Buffer Overflow

# Buffer? Overlow?

* Buffer – temporärer Platz im Speicher der zur Speicherung von Daten genutzt wird
* Buffer Overflow – Passiert, wenn Daten in den Buffer geschrieben werden, die Größer sind als der Buffer und aufgrund nicht ausreichend geprüfter Grenzen des Buffer, und angrenzende Speicherbereiche überschreiben

# Eine einfache angreifbare Funktion

GetInput()

{

char buffer[8];

gets(buffer);

puts(buffer);

}

Gets() prüft nicht ob die Größe der Eingabe > als die Größe des Buffers

# Remember Stack Layout



## Stack Layout unseres angreifbaren Programms



### Beobachtungen

* 12 Byte müssen aufgefüllt werden um zu RET zu kommen
* Dann müssen die nächsten 4 Byte gefüllt werden für den neuen Wert von RET
* Wenn die Funktion zu Ende ist, wird der neue RET Wert genutzt um zu entscheiden was als nächstes ausgeführt wird

# Shellcode

* Wenn wir die Kontrolle über die Return Adresse haben, können wir ihn auf „unseren ausführbaren Shellcode“ zeigen lassen
* Payload ist Maschinencode welcher direkt von der CPU ausgeführt wird
* Dieser Payload wird Shellcode genannt
* Standardmäßig wird es genutzt um eine Shell zu öffnen, daher der Name

## Code in Shellcode umwandeln

1. Code in C schreiben und ausführbare Datei erstellen
2. Disassemblen der ausführbaren Datei und Assembler äquivalent betrachten
3. Unnötigen Code entfernen
4. Code in Assembler neu schreiben
5. Objdump benutzen, um den Shellcode herauszufinden:  
   *objdump –d <assemblerdatei>*