Operating Systems Security Vulnerabilities and Cyber Defenses

תרגיל 2

חלק א:

ניצול הפרצה:

מצורף קוד המקור המלא למימוש הפרצה בקובץ exploit.cpp והקוד המקומפל ב executables\exploit.exe. ה exploit מנתר שינויים בתיקיה "C:\Documents and Settings\Public" ולאחר שנמצא קובץ חדש מנסה להחליף אותו בקובץ אחר.

הערות לבדיקה: יש לוודא כי הקבצים toExec.tmp ו- 1.tmp נמצאים באותו תיקייה שממנה אתה מריץ את הexploit.

תיאור ותיקון הפרצה:

בחלק זה, פרצת האבטחה נובעת מכך שהservice מייצר קובץ עם security descriptor דיפולטיבי שמאפשר כתיבה על הקובץ לכל משתמשי המחשב. כותב אליו קוד ולאחר מכן מריץ child process מקובץ זה מבלי לבדוק שקובץ זה לא שונה. בכך נוצר "חלון זמן" בין הרגע בו ה service מסיים לכתוב לקובץ וסוגר אותו לרגע בו הוא יוצר את התהליך הרץ מקובץ זה בו כל משתמש (גם כזה עם הרשאות נמוכות) יכול **לנסות** להחליף את הקובץ או לשנות את תוכנו (race condition). כיוון שהservice רץ בהרשאות גבוהות גם תהליך הבן רץ בהרשאות גבוהות וכעת (אם הצלחנו להחליף את תוכן הקובץ) תהליך זה יריץ קוד שלנו בהרשאות גבוהות (privilege escalation).

ישנן מספר דרכים לפתור בעיה זו. בפתרון בו אנו מימשנו אני מגדירים את הרשאות הקובץ בעת יצירתו לתת גישה מלאה ובעלות על הקובץ אך ורק למשתמש המריץ את ה service באותו רגע. לשם כך יצרנו security descriptor עם DACL בעל ACE יחיד של ALLOW ALL רק לSID של המשתמש הנוכחי בשדה הOWNER הגדרנו את SID זה והוספנו את הFLAG המגדיר מניעת ירושת הרשאות. בנוסף היה צרוך למחוק את הקובץ שנוצר בעת הקריאה ל-GetTempFileName כיוון שאחרת הקובץ החדש היה מקבל את איחוד ההרשאות החדשות ואלו של הקובץ הקודם. מצורף קובץ קוד בשם fix.cpp ובו ממוש התיקון כולל מציאת הSID הנוכחי.

פתרון נוסף הוא להריץ את תהליך הבן בהראות נמוכות (אם אין צורך שירוץ בהרשאות גבוהות).

הערות לבדיקה: הקוד המצורף עבור התיקון מתקמפל ועובד. אך יש צורך להוסיף resource שממנו ה-service המתוקן יכתוב לקובץ (הקוד יעבוד גם ללא הוספת ה-resource אך תהליך הבן יכשל כיוון שהוא מנסה לרוץ מקובץ ריק).

חלק ב:

ניצול הפרצה:

מצורף קוד המקור המלא למימוש הפרצה בתיקיית MaliciousClient.cpp והקוד המקומפל ב \executables MaliciousClient.exe. הקליאנט מתחבר אל השרת ולאחר מכן שולח לו payload המורכת מגודל (בבתים) של הקוד שנשלח, לאחריו הchecksum של קוד זה ולבסוף הקוד עצמו (קוד PIC ללא null bytes מקומפל לשפת מכונה). בעת הרצת הshellcode בשרת הshellcode מוצא את הbase address של kernel32.dll (שנטען לכל תוכנית) דרך הPE של התוכנית שנטענת לכתובת קבועה([0x30]:fs יחסית למרחב הזיכרון של התוכנית). משם הוא מוצא את הexport table של kernel32.dll ובו את הכתובת שאליה נטענה הפונקציה GetProcAddress. לאחר מכן בעזרת GetProcAddress הshellcode מוצא את הפונקציות CloseHandle, WriteFile, CreateFileA שנטענו עם kernel32.dll איתם הshellcode יוצר קובץ בשם Hello.txt בתיקיית C:\Windows, כותב לתוכו "Hello World!", סוגר את הhandle לקובץ ולבסוף משחזר את הרגיסטרים למצבם הקודם ומבצע return (השרת ימשיך לרוץ לאחר סיום ביצוע הshellcode).

תיאור ותיקון הפרצה:

הבעיה העיקרית פה הינה ששרת העדכונים לא מבצע כלל בדיקה מקדימה למקור העדכון ולתוכנו (פרט אולי לתקינות הקלט שלא נפגע במהל השליחה – checksum) תרם הרצתו בהרשאות גבוהות.

ישנן מספר שיטות למנוע בעיה זו. ניתן ואף רצוי לשלב ביניהן.

- ניתן להגדיר את השרת כך שיקבל בקשות חיבור אך ורק מכתובות ספציפיות של שרת עדכונים אמין וידוע מראש. הבעיה בשיטה זה היא שקל מאוד לזייף כתובת ip ולכן ניתן בקלות להתחזות לשרת האמין אם אנו מגלים את זהותו. החיסרון בשיטה זו היא שמקבל העדכון צריך לדעת מראש מי הוא השולח שממנו הוא אמור לקבל את העדכון. בנוסף אם בדרך כלשהי התוקף משיג את המפתח הפרטי הוא יכול להתחזות לשרת השולח.
- אפשר להשתמש במנגנון חתימה דיגיטלית כגון DSA המשתמש בהצפנה אסימטרית כדי לוודא את זהות השולח.
- במקרים מסוימים לקוד הצפוי להתקבל יש מבנה ידוע מראש. במקרים כאלו לפני הרצת הקוד ניתן לבדוק האם הקוד שהתקבל עונה למבנה הצפוי (בדיקת משתנים, כתובות, פונקציות שבהם הוא משתמש, קבצים שאליהם הוא ניגש ו flow כללי). הבעיה בשיטה זו היא שראשית לא תמיד ניתן להגביל את צורת הקוד וכיוון שעדיין חייב להיות שוני כלשהו בקוד המתקבל תמיד אפשר ליצור קוד שעונה לדרישות ועדיין מבצע פעולות זדוניות (בעיה בשימוש black list במקום white list). בנוסף מדרשת הרבה עבודה כדי לפענח את הקוד שהתקבל.
- לעיתים אין צורך להריץ את הקוד המתקבל בהרשאות גבוהות. במקרה כזה ניתן לשמור את הקוד שהתקבל לקובץ ולהריץ תהליך בן בהרשאות נמוכות עם קוד זה. אך פתרון זה לא אפשרי כאשר אנו זקוקים להרשאות גבוהות לצורך ביצוע העדכון.