**Chapter 2**

**Variables**

* What is a variable?
  + An object that holds data that can be changed
* What is variable scope?
  + The portion of a program where a value is known or available
* Local vs Static vs Global
  + Local is only available in the function it is declared in
  + Static is local inside of its functions but global when declared outside of functions
  + Global is available throughout the entire file system and all functions within
* What are variable types?
  + The type of data that is stored in a variable such as Char or Int
* What is a pointer?
  + A variable that stores the memory address of another variable in memory
  + What can a pointer point to?
    - Any variable or function that matches its declared data type
* What is an array?
  + An object that holds a group of similar variable types
* What is a string?
  + An object that holds a group of char variables
* What is casting?
  + Typecasting is when you change a variable from one data type to another

**Functions**

* What is a function prototype?
  + The declaration of a function that declares its name and type
* How can you tell the return type of a function?
  + A functions return type should match its function declaration
* How do you pass arguments to a function?
  + By adding your arguments to the function prototype
* What is a struct?
  + Similar to an array it can hold a group of variables but they do not have to have the same data type like arrays

**Computer Architecture**

* What is a register?
  + Similar to variables they hold data for processing
* What are the x86 registers and what are they used for?
  + 8 general purpose registers that are used for address calculations, arithmetic, and logical calculations
* What is meant by little endianness?
  + It describes the idea that the least significant bytes are stored before the most significant bytes

**Assembly language**

* What is assembly language?
  + A low-level machine language that is meant to communicate directly with a machines hardware
* What are the sources for the values used in an assembly instruction?
* What happens when the call instruction executes?
  + The call instruction puts the data onto the stack
* What is a function prologue?
  + It is typically a few lines of code at the beginning of the function which prepare the stack for use
* How are function arguments accessed?
  + By referencing the ESP
* How are local variables accessed?
  + By referencing the EBP

**Concepts of a running process**

* What sections is a program divided into?
  + Data section
  + BSS section
  + Text section
* What kind of data is stored in each section?
  + Data declares data or constants
  + BSS declares variables
  + Test section keeps actual code
* What is the stack and how does it work?
  + An array like structure in memory where data can be stored and removed
* What is a stack frame?
  + An organization structure that helps organize local data for function calls
* How can you view the details of a stack frame in gdb?
  + Using the backtrace command

**Misc**

* What is a file descriptor?
  + A number that identifies an files within Unix
* What is a format parameter?
  + Specifies whether to return the original file to the repository rather than running the file

**Chapter 3**

**Buffer Overflow**

* What is it?
  + A system error that occurs when the volume of data being put into a buffer exceeds the storage capacity of the buffer
* How can a buffer overflow lead to gaining code execution?
  + The data that overflows can allow for the execution of malicious code
* Explain a buffer overflow from start to finish
  + A buffer is allocated memory, say 8 bytes. When that buffer is accessed by the program and data is assigned to it, the program assumes that only 8 bytes of data will be assigned, but instead the user attempts to put 10 bytes of memory into the program instead causing an overflow of data, as the excess data spills over into the next memory address. This excess data can be malicious code that is then executed by the machine whenever that memory address is accessed.

**Exploitation concepts**

* Why does using strcpy make a program vulnerable?
  + It is vulnerable because the function does not specify the size of the destination array, so if the destination array is smaller than the size of the string, it can lead to a buffer overflow
* Why is the game of chance input\_name function vulnerable?
  + It utilizes the strcpy() function and allows the user to overflow the memory address and then execute the jackpot function by setting that as the next memory address to be accessed.
* What are some memory locations you might target with an arbitrary write primitive
  + Stack memory, Heap memory, and Global variables

**Chapter 4**

* What are the 7 layers of the OSI model?
  + Please date Nancy Trotter she plays Archery
  + Physical, Data Link, Network, Transport, Session, Presentation, Application
* What is a socket?
  + An endpoint of a two-way communication link that allows computers to communicate with each other over a network
* What is a stream socket?
  + A reliable connection-oriented socket that primarily uses TCP as its protocol.
* What is a datagram socket?
  + A unreliable connection-oriented socket that primarily uses UDP as its protocol.
* What is an ARP message?
  + ARP is a type of network packet that is used to map a network layer address
* What is the difference between TCP and UDP?
  + TCP is a more secure and reliable connection protocol that has error-validation and establishes connection before data is sent
  + UDP is a connectionless protocol that is less reliable than TCP due to its lacking of error-checking but it is far faster in transmission than TCP
* What is a port scan?
  + A cybersecurity technique that is used to verify closed and open ports on networks or computers
* What is a bind shell?
  + A type of shell that is created on a computer that is bound to a specific port allowing remote connection to that particular port