סדנת דבאג

פתרונות

עקרונות מתקדמים בתכנות

MyPointExactly

חלק 1

כדי להדפיס את הערך של p1 ו- p2 צריך להוסיף אותם לפקודת ההדפסה.

חלק 2

אנו מבצעים ההעתקה בגודל לא נכון של שני בתים. הגודל הנכון הוא 2*sizeof(int).

חלק 3

השמה יחד עם ההצהרה על משתנה תגרום לקריאה של אופרטור ההעתקה ולא אופרטור השמה.

אופרטור ההעתקה ביצע העתקה לא נכונה - shallow copy ולא deep copy. כלומר, הועתק רק המצביע לזיכרון וכך נוצר מצב ששני המופעים (instances) הצביעו לאותו זיכרון.

חלק 4

אופרטור ההעתקה במחלקה Point לא ממומש נכון.

לא קיימת בדיקה להשמה עצמית (p2=p2) ולכן מחיקה של coord_ תגרום להעתקה של זיכרון משוחרר.

שימו לב כי יש אפשרות לממש אופרטור העתקה גם ללא בדיקה של השמה עצמית בטכניקה שנקראת copy-and-swap.

קראו עוד על כך כאן:

https://stackoverflow.com/questions/3279543/what-is-the-copy-and-swap-idiom

LoopAgain

המשתנה size הוא מטיפוס unsigned. כאשר ערכו מגיע לאפס ומבצעים חיסור מתרחש integer underflow – מאחר וטיפוס unsigned לא יכול להיות שלילי ערכו הופך להיות הערך הגדול ביותר האפשרי. לרוב יש מקרו שמגדיר אותו (UINT_MAX). ערך זה הוא תלוי סביבה אבל בדר"כ הוא בעל 4 בתים וערכו 4294967295.

Shapes

מדוע שטח המשולש הוא אפס?

לפי מנגנון הפולימורפיזם הפונקציה get_area של מחלקת הבן צריכה להקרא, הרי היא פונקציה וירטואלית. אך אם תשימו לב היטב, החתימה שלהן שונה! כך שבפועל הפונקציה Triangle::get_area מסתירה את Shape::get_area.

SafeAndSound

חלק 1

גודל המחרוזת אליה מעתיקים לא גדול מספיק כדי להכיל את מחרוזת המקור (לא נלקח בחשבון תו 0\ שמסיים את המחרוזת). לכן בזמן ההעתקה ישנה דריסת זיכרון שאסור לנו לגשת אליו.

חלק 2

הבאג נובע מטעות בחישוב אורך מחרוזת המקור. באותו אופן כמו בסעיף 1, לא לוקחים בחשבון שמחרוזת מסתיימת בתו 0\ ומעתיקים פחות תווים. הדפסת המחרוזת dest מדפיסה כל דבר שיש במחסנית עד לתו 0\ ולכן גם הסיסמה הסודית מודפסת למסך.

Password

הדגל incorrect באובייקט Password נמצא מיד אחרי value בזיכרון (זכרו, הזיכרון הוא רציף).

הקלדה של 16 בתים בדיוק (לא משנה ערכם), מאפשרת לנו לבצע העתקה שחורגת מגבול value ודורסת את incorrect בזיכרון, כלומר משנה את ערכו ל- true.

שימו לב שהבדיקה היא למעשה שלילה כפולה (ומאוד מבלבלת).

הערה: הקלדה של יותר מ- 16 בתים תגרום לקריסה של התוכנית. למה?

ImagesAndWords

התמונה הזדונית מגדירה את ערכי הגודל של התמונה (width ו- height) לערכים מאוד ספציפיים.

פונקציית הקריאה של התמונה קוראת את הערכים הללו מהקובץ ומבצעת כפל שלהם.

מפני שתוצאת הכפל מושמת למשתנה מטיפוס unit16_t מתבצעת חריגה (integer overflow) וערך התוצאה הופך להיות מספר מאוד קטן.

מיד לאחר מכן מקצים זיכרון בגודל זה ומבצעים העתקה של מידע רב (תוכן התמונה מתוך הקובץ) לזיכרון בגודל שקטן ממנו. כך שבפועל מתבצעת דריסה של זיכרון שאסור לנו לגשת אליו.