Operating Systems Security Vulnerabilities and Cyber Defenses

תרגיל 2

תאריך הגשה – 25.01.2017

התרגיל ניתן להגשה ביחידים או בזוגות

בדף זה ישנם 2 תרגילים. יש להגיש את התרגילים יחד בקובץ ZIP.

חלק א - תרגיל פירצת אבטחה

רקע

בעיה מאוד נפוצה בקוד היום הוא קוד שלא מניח שהוא מתמודד מול אויב ומבצע פעולות שונות בהנחה שהכל תקין.

דוגמה לחולשה כזאת היא חולשה של Time of Check vs Time of Use. נניח שמערכת ההפעלה מקבלת שני מצביעים לנתיבי קבצים (source and dest) שנמצאים ב-user mode וצריכה להעתיק אחד לשני.

מערכת ההפעלה לוקחת את ה-destination ובודקת שאכן יש לתוכנה הרשאות לכתוב לשם וכותבת. כלומר מתבצע הקוד הבא

1) לבדוק האם יש לתוכנה הרשאות להעתיק ל-Dest
2) להעתיק מ-source ל-Dest

אך מה קורה אם בזמן בין 1 ל-2, התוכנה מחליפה את המחרוזת לנתיב שאין לה הרשאות? מערכת ההפעלה תיפול בפח ותאפשר גישה.

בתרגיל הזה, יש טעות דומה (אך שונה) שעליכם למצוא, להשמיש ולחסום.

דרישות תרגיל

קיימת תוכנה בשם Vulnerable.exe עם קוד מקור (חלקי) מצורף בשם vulnerable.cpp (הקוד לא בהכרח תתקמפל).

אתם צריכים להריץ את התוכנה כ-System (הוראות בהמשך), למצוא בה את פירצת האבטחה, לממש קוד שמנצל אותה (Exploit) ולספק פתרון להתמודדות עם פירצת האבטחה.

הדרכה

1. כדי לדמות את התוכנה רצה בהרשאות גבוהות, נשתמש בכלי בשם psexec (חלק מחבילת sysinternals)
   בעזרתו, נריץ את התוכנה בהרשאות גבוהות, אך interactive כדי שנוכל לראות את הפלט
   psexec -s -i [full executable path]
2. תמצאו חולשה בקוד ☺
3. תכתבו תוכנה שתעזר בחולשה כדי להריץ קוד בהרשאות גבוהות

יש כל מיני כלים שיעזרו לכם

- Process Explorer - https://technet.microsoft.com/en-us/sysinternals/processexplorer.aspx
  יעזור לכם להבין מה מצב הריצה של התוכנות שונות
- MSDN – כמובן, לתיעוד כל פונקציה אפשרית
  - רמז: FindFirstChangeNotification

הגשה

עליכם להגיש 3 קבצים

- קוד מקור של תוכנה (לא משנה באיזה שפה) שאם נריץ אותה היא תצליח להריץ קוד בהרשאות גבוהות
- קובץ שמסביר מהי פירצת האבטחה
- קובץ קוד מקור מתוקן **או** הסבר מפורט כיצד הייתם מונעים את פירצת האבטחה

הערות לבדיקה

בתרגיל זה, הקובץ ייבדק על מערכת הפעלה Windows 7 32-bit אבל ניתן להריץ את הקוד (לצורך פתרון) על כל מערכת הפעלה החל מ-Windows XP.

חלק ב – ניצול מנגנון

רקע

מנגנוני עדכוני תוכנה זה דבר נפוץ ומבורך. זה מסיר מהמשתמש את האחריות של לזכור שצריך לעדכן תוכנה וכל התהליך הכלול בדבר. אך תוכנות רבות ממשות עדכונים אוטומטים בצורה פגיעה ביותר.

בעיות נפוצות הם מחסור בבדיקת חתימות של עדכונים, אי שימוש בתקשורת מוצפנת וגם סתם בעיות מימוש בתהליך העדכון שמאפשר לתוקף לנצל את המנגנון כעוד ערוץ כניסה.

תרגיל

בתרגיל זה, קבלתם שרת (מנוון) שכל מה שעושה זה לקבל תקשורת מלקוח שונים, מוודא את הקלט ומריצה אותו.

שלב ראשון

תפקידכם לעבור את הבדיקות ולהריץ קוד זדוני אשר ינצל את הרשאות התוכנית לכתוב קובץ למקום כרצונכם.

כלומר, בהינתן שהתוכנה רצה כ-Windows Service בהרשאות גבוהות, תוכלו לכתוב קובץ ל-C:\Windows, מקום שכמובן לא נגיש למשתמש תמים מהרשת.

שלב שני

המטרה שלנו היא למצוא פתרון לבעיה בתוכנה שמאפשרת לכל תוקף כמוכם להריץ קוד בהרשאות התוכנה.

תחשבו, תתארו ותממשו הגנה בפני הבעיה שמצאתם.

הדרכה חלק א

לשרת מצורף הקוד מקור (המלא). אתם מוזמנים לקמפל בעצמכם.

עליכם לחקור את הקוד ולהבין כיצד לעבור את שלבי וידוא הקלט ואז לכתוב תוכנה שתתקוף ותריץ קוד מתוך השרת הפגיע.

בפועל עליכם לכתוב **shellcode** פשוט, כלומר קוד שלא תלוי במיקום ריצה שלו.

הקוד שלכם צריך להיות מסוגל ליצור קובץ עם תוכן לא ריק (אבל לבחירתכם) במקום מסוים. השיטה, ותוכן הקובץ, יכולים להיות כרצונכם.

רמז: השיטה הכי טריוויאלית, כמו CreateFile אינה הכי פשוטה למימוש.

עליכם ליצור תוכנה שמקבלת כפרמטר את כתובת ה-IP של השרת ותוקפת אותו. התוכנה יכולה להיות כתובה בכל שפה שבא לכם (את ה-shellcode כנראה תכתבו ב-x86 assembly...). **אין** צורך לוודא שהשרת ישאר למעלה בסוף אבל אסור לו לקרוס באופן שמקפיץ הודעת שגיאה.

כלים שימושיים:

- Masm - אסמבלר 32 ביט ל-windows
- מבוא לכתיבת shellcode

https://www.corelan.be/index.php/2010/02/25/exploit-writing-tutorial-part-9-/introduction-to-win32-shellcoding

אתם לא צריכים את כל המלל שכתוב שם! אבל זה עלול לעזור ☺

במקום IDA, ניתן להשתמש ב-objdump הכלול ברוב הפצות לינוקס

הדרכה חלק ב'

הצגת פתרון שמתומדדת עם הבעיה שמצאתם. מאחר ואין פתרון שהוא אולטימטיבי יש לציין גם מה המגבלות הפתרון שלכם.

הערות לבדיקה

בתרגיל זה, הקובץ ייבדק על מערכת הפעלה Windows 7 32-bit כאשר התוכנה רצה בהרשאות system. כלומר אתם מוגבלים לאותם מקומות כתיבה שלמע"ה יש הרשאות.

הגשה

יש להגיש תיקיה שמכילה את כל הקבצים הדרושים כדי ליצור את התוכנה שלכם וכל התלויות שלה כך שבהינתן IP לתקוף היא תידע לבצע תקיפה.

בתיקיה גם צריך להיות מצורף README המפרט כיצד יש להריץ את הקוד.

עבור החלק השני, יש להגיש קוד **או** הסבר מה עשיתם. ההסבר צריך לכלול מה הבעיה שאותה אתם מנסים לפתור, מה הפתרון עושה ומול מה זה לא מגן.