



TEL AVIV UNIVERSITY

מערכות הפעלה

מרצה: ערן טרומר
סמסטר א' תשע"ב

הרצאה 13
הגנה ואבטחה (המשך)

בשיעור הקודם

❖ הגנה ואבטחה – הקדמה

❖ נזקים

❖ דרכי פריצה

❖ מינוח ושלבי פריצה

❖ הקטנת החשיפה לקוד זדוני

❖ מניעת שירות

❖ מנגנוני אימות זהות

▪ סיסמאות

▪ ביומטרי



הרשאות

- ❖ עצמים ברי הגנה: קבצים, מדריכים, שקעים, התקנים, תהליכים (בחלונות גם מנעולים ואירועים)
- ❖ לכל סוג עצם יש פעולות שניתן לאסור או להתיר למשתמשים
 - קריאה, כתיבה והרצה של קובץ, קריאה וכתיבה מהתקן או שקע
 - להרוג תהליך או לשלוח לו איתות
 - להודיע על אירוע, לחכות לו, לנעול ולשחרר מנעול (בחלונות)
- ❖ למשתמש הכל-יכול (administrator, root) מותר הכל

מטריצת גישה:

ערן	סיון	חזי	
	read, write execute	read, execute	a.out
read	read	read, write	intro.doc
		kill	תהליך 213

ייצוג מטריצת הגישה

❖ **אין** במערכת ההפעלה מבנה נתונים אחד שמייצג את המטריצה

❖ הרשאות מיוצגות באופן מפוזר

- בעיקר: יצוג עפ"י שורות

- יחד עם כל עצם נשמרת רשימת בקרת הגישה שלו (access control list), כלומר שורה במטריצה

- תהליכים עשויים להחזיק הרשאות נוספות שנקראות יכולות

- המטריצה היא איחוד רשימות בקרת הגישה והיכולות

❖ רשימות בקרת גישה שמורות בצורה דחוסה מכיוון שהן עלולות להתארך מאוד במערכות מרובות משתמשים

דחיסת רשימות בקרת גישה

❖ קבוצות משתמשים

- ניתן להגדיר קבוצות שרירותיות של משתמשים (תלמידי שנה ב', למשל)
- איבר אחד ברשימה מתיר גישה עבור כל המשתמשים בקבוצה

❖ רשימות הרשה/סרב

- רשימת בקרת הגישה מכילה רשומות הרשאה ורשומות סירוב לפעולות
- הרשימה נסרקת לפי סדר בזמן בדיקת הרשאות
- האיבר הראשון שתקף לגבי המשתמש והפעולה קובע הרשאה או סירוב
- רשומות מאוחרות קובעות כללים (להרשאה/סירוב), רשומות מוקדמות קובעות יוצאים מן הכלל
- ❖ שיתוף רשימות זהות בין עצמים (במערכת הקבצים של חלונות 2000)

בקרת גישה במערכות ספציפיות

❖ בחלונות 2000/NT (אבל לא ב-FAT): רשימות הרשה/סרב כלליות

❖ ביוניקס ולינוקס רשימות הרשה/סרב עם שלוש קבוצות בדיוק, ושלוש פעולות מותרות או אסורות לכל קבוצה

- קבוצת כל המשתמשים, קבוצה שרירותית (ממופה בעזרת `/etc/group`), וקבוצת המשתמש בעל הקובץ
- קריאה, כתיבה, והרצה של קובץ כתוכנית או פענוח דרך מדריך
- ניתן לקבוע ברירת מחדל שאוסרת כל אחת מתשע הגישות (`umask`)
- דוגמה: `r--w----` מתירה לבעל הקובץ קריאה/כתיבה, אוסרת גישה לחברי הקבוצה, ומתירה קריאה לכל המשתמשים האחרים
- דוגמה: `drwx--x--x tromer math /home/tromer`
- דוגמה: `drwxr-xr-x tromer math /home/tromer/public_html`

יכולות (capabilities)

- ❖ יצוג על פי עמודים במטריצת הגישה
- ❖ מזהה משאב בחלונות ובלינוקס/יוניקס (handle, file descriptor) הוא למעשה מצביע ליכולת שמתירה פעולות מסוימות על עצם מסוים
- ❖ ההרשאות נבדקות בזמן פתיחת העצם והיכולת נשמרת במבנה נתונים של מערכת ההפעלה; התהליך מקבל מזהה ליכולת
- ❖ היכולת ממשיכה להיות תקפה גם אם ההרשאות של העצם משתנות
- ❖ ניתן להעביר יכולות מתהליך לתהליך: ממילא תהליך יכול לבצע עבור תהליך אחר גישה לקובץ וכדומה
 - ביוניקס/לינוקס העברה של מזהי קובץ פתוח דרך שקע

דוגמה ליכולות: הרשאות באנדרואיד

❖ כל משאב רגיש מוגן ע"י יכולת (permission במינוח אנדרואיד)

❖ לכל אפליקציה יש רשימת יכולות אשר אושרו ע"י המשתמש בעת ההתקנה

❖ יכולות נפוצות

- INTERNET
- READ_CONTACTS
- ACCESS_NETWORK_STATE
- WAKE_LOCK
- ACCESS_FINE_LOCATION
- WRITE_SETTINGS
- MODIFY_AUDIO_SETTINGS
- ACCESS_COARSE_LOCATION
- CHANGE_WIFI_STATE

❖ מימוש: שילוב של הרשאות לינוקס (כולל יצירת משתמש חדש לכל אפליקציה) ומנגנון הרשאות ייעודי המנוהל מתוכנית משתמש

מדיניות הרשאות

- ❖ כללים שקובעים מה מותר למי
- ❖ את המדיניות צריך לממש בעזרת
 - ברירות מחדל ליצירת קבצים חדשים (umask)
 - מנגנון ירושת ההרשאות בין מדריכים בחלונות
 - מנגנון קביעת בעלות על קבצים חדשים (סיבית setgid במדריכים)
 - בקרה על הרשאות של קבצים קיימים
- ❖ דוגמה: קבצים חדשים במדריכים של פרווייקטים נגישים לכל חברי הפרוייקט, קבצים חדשים במדריכי הבית נגישים רק למשתמש
 - המימוש ביוניקס ולינוקס מתואר בספר

מנגנון הרשאות SELinux

- ❖ תוספת ללינוקס (ליבה+תוכניות) שנכתבה ע"י NSA ונתרמה כקוד פתוח, כיום חלק סטנדרטי מלינוקס אשר מופעל כבבירת מחדל בהפצות רבות
- ❖ לכל משאב מוצמדת תווית אבטחה (security context), מחרוזת שמשמעותה תלויה במדיניות. עבור קבצים, התווית מאוכסנת כ-
inode-extended attribute ב-inode.
- ❖ גם לכל תהליך מוצמדת תווית.
- ❖ בכל גישה של תהליך למשאב, התוויות נשלחות לבדיקה ע"י security server (מודול ייעודי בליבה) אשר מחליט האם לאשר גישה
- ❖ מדיניות האישור נקבעת ע"י מנהל המערכת ונטענת לתוך ה-security server בעזרת תוכנית משתמש (דרך ממשק קריאות מערכת).
- ❖ המדיניות לא ניתנת לשינוי על ידי משתמשים מזדמנים:
mandatory access control לעומת המנגנונים הקודמים שהם
discretionary access control.

מנגנוני רישום (logging)

- ❖ מי עשה מה (או אפילו רק ניסה)
- ❖ מצביע על ניסיונות גישה לא לגיטימיים ומאפשר לדעת מי קרא או שינה קבצים ומתי
- ❖ בחלונות ניתן לצרף לעצמים רשימת רישום גישה שמורה איזה פעולות של איזה משתמשים יש לרשום (עלול לייצר כמויות מידע גדולות!)
- ❖ אין מנגנון דומה בלינוקס ויוניקס אבל יש תוכנות שמזהות שינויים בקבצים חשובים (TripWire)
- ❖ יש בלינוקס/יוניקס מנגנון רישום לאירועים חשובים (לא ניסיונות גישה) כמו ניסיונות התחברות של root, אל `/var/log/...`
- ❖ כיצד לאפשר לכל המשתמשים לדווח אירועים בלי להסתכן בשיבוש קובץ יומן האירועים?

התחזות מותרת

❖ לעיתים צריך לבצע מטלה מסוימת עם הרשאות של תהליך אחר

- תוכנת שרת שמריצה תוכניות במועדים קבועים עבור משתמשים צריכה להריץ כל תכנית עם ההרשאות של המשתמש שביקש להריץ אותה (ביוניקס, crond)

- הרצה של תסריטים דרך שרת אינטרנט (ASP, CGI)

- תוכנות שרת שמאפשרות התחברות למחשב (telnet, ssh, login)

- תוכנות להעברת קבצים (ftp)

- הרצת פקודות בעזרת sudo

❖ לתהליך של המשתמש הכל יכול מותר להתחזות לכל משתמש

- בחלונות ניתן להעניק יכולת התחזות לתהליכים של משתמשים אחרים

- ביוניקס אפשר להתיר לתהליך שמריץ תוכנית להתחזות לבעל התוכנית

הגברה

❖ לעיתים צריך לעדן את מנגנון ההרשאות

- להתיר למשתמשים לקרוא/לשנות קובץ, אבל רק בדרכים מסוימות
- להציב ולהסיר מערכת קבצים מתקליטור אבל לא מערכות קב' אחרות
- להפעיל שירותי מערכת, לדוגמה מנהל הדפסה או מנהל יומן ארועים

❖ הגברה ביוניקס/לינוקס

- משתמש בעל משאב שרוצה לפקח על הגישות אליו משתמש בתוכנית שמשתמשים אחרים מריצים על מנת לגשת למשאב
- התוכנית בבעלות המשתמש ושמורה עם סיבית setuid דולקת
- התהליך שמריץ את התוכנית יכול לעבור בין זהות המשתמש המריץ ובין המשתמש בעל התוכנית, שהוא גם בעל המשאב; הרשאות המריץ הוגברו

❖ סכנות לבעלי התוכנית: שגיאות תכנות, הפתעות במערכת

- הקבצים (לדוגמה symbolic link מ-`~/finger`), משתני סביבה, הפסקה פתאומית (SIGHUP)