טוב, אז רוצים למצא מה הערך של המשתנה kaput. קודם כל, נראה איפה הוא מופיע בstrtab:

```
000002B0  3C 00 00 00  94 90 04 08  00 00 00 00  10 00 12 00  ....
000002C0  43 00 00 00  94 90 04 08  73 61 64 00  4B 61 70 75  .e3.s.Assad.Kapu
000002D0  00 65 33 2E  73 00 41 73  54 75 72 6B  65 79 00 65  t.Syria.Turkey.e
000002E0  74 00 53 79  72 69 61 00  00 5F 73 74  61 72 74 00  xit.Iran._start.
000002F0  78 69 74 00  49 72 61 6E  61 72 74 00  5F 65 64 61  __bss_start._eda
00000300  5F 5F 62 73  73 5F 73 74                           ta._end.
00000310  74 61 00 5F  65 6E 64 00
```

אפשר לראות כאן שהוא נמצא 12 בייטים מתחילת הstrtab, כלומר האופסט שלו הוא 0xC.
אחלה, עכשיו אנחנו צריך ללכת לטבלת הסימבולים, ולחפש את הסימבול הזה. נזכר בשדות של כל מבנה ב symtab:

Figure 1-16: Symbol Table Entry

```
typedef struct {
        Elf32_Word      st_name;
        Elf32_Addr      st_value;
        Elf32_Word      st_size;
        unsigned char   st_info;
        unsigned char   st_other;
        Elf32_Half      st_shndx;
} Elf32_Sym;
```

השדה הראשון הוא שם הסימבול,כלומר איפה השם שלו נמצא מתחילת הstrtab. עכשיו נעבור על כל הרשומות בsymtab ונחפש אחד כזה שבשדה הראשון שלו כתוב 0xC. שימו לב שכל רשומה תופסת 16 בתים, שזה בדיוק שורה בקובץ.

```
000001B0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001C0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00
000001E0  00 00 00 00  54 80 04 08  00 00 00 00  03 00 01 00
000001F0  00 00 00 00  54 80 04 08  00 00 00 00  03 00 02 00
00000200  00 00 00 00  65 80 04 08  00 00 00 00  03 00 03 00
00000210  00 00 00 00  90 80 04 08  00 00 00 00  04 00 F1 FF
00000220  01 00 00 00  00 00 00 00  00 00 00 00  00 00 01 00
00000230  06 00 00 00  54 80 04 08  00 00 00 00  00 00 01 00
00000240  0C 00 00 00  55 80 04 08  00 00 00 00  00 00 01 00
```

(תתעלמו והקשקושים). אז בשורה של 240 מצאנו את הרשומה עם השם kaput. שיגעון, מה עכשיו?
עכשיו מסתכלים על השדה st_value, הוא אומר לנו בעצם איפה המשתנה מקבל ערך. הכתובת שכתובה שם היא 0x08048055. זה בעצם המקום בזכרון שבו כתוב הערך של המשתנה, אבל אנחנו בקובץ, זה לא מה שאנחנו צריכים. או ש..?
בעצם הקובץ מתחיל מ0x08048000, תמיד! הסיבה לזה לא ברורה, זה סוג של מוסכמה של לינקרים.. בשביל לוודא את זה אפשר לחקור סימבולים אחרים בקובץ (ואת הentry point) ולראות שהאופסט של כולם נשען על הכתובת הזאת, אבל שוב.. זה תמיד ככה.
אוקי, אז עכשיו הולכים לאופסט 55 מתחילת הקובץ:

```
00000040  00 80 04 08  94 00 00 00  94 00 00 00  61 79 72 0A  ....
00000050  00 10 00 00  00 FF FF FF  FF 51 75 73  BB 01 00 00  ....
```

ואפשר לראות שמה שמופיע שם זה 0xFFFFFFFF שזה בעצם מינוס אחד. צריך גם להסתמך כאן על התשובות, כי אני חושב שאי אפשר לדעת מה הגודל של המשתנה kaput, פשוט רואים שזאת התשובה היחידה שמתאימה.

יש פה קצת קסם, אבל מקווה שברור

איך למצוא את ה-String Tables

יש שני סוגים של String Tables.
1) Section Header String Table - מחזיק שמות של סקשנים
2) Symbol String Table - מחזיק שמות של סימבולים (למשל, פונקציות)

1) בשביל למצוא את ה-Section Header String Table:
ב-Header יש שדה e_shstrndx. השדה הזה מחזיק את האינדקס של ה-section header שהוא ה string table. מוצאים את ה-section header שהאינדקס שלו הוא e_shstrndx, ובשדה offset שלו מקבלים את ה-offset מתחילת הקובץ, שם מתחיל ה-String Table.

אם נסמן SH בתור ה-offset של ה- section header הראשון (זה שמופיע ב-Header בשדה e_shoff) אז אפשר לחשוב על זה גם ככה: Section Header String Table Offset = SH[e_shstrndx]->offset

2) בשביל למצוא את ה-Symbol String Table:

א. מוצאים את ה- Section Header ששדה ה-type שלו הוא 2.
ב. 20 בייטים אחרי ה-type (באופסט 24 דצימלי מתחילת הsection) נמצא את sh_link ונקבל אינדקס i
ג. נלך ל-section header שהאינדקס שלו הוא i (חישוב בהערה למטה) ושם נלך לשדה ה-offset.
ד. ה-offset שקיבלנו הוא ה-offset מתחילת הקובץ בו נמצא ה-symbol string table

זה נראה כמו משהו כזה:

section index	sh_type	sh_link	sh_offset
0	...	..	...
1	2	7	0x....
...	...	...	
7	3	0	0x200

בדוגמה שמעל ה-offset של ה-symbol string table הוא 200x0.

* לא הכי נוח נוח אבל ניתן לחשוב על זה גם ככה:
(אם נסמן SYMTAB להיות האינדקס של ה-section header שה-type שלו הוא 2)

Symbol String Table Offset = SH[SH[SYMTAB]->sh_link]->offset

<u>הערה</u>: בהינתן אינדקס i של section header נמצא את הoffset שלו מתחילת הקובץ:
header->e_shoff + i * header->e_shentsize
(בד"כ במקרה שלנו e_shentsize הוא בגודל h28)