

פרק 10

הגנה ואבטחה

על מה נדון

- ❖ הסכנות האורבות למערכות מחשב: פריצות ותקיפות
 - פריצה: מי שאינו משתמש לגיטימי מצליח לבצע פעולות או משתמש לגיטימי מצליח לבצע פעולות שאינו מורשה לבצע
 - תקיפה: גורם חיצוני גורם נזק בלי להשיג תועלת ישירה
- ❖ אימות זהות: מי שם?
- ❖ הרשאות: איזה פעולות מותרות לכל משתמש
- ❖ מנגנוני רישום: מי עשה מה
- ❖ הגברת הרשאות והתחזות מותרת

על מה נדון

❖ הסכנות האורבות למערכות מחשב: פריצות ותקיפות

▪ פריצה: מי שאינו משתמש לגיטימי מצליח לבצע פעולות או משתמש לגיטימי מצליח לבצע פעולות שאינו מורשה לבצע

▪ תקיפה: גורם חיצוני גורם נזק בלי להשיג תועלת ישירה

❖ אימות זהות: מי שם? **authentication**

**permissions,
access control**

❖ הרשאות: איזה פעולות מותרות לכל משתמש

❖ מנגנוני רישום: מי עשה מה **logging**

❖ הגברת הרשאות והתחזות מותרת

נזקים מפריצות

❖ קריאת מידע על ידי מי שאינו מורשה

▪ מבחן, מידע פיננסי

❖ שינוי או הוספת מידע בקובץ

▪ שינוי ציון או יתרה בבנק

❖ מחיקת מידע

▪ תיק פלילי, עדויות לעבירה

❖ מניעת שירות

▪ הפלת אתר אינטרנט

❖ שימוש במשאבים ללא תשלום

▪ כוח חישוב, נייר

נזקים מפריצות



❖ קריאת מידע על ידי מי שאינו מורשה

▪ מבחן, מידע פיננסי

סיסמאות גישה, מפתחות הצפנה

❖ שינוי או הוספת מידע בקובץ

▪ שינוי ציון או יתרה בבנק

שינוי הרשאות

❖ מחיקת מידע

▪ תיק פלילי, עדויות לעבירה

עדויות לפריצה למערכת מחשב

❖ מניעת שירות

▪ הפלת אתר אינטרנט

הפסקת פעולת שירות ניטור

❖ שימוש במשאבים ללא תשלום

▪ כוח חישוב, נייר

תקשורת

דרכי פריצה

❖ גישה פיזית לחומרה

- גניבה של מחשב או רק של דיסקים או סרטים
- ציטוט לקו תקשורת או לתקשורת אלחוטית
- חדירה לחדר מחשב שממנו ניתן להשתמש במחשב ללא אימות זהות

❖ התחזות (גניבת זהות של משתמש לגיטימי)

- מבחינת מערכת ההפעלה, מי שעובר את מנגנון אימות הזהות (סיסמה למשל) הוא משתמש לגיטימי

❖ הרצת קוד זדוני

- תוכנית שגורמת למשתמש לגיטימי לבצע פעולות שמותרות לו מבלי שהוא מודע לכך שהוא מבצע אותן
- הפעולות הללו משרתות את הפורץ (למשל שולחות לו מידע) ו/או מזיקות למשתמש הלגיטימי

❖ ניצול שגיאות בקוד קיים

- שליחת קלט הגורם לתוכנה שכבר רצה לפעול באופן לא מתוכנן
- כולל שגיאות במערכת ההפעלה עצמה

דרכי פריצה

❖ גישה פיזית לחומרה

- גניבה של מחשב או רק של דיסקים או סרטים
- ציטוט לקו תקשורת או לתקשורת אלחוטית
- חדירה לחדר מחשב שממנו ניתן להשתמש במחשב ללא אימות זהות

❖ התחזות (גניבת זהות של משתמש לגיטימי)

- מבחינת מערכת ההפעלה, מי שעובר את מנגנון אימות הזהות (סיסמה למשל) הוא משתמש לגיטימי

❖ הרצת קוד זדוני **malware**

- תוכנית שגורמת למשתמש לגיטימי לבצע פעולות שמותרות לו מבלי שהוא מודע לכך שהוא מבצע אותן
- הפעולות הללו משרתות את הפורץ (למשל שולחות לו מידע) ו/או מזיקות למשתמש הלגיטימי

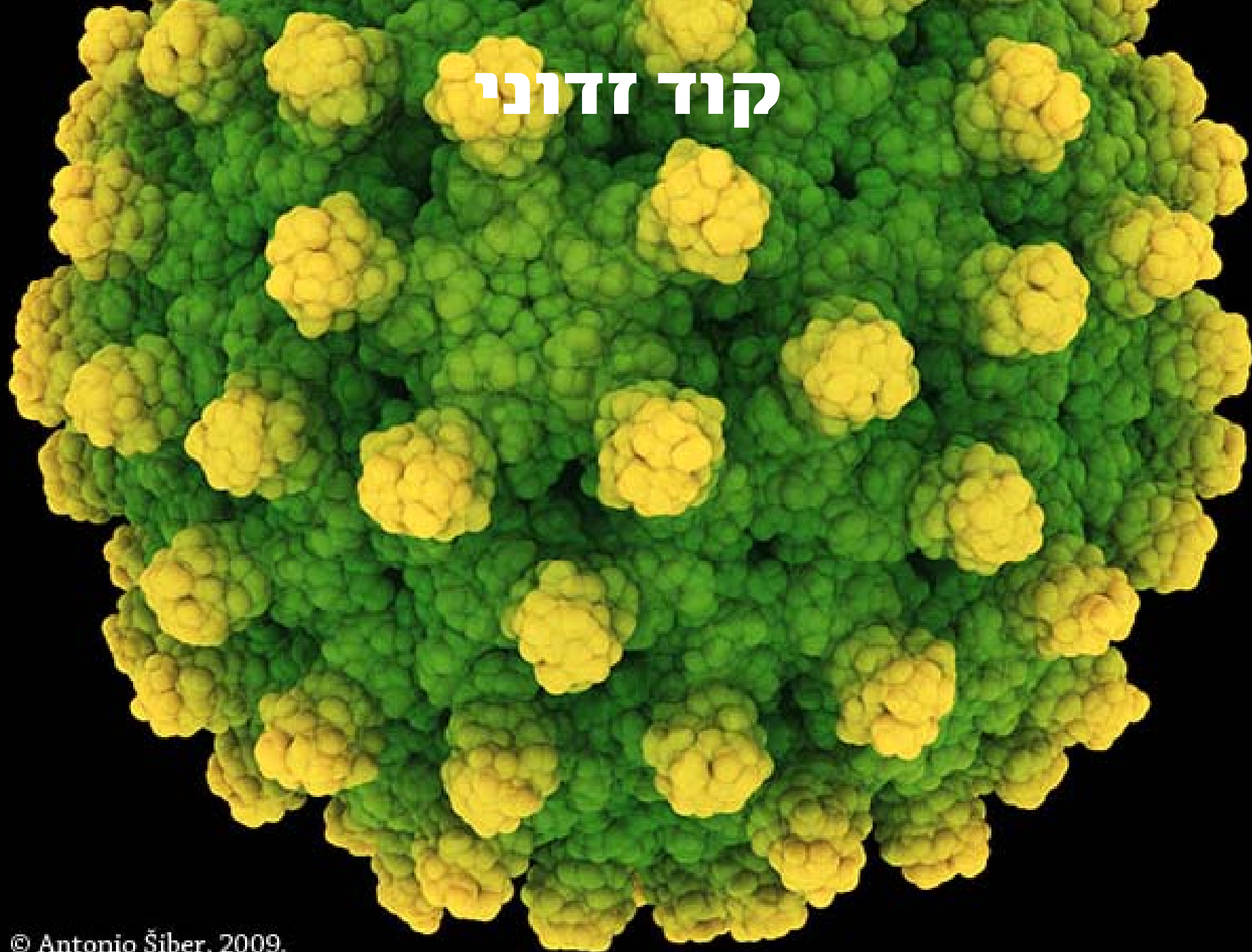
❖ ניצול שגיאות בקוד קיים

- שליחת קלט הגורם לתוכנה שכבר רצה לפעול באופן לא מתוכנן
- כולל שגיאות במערכת ההפעלה עצמה

קוד זדוני



קוד זדוני



קוד זדוני



קוד זדוני

- ❖ קבצי הרצה שמגיעים כתוספת לדואר אלקטרוני
 - המשתמש מריץ תוכנית או פותח קובץ שמכיל גם תוכנית
 - הקוד הזדוני יכול להפיץ עצמו לאנשי הקשר של המשתמש
- ❖ קבצים באתרי אינטרנט
- ❖ תוספים ותוכן דינמי בדפדפן
 - Java applets, ActiveX controls, plugins, וכו'
- ❖ "הפתעות" בתוכנה לגיטימית
- ❖ תוכנות מסחריות ששולחות מידע פרטי לחברה המפתחת
 - התוכנה מבצעת את תפקידה אבל גם אוספת מידע ללא ידיעת המשתמש
 - Amazon "1984" recall, Sony rootkit, וכו'
- ❖ ניצול שגיאה בקוד קיים כדי להריץ תוכנה אחרת

מינוח

- ❖ Bug
might be a
- ❖ Vulnerability
for which someone will write an
- ❖ Exploit
that hijacks control and runs
- ❖ Shellcode
that typically installs a
- ❖ Rootkit
that “own” the computer and hides the traces, often
making the computer a part of a
- ❖ Botnet

דוגמה לפגיעות: **buffer overflow**

```
void some_function() {  
    char my_string[500];  
    printf("name: ");  
    gets(my_string);  
    ...  
}
```

- ❖ ניתן בצורה כזו להשתיל קוד מכונה ולשנות את כתובת החזרה מהפונקציה כך שתקפוץ לתוך הקוד ששתלנו
- ❖ ניתן לנצל פגיעות בכל תכנית של מערכת ההפעלה
- ❖ ניתן לנצל פגיעות בתכנה המממשת אחת משכבות התקשורת

הקטנת החשיפה לקוד זדוני

- ❖ הימנעות משימוש בתוכניות ממקור לא ידוע
 - סיכויים פחותים לסוס טרויאני בתוכנה של חברה ידועה משום שגם לחברה יש אינטרס למנוע הימצאות סוס טרויאני בתוכנה (פגיעה בשמה)
- ❖ במקרים קיצוניים, הימנעות מהרצת תוכנה שלא ניתן לבדוק את קוד המקור שלה
 - תוכנות פופולריות עם קוד פתוח נקראות על ידי רבים והסיכוי לסוס טרויאני קטן
- ❖ הרצת תוכנות שרת שעלולות להיות חשופות לתקיפה (שרתי HTTP למשל) עם הרשאות מינימליות לביצוע תפקידן
- ❖ שימוש בתוכנות אנטי-וירוס שמזהות קוד זדוני מוכר או דפוסי פעולה לא רגילים (למשל סדר קריאות מערכת)
- ❖ ביקורות על גישה לקבצים ועל מידע שיוצא לרשת



הקטנת מעטפת החשיפה

- ❖ firewall חוסם גישה מבחוץ לשירותים ברשת המקומית שעלולים להיות פגיעים (בגלל שגיאות תוכנה או סיבות אחרות)
- חסר מצב: מאפשר גישה לשערים (ports) שהוגדרו וחוסם גישה לשאר
- בעל מצב: מאפשר פעולות לפי מצב הקשר
- ❖ השירותים הפגיעים זמינים בלי הגבלה ברשת המקומית
- ❖ מבוסס על ההנחה (הבעייתית לפעמים) שיש הבדל מהותי בסיכונים מהרשת המקומית לעומת סיכונים מהרשת החיצונית

תקיפות מניעת שירות

denial of service

- ❖ שימוש אינטנסיבי במשאבים שגורם למניעת שירות או מתן שירות איטי למשתמשים לגיטימיים
- ❖ בדרך כלל התקיפה מתבצעת על ידי שימוש בשירות שניתן לכל מחשב ברשת, לא רק לקבוצת משתמשים קטנה, למשל
 - שרתי HTTP מספקים קבצים לכל דורש
 - שרתי דואר אלקטרוני מוכנים לקבל דואר מכל מקור
- ❖ הצפת שרתים כאלה בבקשות שירות מאיטה או מפילה את השרת
 - נפילות בגלל פגמים שלא מתגלים תחת עומס רגיל או בגלל מיצוי של משאבים, כמו קבצי יומן אירועים שממלאים את הדיסק
- ❖ קשה למנוע כאלה תקיפות כי הבקשות לגיטימיות באופיין

הגברת האפקט של מניעת שירות

- ❖ השתלה מוקדמת של קוד תקיפה במחשבים רבים בעזרת וירוסים או תולעים
- ❖ הקוד הזדוני ממתין בשקט לאיתות חיצוני מהרשת
- ❖ המחשב הנגוע נקרא zombie או bot והרשת כולה botnet

אימות זהות: סיסמאות

- ❖ מחרוזת שרק המשתמש הלגיטימי יודע
- ❖ אמצעי זיהוי נוח ואמין
- ❖ אנשים נוטים לשכוח סיסמאות שאינם משתמשים בהן בתדירות
- ❖ אנשים רושמים סיסמאות קשות לזכירה או שמתחלפות תדיר
 - פורצים עלולים למצוא או לגנוב את הרישומים הללו
- ❖ אנשים בוחרים סיסמאות צפויות אם נותנים להם לבחור
 - שמות פרטיים, ימי הולדת, וכדומה

גניבת סיסמאות

- ❖ ניחוש (אם הן קלות)
- ❖ בדיקת סיסמאות רבות באופן אוטומטי על ידי תוכנית
 - אפשרי אם ניתן להפעיל את תוכנית ההתחברות באופן לא אינטראקטיבי
- ❖ ציתות לסיסמה כאשר היא עוברת ברשת או בקו תקשורת
- ❖ גישה לקובץ שסיסמאות רשומות בו
- ❖ שימוש במידע שדולף ממנגנון בדיקת הסיסמה (מעבר לכן/לא)
 - למשל פריצת סיסמאות ב-TENEX בעזרת מנגנון חריגי הדף
- ❖ הרצת תוכנית שמתחזה לתוכנית ההתחברות הרגילה של המחשב וקולטת את שם המשתמש והסיסמה שלו

התגוננות מפיצוח סיסמאות

- ❖ להכריח משתמשים לבחור סיסמאות ארוכות וקשות לניחוש
 - ניתן גם להכריח משתמשים להחליף סיסמאות לעיתים קרובות
 - בומרנג: אנשים נוטים לרשום סיסמאות קשות או שמתחלפות תדיר
- ❖ מנגנון בדיקת הסיסמאות צריך לענות רק כן/לא ולא לאפשר ניסיונות התחברות חוזרים מהירים
- ❖ סיסמאות צריכות להיות מועברות בקווי תקשורת ולהיות שמורות בקובץ בצורה מוצפנת
 - מערכת ההפעלה שומרת רק פונקציה $f(p)$ של הסיסמה p . בכל פעם שהמשתמש מקליד את p מחשבים מייד את $f(p)$ על מנת לבדוק את הסיסמה, ומוחקים מייד את p מהזיכרון
 - עדיין פגיע לחיפוש ממצה, ולכן קובץ הסיסמאות צריך להיות מוגן מפני קריאה על ידי משתמשים רגילים (`/etc/passwd` לעומת `/etc/shadow`)
- ❖ מנגנון מיוחד בלתי ניתן להתחזות לבקשת סיסמה (`alt-ctrl-del`)

אימות זהות: אתגרים וסיסמאות

חד-פעמיות

❖ **אימות זהות על ידי הוכחת בעלות על חפץ פיזי**

- בדרך כלל בשילוב סיסמה רגילה למניעת פריצה על ידי גניבת החפץ

❖ **סיסמה חד פעמית:**

- דף מודפס שבכל התחברות משתמשים בסיסמה אחרת שבו

- מחשבון מיוחד שמייצר סיסמה חדשה כל פרק זמן קצר (כדקה)

- המחשב יודע מהי הסיסמה הבאה בסדרה או הסיסמה לכל נקודת זמן

- ציטוט אינו מאפשר התחברות

❖ **אתגר**

- מחשבון שעונה על חידות שהמחשב שמתחברים אליו מציג

- המחשב יודע מה התשובה הנכונה לכל חידה (תלוי בזמן ובמחשבון)

❖ **שיטה יקרה (מחשבוניים), מסורבלת ליישום ושימוש, אך בטוחה**

אימות זהות ביומטרי

- ❖ זיהוי אדם על פי תכונות פיזיות: מאפייני קול, טביעת אצבע, פנים, נימי דם ברשתית, צורת כף היד וכו'
- ❖ תכונה ייחודית אבל לא בהכרח סודית; לא ניתנת להחלפה
- ❖ לחלק מהתכונות דרושים אמצעי קלט לאחרים רק מיקרופון או

מצלמה

- ❖ אחוז טעויות מסוים בדרך כלל

- סירוב למשתמש לגיטימי או אישור למתחזה

- ❖ ניתן לעיתים לשטות במנגנון האימות בעזרת

הקלטה/בובה/תמונה

- יש דרכים להתגבר על כך, למשל על ידי כך שמבקשים מהמשתמש לומר

משפט אקראי ובודקים גם את תוכן הדברים וגם את חתימת הקול



הרשאות

- ❖ עצמים ברי הגנה: קבצים, מדריכים, שקעים, התקנים, תהליכים (בחלונות גם מנעולים ואירועים)
- ❖ לכל סוג עצם יש פעולות שניתן לאסור או להתיר למשתמשים
 - קריאה וכתובה של קובץ, קריאה וכתובה 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

ערן	סיון	חזי	אמיר	
			read,wr.	a.out
read	read	read,wr.	read	intro.doc
		kill		תהליך 213

מטריצת גישה:

ייצוג מטריצת הגישה

- ❖ **אין** במערכת ההפעלה מבנה נתונים אחד שמייצג את המטריצה
- ❖ הרשאות מיוצגות באופן מפוזר
 - בעיקר בייצוג על פי שורות
 - יחד עם כל עצם נשמרת רשימת בקרת הגישה שלו (access control list), כלומר שורה במטריצה
 - תהליכים עשויים להחזיק הרשאות נוספות שנקראות יכולות (capabilities)
 - 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--r--r--` מתירה לבעל הקובץ קריאה/כתיבה, אוסרת על כל סוגי הגישה לחברי הקבוצה, ומתירה קריאה לכל מי שאינו בקבוצה

יכולות

- ❖ ייצוג על פי עמודות 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

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 פרוייקטים נגישים לכל חברי הפרוייקט, קבצים חדשים במדריכי הבית נגישים רק למשתמש
 - המימוש ביוניקס ולינוקס מתואר בספר

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 פרויקטים נגישים לכל חברי הפרוייקט, קבצים חדשים במדריכי הבית נגישים רק למשתמש
 - המימוש ביוניקס ולינוקס מתואר בספר

מנגנון ההרשאות ב-SELinux

- ❖ תוספת ללינוקס (גרעין+תהליכים), כיום חלק סטנדרטי מלינוקס אשר מופעל כבברירת מחדל בהפצות רבות
- ❖ לכל משאב מוצמדת תווית אבטחה, מחרוזת שמשמעותה תלויה במדיניות. עבור קבצים, התווית מאוכסנת ב-inode
- ❖ גם לכל תהליך מוצמדת תווית
- ❖ בכל גישה של תהליך למשאב, התוויות נשלחות לאישור סירוב ע"י מודול ייעודי בגרעין, שתוכנית מיוחדת טוענת לו מדיניות
- ❖ המדיניות לא ניתנת לשינוי על ידי משתמשים מזדמנים: זה mandatory access control לעומת המנגנונים הקודמים שהם discretionary access control

מנגנוני רישום

- ❖ 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

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
- הרצה של תסריטים דרך שרת אינטרנט (ASP, CGI)
- תוכנות שרת שמאפשרות התחברות למחשב (telnet, ssh, login)
- תוכנות להעברת קבצים (ftp)
- הרצת פקודות כמישהו אחר (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 לשנות קובץ, אבל רק שינויים מסוימים
- להציב ולהסיר מערכת קבצים מתקליטור אבל לא מערכות קב' אחרות
- להפעיל שירותי מערכת, כמו מנהל הדפסה או מנהל יומן אירועים

❖ הגברה ביוניקס/לינוקס

- משתמש בעל משאב שרוצה לפקח על הגישות אליו שומר תוכנית שמתמשים אחרים יריצו 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-`~/plan`), משתני סביבה, הפסקה פתאומית (SIGHUP)