Linux Essentials 010-160 + Bonuses

קורס זה הינו שדרוג של ההסמכה המוכרת **Linux Essentials 010-160** בתוספת נושאים של **LPI 101(LPIC-1)** עם דגש על **אבטחת מידע והכנה ל-OSCP**.

קורס זה הינו בפיקוח של Linux Professional Institute (LPI) ומזכה את התלמיד בהנחה לבחינה הבינלאומית LPIC הנחשבת!

קורס זה מתאים לבני נוער ולחסרי רקע בעולם ה-Linux, Linux היא מערכת הפעלה הכרחית הן בעולם ניהול הרשת והן בעולם אבטחת המידע והסייבר.

בקורס הזה אנחנו נעבוד עם מערכת ההפעלה Kali Linux ו-Ubuntu.

שיעור – 3 – נתיבי מערכת ומערכת הקבצים

מערכת הקבצים FHS – מה נמצא איפה?

FHS - File Hierarchy Standard הינו הסטנדרט העולמי לעבודה עם קבצים במערכת ההפעלה Linux ובגדול מה נמצא איפה?

/bin אלא הפקודות שמשתמש רגיל יכול לבצע binaries

/sbin אלא הם פקודות מערכת system binaries

/boot כאן נמצא ה- grub וכל הקבצים הדרושים למערכת ההפעלה לעלות

/dev בתיקייה זו מאוחסנים קבצים אשר מייצגים את ההתקנים השונים במחשב devices, ההתייחסות להתקנים היא כאל קבצים, כל התקן אשר ה-kernel יכול להבין ייוצג על ידי תיקייה או קובץ.

/etc כל קבצי הקונפיגורציה נמצאים כאן בדרך כלל מדבור בקבצים עם הסיומת conf.

/home תיקיות הבית של כל המשתמשים

/lib כל קבצי הספרייה נמצאים כאן עבור פקודות של המערכת

/proc כל הקבצים כאן הם לא פיזיים אלא נוצרים בעת העלאת המכונה ומטרתם להציג את מצב המערכת לכן אין טעם לשנות ערכים בתיקייה זו

/root תיקיית הבית של המשתמש ה-root

/tmp זו תיקייה שמוגדר בה **sticky bit** לטובת קבצים זמניים של המשתמשים

/usr כל התוכנות שמשתמשים יצטרכו, בגדול מדובר במשחקים ותוכנות עבור המשתמשים

/var קבצים הקשורות יותר לפעולת השרת בדרך כלל קבצי לוגים או אתר וכו'

/media תיקייה ל-mount עבור replaceable media, לדוגמה דיסק און קי.

/mnt זו תיקיה ל-mount של כוננים קשיחים.

/ lost+found תיקייה זו מכילה קבצים אשר משמשים לשחזור במקרה של קריסת המערכת או כאשר למחיצה לא בוצע unmounts לפני כיבוי המערכת.

/sys הקישור בין החומרה לתוכנה

/swap זיכרון נוסף שהוא בעצם חלק מה – ram במידה ונגמר המקום

/opt תיקיית optional המשמשת לחבילות Packages שאנחנו מתקינים בעצמו – לא בשימוש יותר.

mounting & unmounting

כדי לעבוד עם כונן עלי לבצע לו קודם כל mount, לדוגמה הוספת דיסק און קי או cdrom.

```
mount -t ext3 /dev/sdb1 /mnt/hard_drive
```

כדי לראות את כל הדברים במכונה שבצענו להם mount נבצע

```
mount
```

כדי לבטל mount שבצענו עלנו לבצע את הפקודה

```
umount /dev/sdb1
umount -a
```

אפשר גם לבצע bind בין תיקיות ולהגדיר להם הרשאות

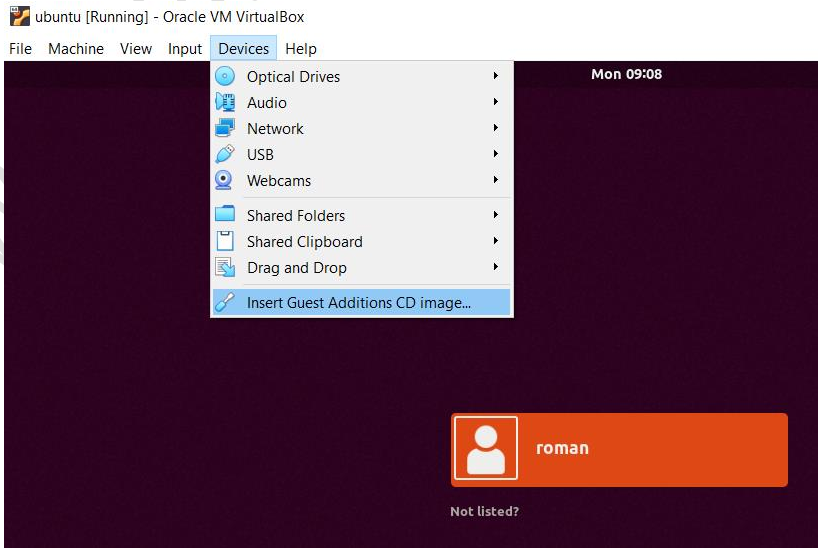
```
mount --bind /home/roman /backup            #hard link for directory…
mount -o remount,noexec /tmp                #remount without exec option
```

כעיקרון נושא זה הינו חלק מ-101 LPI ושם נדון בנושא זה בהרחבה.

תרגיל

1. מה תפקיד התיקייה **/dev**?
2. מה תפקיד התיקייה **/mnt**?
3. הכנס את ה-guest additions כפי שעשינו בשיעור הראשון ב-day1 על ידי לחיצה על devices ואז Insert guest additions.

תרגיל זה יש לעשות על מכונה שהותקנה ולא הופעלה ב-livecd מכיוון שמכונת ה-live היא בעצמה מופעלת מתוך הcdrom.



בפעולה זו חיברת cdrom למכונה, ה-cdrom נמצא ממופה לקובץ

```
/dev/sr0
```

בצעו unmout –a לפני שאתם מתחילים את התרגיל

```
umount -a
```

עליכם לבצע מיפוי של ה-**cdrom** לתוך התיקייה **/media/guest** באמצעות הפקודה mount ולהציג את תוכן התיקייה.

4. העתיקו את תוכן התיקייה **/media/guest** לתוך התיקייה **/root/guest** בתיקיית הבית של המשתמש root.

הגדרת משתנה סביבתי (שדרוג לפי LPI 101)

כאשר משתמש מתחבר לחשבון שלו בלינוקס הקובץ /etc/profile נקרא והוא מבצע את הפועלות המוגדרות לו, אחת הפעולות היא לקרוא את הקובץ bash.bashrc שהינו מכיל את הגדרות הסביבה וקובע אליו פקודות אתם יכולים להריץ ומאילו נתיבים.

בדיוק כמו במערכת ההפעלה Windows.

על מנת להציג את הסביבה אנו משתמשים באחת הפקודות הבאות:

- env
- printenv
- set

אחד המשתנים הסביבתיים החשובים ביותר הינו **PATH**, משתנה סביבתי זה קובע מהיכן ניתן להריץ פקודות במערכת, הוא מופיע או בקובץ /etc/profile או בקובץ /etc/bash.bashrc תלוי בהפצה של הלינוקס ונראה כך:

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
```

אופן הרצת הפקודה הינו משמאל לימין לכן אם תהיה התאמה בצד שמאל של התיקיות זאת הפקודה שתתבצע במידה ותופיע יותר מפעם אחת.

כדי להגדיר משתנה חדש לסביבה משתמשים בפקודות הבאות:

```
abc=123
echo $abc
set
export abc
env
```

כדי להציג את המשתנה שלנו עלינו לכתוב:

```
$abc
echo $abc
```

אז איך מעדכנים את path בלי לפגוע בpath הקודם?

```
PATH=$PATH:.
```

או

```
PATH = $PATH:/root/scripts/
```

הפקודה export הופכת את המשתנה לגלובלי עבור sub process כך שגם תת תהליכים יכירו את המשתנה החדש, דוגמה טובה ליצור script בbash

```
export PATH
export EDITOR=vi
export HOME=/tmp
export PWD=/var/www
```

כשעושים export לפונקציה אנו מוסיפים f-

```
function roman() { echo hello; }
export -f roman
```

כדי למחוק משתנה מהenv שלנו עלינו לכתוב

```
unset abc
```

לינקים hard and symbolic links

גם ב-Linux כמו ב-windows ניתן ליצור shortcuts (קיצורים) כך שפניה לקובץ אחד תפנה אותנו לקובץ אחר.

כדי ליצור לינוק אנו משתמשים בפקודה **ln** ולפקודה יש 2 אפשרויות:

- **Hard link** – מקושר לאותו inod **לא עובד** על תיקיות, נושא ה-inod הינו חלק מחומר של LPI 101.
- **symbolic\Soft link** – מקושר לשם הקובץ ולא ל-inod ולכן אסור לשנות את שם הקובץ במידה ואני עובד עם soft-link **עובד** גם עם תיקיות.

כדי ליצור soft לינק משתמשים בפקודה

```
ln -s file.txt softy.txt
```

אפשר לעשות גם לינק לתיקייה וכך כל שינוי מתעדכן בתיקייה השנייה

```
ln -s /root /home/roman/root
```

הבעיה שתיקיית היעד תראה כמו קובץ וזה טיפה לא נוח.

למה אני משתמש בכלל ב-soft לינק?

כיוון שלא ניתן לבצע hard link בין מחיצות וכוננים שונים בגלל שה-inodים בכל כונן שונים.

אחד השימושיים העיקריים של הלינקים הוא לארגן את הסביבה שתיהיה לכם נוחה מבלי לשנות מיקומים של קבצים.

```
ln -s grub.conf /boot/grub/menu.lst
```

אם אני מבצע לינק לא תקין כמו קובץ לתיקייה המערכת יוצרת את הקובץ אבל הלינק יהיה שבור וצבוע באדום.

כמו כן בעת ביצוע לינקים יש להגדיר נתיב מלא אחרת אם נעביר את הקובץ הלינק לא יעבוד.

```
ln file.txt folder/
ln /home/roman/file.txt folder/
```

כדי לבצע hard link אני בסך הכל כותב את הפקודה כמו שהיא ולא מוסיף פרמטרים מה שיראה כך:

```
ln file.txt hardlink.txt
ls -li
```

פקודות חיפוש

ישנם כמה פקודות שימושיות לחיפוש קבצים במערכת, אחת הפקודות המרכזיות הינה **find**, פקודה זו איטית יחסית ומכילה אפשרויות חיפוש רבות:

- **-size M\G\K** חיפוש לפי גדלים
- **-name** חיפוש על פי שם
- **-type f\d\l** לחפש קובץ או תיקייה

להלן מספר דוגמאות המציגות שימוש ב-**Globing** בחיפוש:

```
find /root/ -name "*.py"
find . -name "f0*"
find /root/ -name "[a-z]*.*"
find . -size +5M
```

חיפוש קבצים, תיקיות או לינקים

```
find . -type f -name roman
find . -type d -name roman
find . -type l -name roman
```

כמו ניתן להוסיף ls- כדי לראות הרשאות

```
find / -name passwd -ls
```

הרצת פקודה על הקבצים שנמצאו על ידי find (חומר של LPI 101)

- + עם שם הקובץ
- \ ללא שם הקובץ

```
find / -name "test*" -exec grep aaaa {} \;
```

התוצאה תראה כך:

```
root@debian:~# find -name "test*" -exec grep aaa {} +;
./testfile1:aaaa
./testfile:aaaa
./testfile2:aaaa
./testfile3:aaaa
root@debian:~# find -name "test*" -exec grep aaa {} \;
aaaa
aaaa
aaaa
aaaa
root@debian:~#
```

תרגיל

1. בצע חיפוש במערכת עבור הקובץ services, בעת ביצוע החיפוש אל תציג שגיאות על המסך ובתוך הקובץ תמצאו את הפורט של שירות ה-FTP.

פקודה נוספת שיש להכיר בעת ביצוע חיפוש הינו **locate** פקודה זו מחפשת במסד נתונים שהיא יוצרת ולכן היא מבצעת חיפוש מאוד מהיר, **לא לשכוח לעדכן את מסד הנתונים לפני החיפוש.**

```
updated
locate roman
```

קובץ ההגדרות של השירות נמצא ב

```
/etc/updated.conf
```

אם נרצה שהפקודה לא תחפש בנתיב כלשהו נוסיף אותו ב-PRUNEPATHS ב-PRUNEFS.

אם נרצה לבצע חיפוש מהיר לפקודת מערכת כלשהי שנמצאת ב-path אפשר להשתמש בפקודות:

- **which** – מציג איפה נמצאת הפקודה
- **whereis** – מציג איפה נמצאת הפקודה והמדריך לפקודה

בצורה הבאה:

```
which ls
```

אם נרצה לראות את הקבצים הנוספים כמו איפה נמצא ה-man של הפקודה אז נשתמש ב-whereis

```
whereis
```

דחיסת קבצים

דחיסה היא פעולה חיונית לצורך תחזוקת המכונה שלכם ושמירה על גיבויים הן לוגים והן מערכות שאתם צריכים.
תוכנות רבות ופרויקטים מגיעים בפורמט דחוס על מנת לא להעמיס על תעבורת הרשת וכו...

קיימים פורמטי דחסיה רבים בעולם ה-Linux בפרק זה אפרט את רובם.

zip

פקודת הדחיסה המוכרת ביותר.

הדחיסה מתבצעת באמצעות הפקודה zip והפתיחה באמצעות unzip, במידה והפקודה אינה מותקנת ניתן להשתמש ב-apt על מנת להתקין אותה כך:

```
apt install zip
```

השימוש בפקודה יראה כך:

```
zip file.zip file1 file2 file
zip -r backup.zip /root/prog/*
unzip backup.zip
```

gzip

פקודה דחיסה מעולה ששומרת על הרשאות הקובץ בעת הדחיסה.

c – לא באמת מכווץ אלא מציג על המסך את הכיווץ לכן יש לציין stdout אם אנו רוצים לשמור את התוצאה.

d – פותח את הכיווץ

r – רקורסיבי ייכנס לתיקייה וידחוס את כל הקבצים בפנים.

```
gzip keydrive.img
```

אם אני משתמש בדגל c– צריך לציין stdout כך לא יפגע קובץ המקור

```
gzip -c test > test.gz
gzip -c test > [file name].gz
```

כדי לפתח את הכיווץ נשתמש בפקודות הבאות

```
gunzip keydrive.gz
gzip -d roman.gz
```

כאשר דוחסים תיקייה אי אפשר לפתוח אותה

bzip2

פקודה שמכווצת באמצעות אלגוריתם שונה

```
bzip2
```

k – keep כדי להשאיר את הקובץ זהה ל file.gz -> gzip -c file

```
bunzip2
```

סוג הדחיסה הינו bz2.

tar

פקודה שמאחדת בתוכה את שני הדוחסים הקודמים ויודעת לעבוד גם עם תיקיות, gzip או bzip לא דוחסים וכאן tar נכנס לתמונה, בנוסף זהו הפורמט הסטנדרטי בעת דחיסה ב-Linux.

הדגלים של הפקודה:

- c – בצע דחיסה
- x – פתיחת דחיסה (ההפך מ-c)
- t – מציג את תוכן ה-tar מבלי לפתוח אותו
- v – תציג מידע נוסף בעת הדחיסה כמו גודל הקובץ והרשאות
- f – להציג את הקבצים ותמיד בא ביחד עם v–
- z – מציין gzip
- j – מציין bzip2
- J – מציין xz **פורמט נוסף השייך לחומר של LPI 101**
- C – לאן לחלץ את המידע

דוגמה לשימוש בפקודה:

```
tar -xvf roman.tar
tar -zcvf archive.tar.gz roman/ test/
tar -jcvf archive.tar.bz2 roman/ test/
tar -zxvf roman.tar.bz2 -C /media/
```

תרגיל

על מנת לבצע דחיסה ולתת לשם הקובץ את התאריך של היום נבצע את הפקודה הבאה:

```
date +%Y-%m-%d
```

שרשור תוצאות הפקודה בתוך פקודה מתבצע באחת מהשיטות הבאות:

- $()
- ``

לדוגמה:

```
echo `ls`
echo $(ls)
```

עליכם לדחוס את כל הקבצים בתיקיה root וליצור גיבוי בשם backup_file_current_date במקום current_date יש להציג את התאריך של היום ולשמור את הקובץ בתיקיה tmp/.

פתרון:

```
tar -zcvf /tmp/backup_file_$(date +\%Y-\%m-\%d).tar.gz  /root
tar -zcvf /tmp/backup_file_`date +%Y-%m-%d`.tar.gz  /root
```