סדנה באבטחת מידע – תרגיל 4

המודול

למודול חמישה מרכיבים עיקריים: Hooks, Rules, Log, Connections, Hosts.

הפונקציה processPacket של Hooks רשומה כ-hook בנקודות PRE_ROUTING, POST_ROUTING כדי לנתח את כל הפקטות. פונקציה זו נעזרת ב-Rules, Connections ובעקיפין גם ב-Hosts כדי להחליט אם לקבל או לדחות את הפקטה. לאחר מכן היא נעזרת ב-Log כדי לכתוב את הפקטה ללוג.

Hooks

הפונקציה המרכזית של הקובץ הזה היא processPacket.

הפונקציה תחילה קוראת ל-buildPacketInfo כדי לשמור מידע על הפקטה במבנה מסוג packet_info_t. לאחר מכן היא קוראת ל-setPacketAction כדי להחליט האם לקבל או לדחות את הפקטה. לבסוף היא קוראת ל-writeToLog.

לוגיקת setPacketAction: אם ה-firewall לא פעיל, הפקטה מתקבלת. אחרת, מתבצעות מספר בדיקות. עבור פקטה שאינה TCP, נקראת הפונקציה setPacketActionAccordingToRulesTable של Rules. עבור פקטת TCP שכבר קיים עבורה חיבור (פקטה שלא יוצרת חיבור חדש), נקראת הפונקציה updateConnection של Connections. עבור פקטת TCP שיוצרת חיבור חדש, נקראת הפונקציה setSynPacketAction.

setSynPacketAction: תחילה הפקטה נבדקת ב-Rules באמצעות setPacketActionAccordingToRulesTable. אם הפקטה לא מאושרת על פי טבלת החוקים, היא עדיין יכולה להיות מאושרת אם היא יוזמת חיבור ftp-data חדש שמתאים לחיבור ftp קיים. בדיקה זו מתבצעת באמצעות הפונקציה isRelatedToFtpConnection של Connections. לבסוף, אם הפקטה מאושרת, אז הפונקציה addNewGenericConnection של Connections נקראת על מנת ליצור חיבור חדש בטבלת החיבורים.

Rules

קובץ זה אחראי על טבלת החוקים שעל פיה מתקבלות/נדחות פקטות. יש לו שתי מטרות עיקריות:

1. לייצא ל-Hooks את הפונקציה setPacketActionAccordingToRulesTable, שקובעת האם הפקטה מתקבלת או נדחית. הפונקציה תחרוץ את דינה של הפקטה על פי החוק הראשון שיתאים לה. אם אין אף חוק כזה, הפקטה תתקבל.
2. לאפשר למשתמש לראות ולעדכן את טבלת החוקים, ולהפעיל ולכבות את ה-firewall.
התקשורת עם המשתמש נעשית באמצעות char device הנקרא fw_rules, הממומש באמצעות sysfs.
יש לו כמה sysfs attributes (השמורים במערך לשם נוחות), כאשר החשוב שבהם הוא rules_table.
הפונקציות showRulesTable, setRulesTable הן פונקציות ה-show, store שלו.

Log

קובץ זה אחראי על הלוג שבו כל שורה מתארת פקטה שהתקבלה (או כמה פקטות שאוחדו לשורה אחת).
הרשומות בו שמורות כרשימה מקושרת (list של ה-Linux kernel).

גם ללוג שתי מטרות עיקריות:

1. לייצא ל-Hooks את הפונקציה writeToLog.
פונקציה זו בודקת אם רשומה דומה (זהה חוץ מאשר בשדות timestamp, count) כבר נמצאת ברשימה.
אם כן, היא מעדכנת לה את ה-timestamp ואת ה-count. אחרת, היא מוסיפה רשומה חדשה לרשימה.
2. לאפשר למשתמש לראות את שורות הלוג ולאפס אותו.
התקשורת עם המשתמש נעשית באמצעות char device הנקרא fw_log. פונקצית ה-read שלו מחזירה בכל פעם את השורה הבאה בלוג. כמו כן, יש לו גם sysfs attribute שמאפשר לאפס את הלוג (ע"י מחיקת הרשימה).

Connections

קובץ זה אחראי על טבלת חיבורי ה-TCP. החיבורים שמורים ברשימה מקושרת.

הפונקציות המיוצאות הן:

- addNewGenericConnection: יצירת חיבור TCP חדש.

- updateConnection: עדכון חיבור קיים בהתאם לפקטה שהתקבלה עכשיו, וקביעה האם לקבל או לדחות את הפקטה בהתאם למצב הנוכחי של החיבור. אם לפקטה יש data רלוונטית (ftp או http), היא נשמרת בתוך החיבור כדי לאפשר פרגמנטציה של הפקטות. לאחר מכן יש בדיקה האם החיבור מכיל פקטה מלאה, ואם כן היא מטופלת בהתאם לסוגה. פקטות FTP יכולות לגרום לשמירה של פורט ספציפי שיתאים לחיבור ftp-data בהמשך, ופקטות HTTP יכולות להיחסם אם הן מבקשות host חסום.

- isRelatedToFtpConnection: עבור פקטה היוזמת חיבור ftp-data, פונקציה זו בודקת האם יש חיבור ftp שקודם לכן אפשר חיבור כזה.

בנוסף, הקובץ מתחזק char device שמאפשר למשתמש לראות את טבלת החיבורים.

Hosts

קובץ זה אחראי על טבלת ה-hostים החסומים. הוא מייצא את הפונקציה isHostAccepted שבודק האם ה-host תקין, בהתאם לטבלת ה-hostים החסומים. כמו כן הוא מתחזק char device שמאפשר למשתמש לראות את טבלת ה-hostים או לטעון טבלה חדשה.

ממשק המשתמש

ממשק המשתמש מאפשר לנהל את המודול בצורה נוחה, מבלי לגשת לchar devices ישירות. הוא נועד להרצה כאשר המודול כבר טעון.

הקובץ userManager מכיל פונקציות המתאימות לפעולות שהממשק מאפשר ופונקציות עזר (בעיקר להדפסה).

הערה

לא אפרט מעבר לכך על הפונקציות, אך רובן מתועדות בקוד עצמו (קודם להן בלוק תיעוד).

<u>הנחות והערות נוספות</u>

- כפי שנכתב בפורום, ההנחה היא שפקטות TCP יכולות להיות מפוצלות אך הן מגיעות לפי הסדר.
- המצבים המתוארים בטבלת החיבורים מתייחסים רק לפקטות שאנחנו יודעים בוודאות שנשלחו. אין הנחה שפקטה מסוימת שקיבלנו גם התקבלה בצד השני. לכן כל המצבים הם מהצורה ...sent ולא ...received.
- אם פקטת fin מכילה בקשת http get ל-host חסום, היא תיחסם כמו כל פקטה רגילה שתעשה זאת. זה יכול לגרום לחיבור להישאר פעיל מבחינת טבלת החיבורים.
- הקובץ hosts שטוענים צריך להכיל את ה-hostים בשורות נפרדות, כאשר כל שורה נגמרת ב-\n בלבד (ולא ב-\r\n, כפי שקורה בחלק מעורכי הטקסט).
- כאשר המודול נטען, מנגנון החוקים אינו פעיל.
- כדי שלא ייווצרו בעיות סנכרון, יש לטעון/לנקות את קובץ החוקים כאשר מנגנון החוקים אינו פעיל.
- בקובץ החוקים לטעינה, בדומה לקובץ הדוגמה:
  - הפרוטוקול צריך להיות באותיות גדולות (חוץ מany).
  - accept/drop, yes/no, in/out באותיות קטנות.
  - שם החוק המקסימלי הוא למעשה 19 תווים ולא 20, כי האחרון נחוץ ל-null terminator.
- ייתכן שיש הנחות נוספות המפורטות בקוד.