* 1. ברגע הכנסת esp לתוך ebp, אנחנו בעצם שומרים מצביע לתחילת מסגרת הפונקציה הנוכחית: כלומר, מתחת לכתובת עליה מצביע ה-ebp החדש, יישמרו רגיסטרים שהפונקציה צריכה לזכור לאחזר אחר כך, קריאות פנימיות לפונקציות וכו. שמירה על ערך זה חשובה כדי לאפשר לפונקציה לגשת לפרמטרים שלה שמועברים אליה על המחסנית ע"י הפונקציה הקוראת – לכן, "מעל" ebp החדש יהיו הפרמטרים.
  2. ניתן להשתמש ישירות ב-esp בכל פעם שנרצה לגשת לפרמטר / משתנה.
  3. החיסרון המרכזי הוא שנצטרך לעקוב אחרי גדילת המחסנית במהלך ריצת השגרה – אם יש לנו פרמטר ב-8(%esp) ואז אנחנו דוחפים משהו למחסנית, בפעם הבאה שנרצה לגשת לפרמטר נצטרך לכתוב מספר גדול מ-8. היתרון בגישה זו היא שאין צורך לשמור את ebp, לשים בו את esp ולאחזר אותו אח"כ – גם 3 שורות קוד "מיותרות" וגם רגיסטר תפוס.
  4. למנוע מתכנה זדונית / מתכנת מטומטם לגרום לנזק בלתי הפיך / קשה מאוד לחומרה, ובאופן כללי למנוע ממשתמש שליטה מלאה בחומרה. אם כל הפעולות היו מותרות גם למשתמש, הוא היה יכול למשל לבצע פסיקה משל עצמו ולהריץ קוד ללא הפרעה מצד מערכת ההפעלה (כלומר: משתמש יקבל זמן מעבד בלתי מוגבל וימנע החלפת הקשר).
  5. ע"י קריאה לפונקציות מערכת. מערכת ההפעלה מספקת שירותים למשתמש בדיוק למקרים אלו: אם המשתמש רוצה לבצע פעולה "אסורה", הוא יבקש ממערכת ההפעלה לבצע אותה במקומו. כך, מערכת ההפעלה דואגת למנוע שימוש לא נכון / מזיק בחומרה, ומשתמש יכול לקבל שירותי חומרה.
  6. זה קורה בכניסה לקוד גרעין, ע"מ שנדע לאן לחזור עם חזרתנו לקוד משתמש. לקוד שרץ ב-CPL3 יש מחסנית נפרדת מקוד שרץ בגרעין, ולכן כאשר אנו עוברים מקוד משתמש לקוד גרעין, ע"מ לא לאבד את מיקום מחסנית המשתמש אנו שומרים את ss:esp על מחסנית התהליך בגרעין.
  7. TSS הוא אזור בזיכרון הגרעין (יש אזור TSS לכל מעבד) שבתוכו יש שדה המצביע למחסנית של הגרעין. בעת קריאת מערכת, משתמשים בשדה זה ע"מ לדעת איפה מחסנית התהליך בגרעין, שבתוכה המערכת צריכה לשמור את ה-ss:esp של התהליך ב-UserMode.
  8. עדכון ה-TSS מתבצע בעת החלפת הקשר: בזמן החלפת הקשר התהליך שרץ על המעבד משתנה, ולכן כתובת המחסנית שלו בגרעין משתנה. זה מה ששמור ב-TSS, לכן עליו להתעדכן.
  9. הקוד הזה יוצר כמות גדולה מאוד של זומבים (דבר שמסוכן מאוד במערכת עם כמות מוגבלת של כדורים, רובים וציוד הישרדות אחר): בכל איטרציה, תהליך הבן מיד יצא מהלולאה (כי הערך המוחזר מ-fork() יהיה 0) ולכן יעשה exit (כי הוא יצא מה-main). כיוון שהאבא לא מבצע wait(), הבן יישאר במצב זומבי. לעומת זאת, האב יראה ערך חיובי חוזר מ-fork() ולכן הוא ימשיך בלולאה.
  10. קודם כל, התהליך היחיד שיכול ליצור בנים הוא התהליך המקורי שנכנס לקוד (כי כל בן שלו מת לפני שהוא מגיע לפקודת fork()). לעומת זאת, התהליך המקורי יגיע למגבלה של מערכת ההפעלה (חסום ע"י 32K פיסית, וע"י הערך RLIMIT\_NPROC בפועל).