

תכנות מערכות דפנסיבי – ממ"ן 12

מאת: יאיר חריט 207282955

1. א. החולשה הקיימת בקטע הקוד הנתון היא חולשת גלישה נומרית המתרחשת כאשר משווים בין `bound` ו-`credit`. מאחר והאחד מסוג `int` והשני מסוג `unsigned int` כאשר הביט הגבוה ביותר יהיה 1 יציג כל משתנה ערך שונה עבור אותם ביטים בזכרון, זאת משום שבמשתנים מסוג `int` מסמנים מספרים שליליים בעזרת הדלקת הביט הגבוה. על כן, כאשר ישוו בין `int` שלילי ל-`unsigned int` תתקבל תוצאה כי `credit >= bound`

על כן, ניתן לנצל חולשה זו ולתקוף את המערכת ע"י כך שנכנסים למינוס בבנק! לקוח אשר יהיה סכום שלילי בחשבונו יקבל גם הוא את המתנה (בנק מאוד נחמד)

ב. ניתן לתקן חולשה זו ע"י הגדרת המשתנה `bound` כ-`int` (אם `credit` יכול להיות שלילי ולכן אינו יכול להיות מטיפוס `unsigned int`).

כאשר משווים בין 2-`int` אנו נמנעים מהשוואות שגויות כמתואר בסעיף א', שכן,

שני המשתנים נשארים באותו "קנה מידה".

```
bool is_entitled_for_promotional_gift(int ID)
{
    int bound = 750;
    int credit = get_credit(ID);
    return (credit >= bound);
}
```

2. החולשה:

החולשה המתחבאת בקטע הקוד הנתון הינה חולשת גלישת חוצץ (Buffer Overflow):

בפונקציה `handle_escape`, כאשר מעתיקים את הפרמטר `str` ל-`l.buffer` אין דבר המונע מאיתנו לדרוס את הזכרון של `h` (Handler), כלומר לשנות את הטבלה הוירטואליות של האובייקט. על כן, כאשר נזין מחרוזת ארוכה מספיק נדרוס את ראש הטבלה הוירטואלית כך שתצביע על פונקציות אחרות מכפי שהגדיר המשתמש. כאשר ינסה המשתמש לעשות שימוש בפונקציות שהגדיר, יקראו פונקציות אלו **במקום** פונ' המשתמש ההתחלית.

ההתקפה:

תחילה, על מנת לאפשר שימוש ב'\' עלינו להגדיר משתנה סביבה בשם `"ECHOUTIL_OPT_ON"` ולתת לו ערך כלשהו (אפילו 0) על מנת שהמשתנה `env` לא יהיה NULL וכך `allow_options = True`. שנית, על מנת להגיע לקטע הקוד הפגיע עלינו להכניס את הארגומנטים הבאים לתכנית הראשית:

- ארגומנט 1: הדגל `'-e'`

```
case 'e':  
do_escape = true;
```

כך ש-`do_escape` יהיה True.

- ארגומנט שני:

`*123456789012345x`

```
154 if (do_escape && s[0] == '\\')  
155     handle_escape(s);
```

- יתחיל ב'\' כך שהתכנית תכנס לתנאי בשורה 154
- התו השני יהיה 'x', על מנת שהתכנית תכנס לפונ' `l.h.interpret`
- כעת נכניס 15 תווים נוספים על מנת למלא את ה-`buffer`
- לאחרים נכניס את ערך ה-`ascii` של הבית הנמוך של כתובת הטבלה החדשה (אין צורך לשנות את שאר הבתים במקרה שלנו):
 - o באמצעות ה-`debugger` נמצא את הכתובת הקיימת ונשמור את הבית הנמוך ביותר (lsb), במקרה זה `LSB = 0x10`
 - o מאחר וגודל כל כתובת הינו 4 בתים נחסר מה-`LSB` שמצאנו 4 - כך שכאשר ננסה לגשת למקום 1 (helper) בטבלה נגיע למקום 0 (unreachable). במקרה זה הערך החדש יהיה `0x0C`.
 - o מאחר והקלט נעשה כמחרוזת נמיר את התוצאה לערכי `ascii` ונוסיף אותה לסוף הארגומנט.

כעת כאשר תנסה המערכת לגשת לפונק' `l.h.helper` על מנת לעבד את הארגומנט השני תקרא הפונק' `l.h.unreachable`.

ההגנה:

ניתן למנוע התקפה מהסוג המתואר לעיל ע"י עטיפת הלולאה המעתיקה בפונקציית `handle_escape` בתנאי הבדוק את אורך המחרוזת.

```
71 if (strlen(s) < 16)  
72     while (*s)  
73         *p++ = *s++;
```

DefMMN12 Property Pages

Configuration: Active(Debug) Platform: Active(Win32) Configuration Manager...

Configuration Properties

General

Advanced

Debugging

VC++ Directories

C/C++

Linker

Manifest Tool

XML Document Generator

Browse Information

Build Events

Custom Build Step

Code Analysis

Debugger to launch:

Local Windows Debugger

Command	\$(TargetPath)
Command Arguments	-e \x123456789012345
Working Directory	\$(ProjectDir)
Attach	No
Debugger Type	Auto
Environment	ECHOUTIL_OPT_ON=0
Merge Environment	Yes
SQL Debugging	No
Amp Default Accelerator	WARP software accelerator

Command

The debug command to execute.

OK

Cancel

Apply