דו״ח מעבדה-SEED Labs – Return-to-libc Attack Lab

שם : קוסטיה קזקוב

ת.ז. : 321827834

1. מכיוון שהמחסנית גדלה הפוך, מתחילים מכתובות גבוהות לקטנות, ומכיוון שאנו צריכים לחרוג מ12 על מנת לדרוס את מה שבמחסנית לפני(המערך שהוקצה לפני כן), אנו צריכים להכניס את הכתובת של system לתוך המקום במחסנית שנמצא המצביע לכתובת חזרה של הפונקציה bof(מחישוב לוגי, 12 המערך המוקצה, 4 compiler padding, 4 הEBP הישן, אם נסכום את כל זה נקבל 20, כלומר הכתובת חזרה נמצאת 24 בתים לפני סוף ההקצאה, לפני שהבנתי שיש compiler padding השתמשתי בניסוי וטעייה עם הכתובת 20 וקיבלתי שגיאה), את הארגומנט של הפקודה system \bin\sh למקום 32(חזרה+8) והיציאה(שלא חשובה כל כך אבל על מנת לשמור על חשאיות) למקום 28(חזרה+4) לפי הסדר במחסנית.
נ.ב. המיקום שקיבלתי מהקוד הביא לי כתובת קדימה ולא כללה את b\ לכן הקטנתי בשתי כתובות אחורה את הכתובת של הפקודה(רואים בצילום מסך).
2. ההתקפה לא הצליחה מכיוון שכאשר משנים(לגודל שונה) את השם של הקובץ, הכתובות זזות בהתאם, אם אנו נגדיל את השם של הקובץ, הכתובות יזוזו למטה מכיוון שהשם של הקובץ נשמר במחסנית(במשתנה ARGV) וכך כל הפרמטרים בעצם זזים למטה(כל תו במקום אחר בזיכרון חדש).
3. הפריצה לא תעבוד, הבעיה היא שכאשר הגנת אקראיות מרחב הכתובות דלוקה, אין סדר על הכתובות ומיקומם נמצא במקום שונה באקראיות בלי שום סדר כרונולוגי. הגנה זו מקשה על הפריצה מכיוון שמכיוון שהכתובות נמצאות לא בסדר כרונולוגי(או סדר פשוט שניתן לחשב), גם עם נמלא את הזיכרון בקוד זדוני, לא בהכרך התכנית תחזור לאן שאנו רוצים(בדוגמא שלנו רצינו לגרוס את הכתובת חזרה, אם לא נדע היכן בזיכרון הכתובת נמצאת, הקוד הזדוני בעצם לא בהכרך יגרוס אותו).