ממן 12

רז בן צבי – 325843944

3. כדי לתקן את הקוד צריך להחליף את קטע הקוד משורות 81-85 שמעתיק את str לתוך l.buffer בשורה:

strncpy(l.buffer, str + 1, sizeof(l.buffer) - 1);

ככה נעתיק רק מחרוזות בגודל שמתאים לבאפר ולא נדרוס את h.

4. להלן מסמך המחקר:

החולשה: buffer overflow -

חולשה מתרחשת כאשר תוכנית מנסה לכתוב נתונים לתוך באפר גדול יותר מהקיבולת שהוקצתה לו. זה גורם לנתונים לזלוג מחוץ לגבולות הבאפר ולדרוס אזורי זיכרון סמוכים, דבר שיכול לגרום להתנהגויות לא צפויות של הקוד.

בקוד של התרגיל אפשר לראות את החולשה בפונקציה handle\_escape שמקבל מחרוזת ומעתיקה את התוכן שלה לתוך באפר שנמצא בתוך struct ביחד עם עוד משתנה מסוג Handler.  
ההעתקה של המחרוזת מתבצעת ללא בדיקה של גודל המחרוזת ואז למרות שגודל הבאפר הוא 16, אפשר להעתיק מחרוזת בגודל גדול מ16 בתים כדי לדרוס את המשתנה השני בstruct ואפילו יותר!

ההתקפה: לאחר שמצאנו את החולשה, נבין איך משתמשים בה כדי להגיע לפונקציה unreachable.  
הפונקציה unreachable נמצאת במחלקה Handler עם עוד פונקציה בשם helper, שניהן פונקציות וירטואליות.  
זה ששניהין וירטואליות אומר שלאובייקט מסוג Handler יהיה מצביע לvirtual table שמחזיק את הכתובות של שני הפונקציות האלה, כל אחת היא 4 בתים ובעת קריאה לאחת הפונקציות האלה מה שיקרה זה שלפי הכתובת של השולחן הזה, יקראו לפונקציה שבאופסט של האינדקס \* 4 ממנו בצורה רלטיבית.  
הרצון שלי הוא לשנות את השולחן הוירטואלי שישמש אותי לצרכים שלי.  
מה שאני עשיתי זה למצוא את הכתובת בזכרון של השולחן הוירטואלי, ולדרוס את הכתובת האמיתית שלו בעזרת החולשה שמצאנו להיות הכתובת האמיתית פחות 4 בתים, דבר זה גורם לכל קריאה לפונקציה של האובייקט הנוכחי ששינינו לו את הכתובת של השולחן הוירטואלי לקרוא לפונקציה ש"אחת לפניו".  
לפי הסדר עכשיו נקרא כשבצורה טבעית נקרא לhelper, זה ייקרא לunreachable במקום.  
בגלל שלא קוראים לפונקציה unreachable בצורה טבעית אנחנו לא צריכים לחשוש שנשנה את הקריאה אליה למשהו לא חוקי ושזה יקריס את התכנית.

ההגנה: כפי שהוסבר בסעיף 3, כדי להגן ממתקפות כאלה צריכים תמיד לוודא את הקלט מהמשתמש עם פונקציות כמו strncpy שבודקות את הגודל של מחרוזת שאנחנו מעתיקים ובככה מונעת buffer overflow