alder out of date architectures than windows. Windows at somepoint puts an end of life on their software. Linux doesnt, its all available out there, so if I wanted to fire up a PC and run the second linux kernel version. its there for me to do so, and chances are more tinkerers are out there anbd theyve found veulnerabilities and ways to patch those holes. and ways to make it run on things it wasnt intended to run on. You have the ability to modify and write drivers for any device you want to move things to, as well.