שם: שמואל ביסברג

ממ"ן 12 שאלה 2

1.

2. החולשה נמצאת בפונקציה `handle\_escape` הגורמת לגלישת חוצץ מה שמאפשר לשנות את הערך של הvptr ולגשת לכתובת הפונקציה `unreachable `

נקבל את הסוד בעזרת הקלט:

./bin/mmn12-q2 -e "\\x$(python3 -c 'print("A"\*16 + "\x52\x17\x00\x00")')"

כלומר, אחרי הbuffer של 16 בתים, אנחנו דורסים את *הvptr של h עם הכתובת של unreachable*

3.

נשנה את `handle\_escape` ככה שלא יהיה אפשר לקרוא יותר מ16 בתים ונבטיח שיש ‘\0’ בסוף הbuffer.

ראה קוד בfixed-mmn12-q2.cpp

4.

ניתוח החולשה:

החולשה נמצאת בפונקציה handle\_escape. לולאת while פשוטה מעתיקה את קלט המשתמש לתוך החוצץ ללא כל בדיקת גבולות. כאשר המשתמש מספק קלט ארוך מ15 תווים, ההעתקה ממשיכה מעבר לגבולות ה-buffer ודורסת נתונים סמוכים על המחסנית. הנתון הראשון שנדרס הוא הvptrשל האובייקט h. ה-vptr הוא מצביע המשמש במחלקות פולימורפיות (בעלות פונקציות וירטואליות) כדי לאתר בזמן ריצה את הפונקציה הנכונה להפעלה מתוך טבלת הפונקציות הווירטואליות (vtable). דריסת מצביע זה פותחת פתח להתקפות מתקדמות..

ההתקפה: vtable Hijacking

1. בניית payload באורך 20 תווים עם כתובת הפונקציה.
2. בניית vtable מזויפת במיקום 4 buffer + עם מצביע לפונקציה unreachable
3. כאשר הקוד קורא לפונקציה l.h.helper() אזי ניקראת הפונקציה unreachable

הגנה ותיקון:

ניתן להגן מפני תקיפה זו ב2 דרכים:

1. נפעיל את מנגנון הASLR כאשא אנחנו מריצים את התוכנית ובכך יתקשה התוקף לגשת לפונקציה הרצויה.
2. יש להחליף את לולאת ההעתקה הפגיעה בפונקציה strncpy, המגבילה את מספר הבתים המועתקים לגודל החוצץ. חיוני לוודא גם סיום של המחרוזת בתו \0 (null terminator).