

Systemsisicherheit - 4. Übung

Dennis Rotärmel, Niklas Entschladen, Tobias Ratajczyk, Gruppe Q

June 5, 2019

1 Aufgabe 1: Control-Flow Graph (CFG)

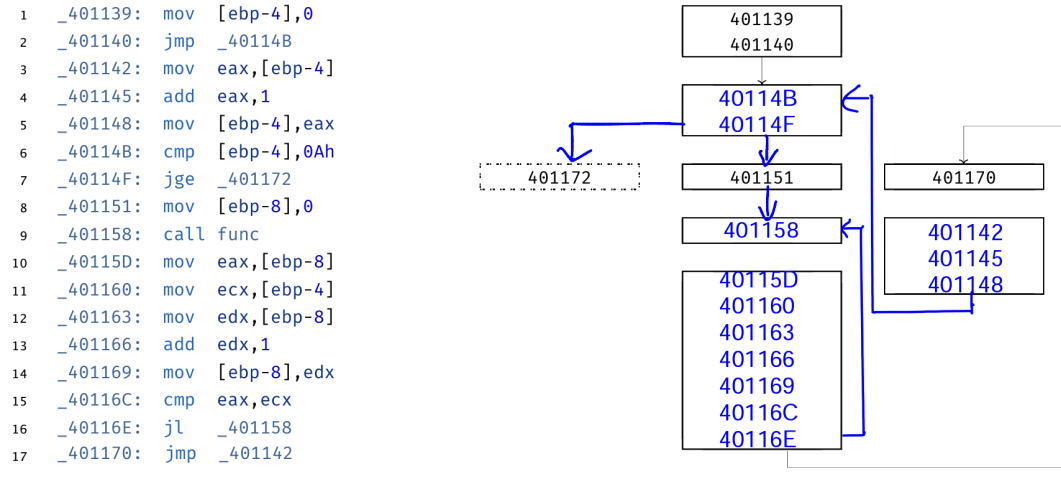


Figure 1: Control-Flow Graph

2 Aufgabe 2: Data Execution Prevention (DEP)

2.1 a)

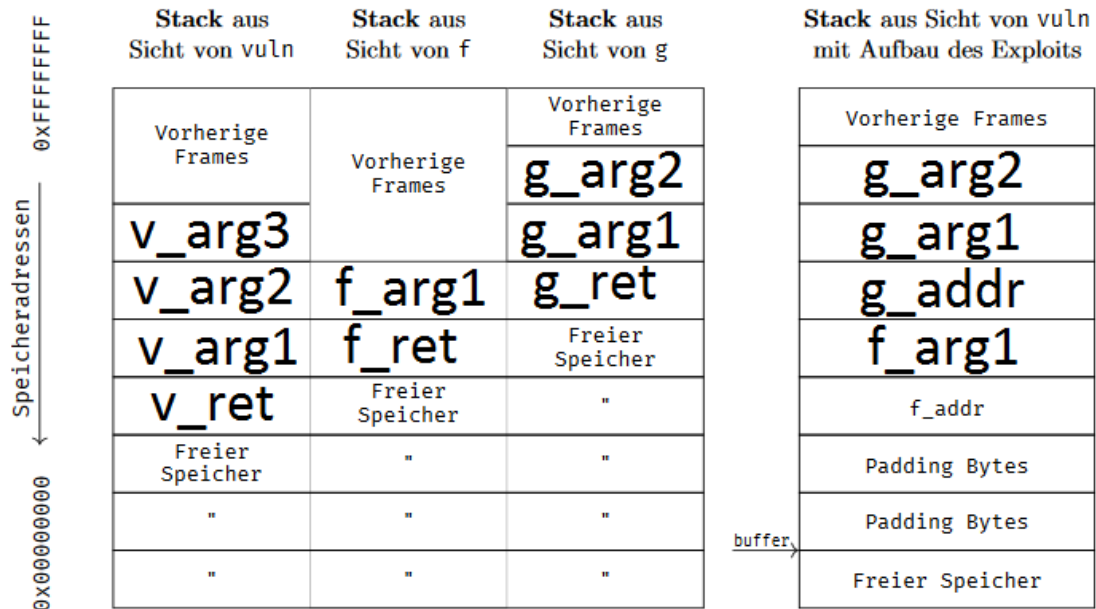


Figure 2: Control-Flow Graph

Sobald eine Funktion returnt, wird die nächste Instruktion im Stack ausgeführt. Dies stellt in unserem Fall den Sprung zu einer neuen Funktion dar, welche für einen Code Reuse Angriff benutzt werden.

Es wäre ein Problem, wenn die Funktion **f** einen Parameter mehr erwartet, da sie dann die Adresse der Funktion **g** als zweiten Parameter annimmt, während der erste Parameter der Funktion **g** fälschlicherweise die Adresse dieser Funktion darstellt. Für die Funktion **g** gilt nicht diese Einschränkung, da die Funktion **g** die letzte Library-Funktion der Chain darstellt. Die vorigen Stackframes haben keinen Einfluss auf die Code Reuse Attack.

Das Problem, wenn wir nach Funktion **f** und **g** noch eine weitere Funktion aufrufen wollen, liegt darin dass der Inhalt des vorigen Stackframes korrumpiert wird, sodass das Programm abstürzen könnte.

2.2 b)

3 Aufgabe 3: Stack Canaries

4 Aufgabe 4: Address Space Layout Randomization (ASLR)