**Zadanie č.4**

Úvod do počítačovej bezpečnosti

Zraniteľnosť programov

2016/2017

**Autor: Michal Drahovský, Ján Dzvoník**

Obsah

[1 Postup riešnia 3](#_Toc467076473)

[1.1 Funkcie na generovanie a aplikovanie saltu 3](#_Toc467076474)

[1.2 Funkcie na zahashovanie saltového hesla 3](#_Toc467076475)

[1.3 Funkcie na kontrolu zložitosti hesla 4](#_Toc467076476)

[1.4 Funkcie na vytvorenie časového odstupu medzi pokusmi o prihlásenie 6](#_Toc467076477)

[2 Záver 8](#_Toc467076478)

# Postup riešnia

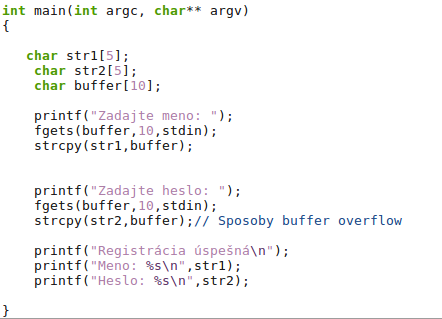
## Vytvorenie troch zraniteľných aplikácií

### Prvá aplikácia - Buffer Overflow

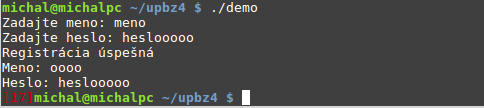
Buffer overflow je pretečenie zásobníka čiže chyba v programe, ktorá vedie k zápisu mimo vyhradeného priestoru v pamäti a k chybnému behu, prípadne aj k pádu programu.

Prvá aplikácia sa stará o registráciu uživateľov.

Ukážka kódu aplikácie



Ak do aplikácie zadáme neočakávané vstupy napríklad dlhšie heslo ako je vyhradená pamäť aplikácia sa správa neočakávane.

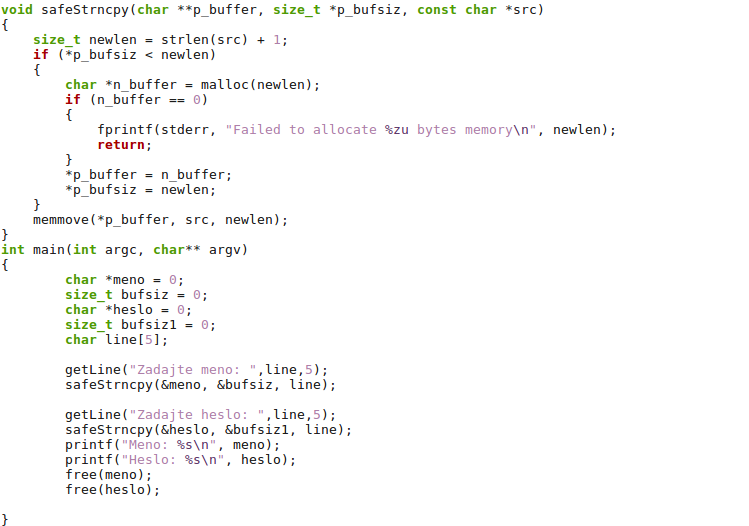


Aplikácie nemá ošetrené vstupy a tým pádom môže dôjsť k ich zneužitiu. Chyba aplikácie spočíva vo funkcií strcpy, ktorá spôsobí buffer overflow, čím je porušená pamäť a následne premenná str1 ktorá obsahuje meno je prepísaná.

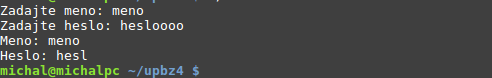
**Zabezpečenie kódu**

Funkciu strcpy sme nahradili vlastnou funkciou safeStrncpy, ktorá zabezpečí aby nedošlo k buffer overflow a tak zabezpečila aplikáciu.

Ukážka kódu funkcie safeStrncpy aj s upravenou funkciu main



Následne aplikácie už nie je zraniteľná voči buffer overflow, čo zobrazuje výstup



### Druhá aplikácia - Heap overflow

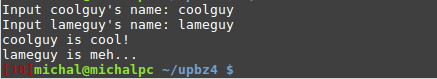
Heap overflow je pretečenie heap pamäte čiže chyba v programe, ktorá vedie k zápisu mimo vyhradeného priestoru v pamäti a k chybnému behu, prípadne aj k pádu programu.

Druhá aplikácia dynamicky vytvára dva typy užívateľov, ktorý majú vnútornú funkcionalitu. Touto funkcionalitou dokážu vypísať zadanú správu.

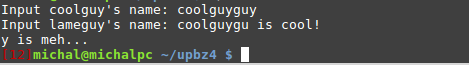
Ukážka kódu aplikácie



Po zadaní mena sú volané funkcie jednotlivých objektov s vnútornou správou.



Ak do aplikácie zadávme neočakávaný vstup napríklad dlhší ako je jeho vyhradené miesto v pamäti dôjde zvláštnemu správaniu aplikácie.



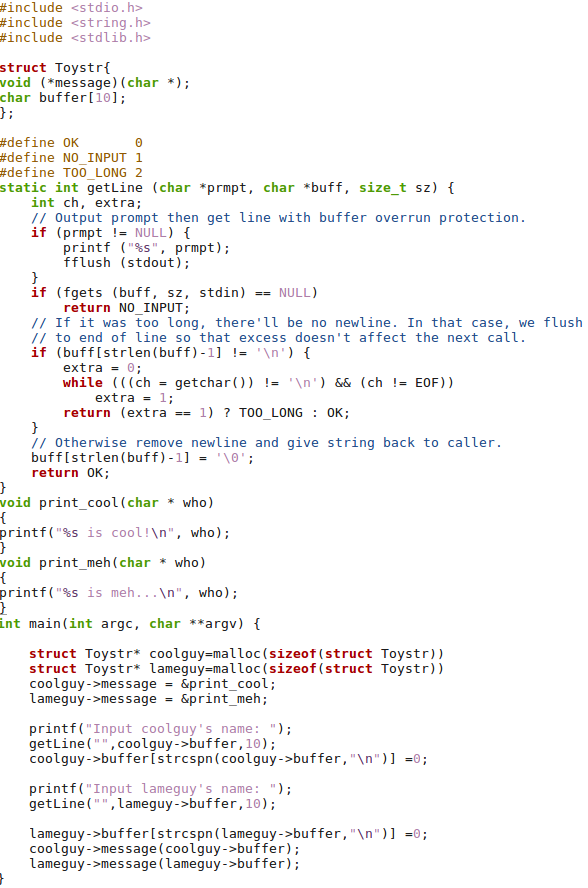
Vstup pre meno lameguy je prepísaný prvým vstupom a vnútorná správa lameguy je porušená a práva pre coolguy nieje vôbec vypísaná.

Chyba aplikácie je vo fikcií fgets, pri ktorej dôjde v heap overflow a tým pádom premenné v pamäti sú porušené.

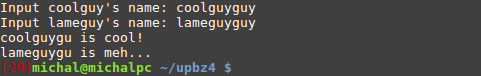
**Zabezpečenie kódu**

Kvôli zabezpečeniu vstupu sme funkciu fgets nahradili nami vytvorenou novou funkciu getLine. Funkcia na základe zadanej dĺžky buffra odstrihne ďalší nežiaduci obsah získaný zo vstupu.

Ukážka kódu aplikácie po zabezpečení vstupov



Následne aj po zadaní dlhších vstupov ako má aplikácia vyhradené pre premenné v pamäti aplikácia pracuje správne a predišlo sa heap overflow



# Záver

Je potrebné dodržiavať určité bezpečnostné pravidla ak chceme ochrániť našu aplikáciu pre únikom informácií. Preto aj pri tvorbe našej aplikácie sme sa snažili dodržiavať zadané bezpečnostné pravidlá. Funkcie, ktoré bolo potrebné doprogramovať sme doprogramovali aby sme našu aplikáciu ochránili pred ľahkým zlomením hesla účtov používateľov hackermi.