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 – 2 – עבודה בסיסית בסביבת Linux

על מנת לעבוד בלינוקס עם ממש הפקודה עלינו לפתוח shell, להלן רשימת סוגי shell שניתן לפתוח:

- **sh** – אחד ממשקי הפקודות הראשונים
- **busybox** – ash זה ממשק פקודה למוצרי iot.
- **bash** – Bourne Again Shell ממשק הפקודה
- **ksh** – Korn shell
- **tcsh** – Tee See Shell
- **zsh** – Zee Shell

התוכנה שמפעילה את ה-shell ב- Terminal:

- **gnome-terminal**
- **konsole**
- **console**
- **xterm**
- **rxvt**
- **kvt**
- **nxterm**
- **eterm**

לחיצה ימנית על המסך ו-open terminal ברוב ה-UI גם עובד.

פקודות שימושיות

לאחר שהבנתם איך פתוחים ממשק פקודה עליכם להכיר מספר פקודות חשובות שיעזרו לכם לעבוד עם מערכת ההפעלה.

ls – מציג את כל הקבצים במקום בו אתם נמצאים

- d – בתיקייה הנוכחית.
- i – מציג inode.
- l – מציג את מידע מפורט על כל קובץ.
- t – מציג לפי תאריך.
- r – מציג לפי reverse.
- S – לסנן לפי גודל, שימו לב 'S' גדולה ולא קטנה 's'.

לדוגמה:

```
ls -l
```

cd – פקודת הניווט במערכת, זו הפקודה השימושית ביותר.

- / – ראש העץ, מתחיל את הניווט מתחילת העץ.
- ~ – ייקח אותנו לתיקיית הבית של המשתמש שאנו נמצאים בו כרגע.
- - – חוזר לתיקייה הקודמת שהייתם בה.
- . – משאיר אותנו באותו מקום.
- .. – מעביר אותנו תיקייה אחת אחורה.

לדוגמה:

```
cd /var/www/
```

pwd – מציג את מקום בו אנחנו נמצאים כרגע.

mv – פקודה להעברת קבצים ממקום למקום, בנוסף ניתן להשתמש בפקודה לצורך שינוי שם לקובץ או תיקייה.

cp – פקודה להעתקת קובץ ממקום למקום.

rm – פקודה למחיקת קובץ או תיקיות.

mkdir – פקודה ליצירת תיקייה.

rmdir – מחיקת תיקיות.

touch – פקודה ליצירת קובץ.

whoami – פקודה המציגה את המשתמש אליו אתם מחוברים כרגע.

echo – פקודה להדפסה על המסך.

history – פקודה להצגת היסטוריית פקודות שבוצעו ב-terminal, שימוש מאוד לאחר פריצה למכונה או ניתוח הנעשה במכונה.

clear – הפקודה מנקה את המסך.

passwd – פקודה לשינוי סיסמא של המשתמש הנוכחי.

- d – יצירת משתמש ללא סיסמא

```
passwd -d
```

wc – פקודה לספירת מילים או אותיות בקובץ, לפקודה 3 שדות בעת הדפסה:

1- כמות השורות
2- כמות המילים
3- כמות התווים

cat – מציג תוכן של קובץ

cut – הפקודה גוזרת תווים מהפלט ויוצרת קובץ חדש או מדפיסה אותם על המסך.

- c – איזה תווים לחתוך
- d – לפח איזה תו לבצע חיתוך לאורך
- f – איזה שדות להציג לאחר החיתוך

לדוגמה:

```
cut -c 3,4,5 hello.txt > file.txt
cut --delimiter=":" --fields="1" /etc/passwd
cut -d ":" -f "1,2,3" /etc/passwd
```

sort – פקודה שממיינת קבצים, פקודה שימושית כאשר רוצים למיין קבצי סיסמאות ולמחוק כפילויות.

```
sort hello.txt
sort -r hello.txt  [reverse]
```

```
sort -R hello.txt  [random]
```

uniq – מוחק כפילויות בקובץ, בדרך כלל משלבים אותו עם sort.

less/more – מציג תוכן של קובץ בצורה אינטראקטיבית.

head – הפקודה מציגה את 10 השורות הראשונות בקובץ שנבחר. כברירת מחדל, ניתן לשנות זאת באמצעות הפרמטר n- בצורה הבאה:

```
head -n 2
```

tail – מציג את 10 השורות האחרונות וניתן לשמור עליו

```
tail -f /var/log/syslog
tail -n 2
```

man – מפתחי לינוקס בדרך כלל כותבים מדריכים לפקודות שאותם הם בונים עבור המשתמש, פקודה זו תאפשר לנו לצפות במדריך

- k– חיפוש בכל המדריכים
- /show – מחפש מילה בכל המדריכים
- /usr/share/doc/

<u>כשמריצים פקודה יש לחשוב בצורה הבאה:</u>

[ארגומנטים של הפקודה] [אפשרויות של הפקודה] [הפקודה]

- **הפקודה** – זו בעצם הפקודה שאתם רוצים לבצע
- **אפשרויות** – לכל פקודה קיימים אפשרויות רבות שמאפשרים למשתמש להתאים את הפקודה לפעולה אותה הוא רוצה לבצע
- **ארגומנטים** – בדרך כלל מדובר בקבצים או מידע כלשהו שניתן לספק לפקודה.

כאשר אנו מבצעים פקודה על קובץ אשר מורכב מ-2 מילים או מכיל סימנים מיוחדים יש להשתמש באחת השיטות הבאות:

- או מרכאות או גרש
- אפשר גם \

```
touch "long file name"
touch long\ file\ name
```

תרגיל

1. עברו לתיקייה tmp/
2. תיצרו – 5 תיקיות ריקות בשם folder1, folder2, folder3, folder4, folder5.
3. היכנסו לתוך התיקייה folder1 ותיצרו 2 קבצים ריקים file1, file2.
4. **העתיקו** את הקבצים file1 ו-file2 לתיקייה folder2.
5. שנו שם לתיקייה folder2 ל-folder6.
6. **העבירו** את הקבצים מ-folder6 ל-folder5.
7. מחקו את כל התיקיות הריקות.
8. הציגו את ה-8 השורות האחרונות בקובץ /var/log/messages.
9. הציגו את ה-8 השורות הראשונות בקובץ /var/log/messages.
10. עברו לתיקייה /etc/ והציגו את תוכן הקובץ passwd.
11. עברו לתיקייה /etc/ וסננו את הקובץ group, הציגו את שמות הקבוצות בלבד עם הפקודה cut.

נקודה חשובה שיש לציין, הקבצים בLinux הם key sensitive ו-windows הוא לא זאת אומרת שהקבצים הבאים:

- Roman.txt
- roman.txt
- roman.TXT

ב-Linux מדובר בקבצים שונים, וב-windows מדובר באותו הקובץ. לכן עדיך להתייחס לשתי מערכות ההפעלה כאילו יש key sensitive וכתוב הכל באותיות קטנות.

קיצורי מקלדת

ב-Linux ישנם קיצורי מקלדת רבים שבאים לעזור לנו בעת העבודה עם ממשק הפקודה:

Shift + page up – לגלול במסך למעלה.

Shift + page down – לגלול במסך למטה.

Ctrl+l – למחוק את המסך

Ctrl+e – סוף השורה

Ctrl+a – תחילת השורה

Ctrl+u – למחוק את השורה אחורה מהמקום הנוכחי.

Ctrl+c – לבטל פעולה

Ctrl+z – לשלוח פקודה ל-background נושא של LPI 101.

מניפולציה של קלט ופלט

כמה מונחים חשובים שיש להכיר טרם נתחיל לבצע את המניפולציה על הפקודות:

- **stdin** – כל מה שהתוכנה מקבלת כקלט.
- **stdout** – כל מה שהתוכנה פולטת כפלט.
- **stderr** – שגיאות שמתקבלות בעת הפעלת התוכנה.

>	שליחת פלט + דריסת תוכן הקובץ
>>	שליחת פלט תוך כדי שמירה על התוכן הקיים (append)
&	להריץ את התוכנה ב-background, נושא שנלמד ב-LPI 101.
&&	אם הפעולה הקודמת בוצע בהצלחה בצע גם את הפעולה הנוספת.
||	בצע פקודה שנייה רק אם פקודה ראשונה לא הצליחה
;	הרצת פקודה נוספת בין אם הפעולה הראשונה הצליחה ובין אם לא.
<	קבלת קלט מקובץ מה שנראה כך wc < roman.txt
|	מעבר של stdout ל-stdin של פקודה נוספת.

חיפוש בכל הקבצים במקום הנוכחי

```
grep "roman" *
```

חיפוש רקורסיבי בתת קבצים

```
grep -rni "roman" *
```

tee - פקודה ששולחת את ה-stdout גם לקובץ וגם למסך

```
ls | tee roman.txt
```

לא כל פקודה מקבלת stdin באמצעות "|" לכן ישנו משתנה עזר שמוסיף stdin לפקודה הנקרא xargs:

```
ls | xargs echo
```

ההבדל בין

```
wc file.txt
wc 0< file.txt
```

הוא שבפלט השני הפקודה לא יודעת מה שם הקובץ שאיתו היא עובדת

יצירת קובץ ריק

```
> file.txt
```

עבודה עם stderr

```
ls 1> ls.txt     [redirect correct]
ls 2> ls.txt     [redirect errors]
ls >> ls.txt     [append]
```

ניתן לשלב ביניהם:

```
ls bob 1>> results.txt 2>> errors.txt
ls bob 1> results.txt 2> errors.txt
ls bob 1> results.txt 2>&1
```

אם אני רוצה להעביר גם Output וגם error קיים קיצור &>

```
ls bob &> logs.txt
```

כאשר יש שגיאות בהדפסה אפשר לזרוק אותם ל-/dev/null

```
grep david /etc/* 2> /dev/null
```

תרגיל

1. הפקודה **ifconfig** מציגה את כתובת ה-IP של המכונה שלכם, באמצעות סינון הציגו את כתובת ה-IP בלבד.
2. הפקודה find מאפשרת לכם לחפש במערכת, בצעו את הפקודה:

```
find / -name "messages"
```

העבירו את כל השגיאות לקבוץ error.log ואת התוצאה לקובץ success.log

3. בצעו שוב את הפקודה מסעיף 2 והפעם הציגו על המסך רק את התוצאות התקינות, אין להציג שגיאות ב-stdout.

4. בצעו את הפקודה והבאה

```
cat /etc/passwd
```

והציגו את כל המשתמשים במערכת בשורה אחת!

Wildcard and Globing

כאשר אנו מעוניינים לחפש קבצים בלינוקס ואנו לא בטוחים מה שם הקובץ המדויק אנו יכולים לבצע חיפוש מותאם תבניות. חיפוש זה נקרא Wildcard או Globing.

קיימים 4 תווים שמאפשרים לכם לעשות חיפוש וכמעט כל פקודת לינוקס תומכת בהם:

- ? – מחליף תו
- * – מחליף את כל התווים.
- [] – רשימת תווים אפשריים עבור תו בודד.
- **{..} – תווך חיפוש (לא חלק מה-globing).**

לדוגמה, על מנת להציג את כל הקבצים שמכילים את התווים abc נשתמש בכוכבית כך:

ls abc*

תרגיל

צרו כ-100 קבצים באמצעות הפקודה touch ותווים רנדומליים עם רווחים, לא ללחוץ בטעות על סימנים מיוחדים או על המקש capsLock לדוגמה:

```
touch a23{1..10}
touch b{a..z}
touch hasd sad7 sd asd asd 7gasd a23b asd 8hsad has8d as8dh asads as vacx
asc a…
```

לאחר מכן בצעו את התרגיל הבא:

1. הציגו את כל הקבצים שמכילים 3 תווים בלבד.
2. הציגו את כל הקבצים שמתחילים בתו a.
3. הציגו את כל הקבצים שמתחילים בתו a לאחריו יש 2 תווים ואז תו כלשהו מהתווים a עד k.
4. הציגו את כל הקבצים שמכילים את התו h
5. הציגו את כל הקבצים שמתחילים בa,b או c ומסתיימים באחד התווים מ-g עד z.

ביטוי רגולרי והפקודה egrep

בעולם הלינוקס הקובץ הנפוץ ביותר הינו קובץ טקסט והוא משמש את המערכת לקבצי לוגים, לקבצי הגדרות, לסקריפטים ולעוד שימושים רבים נוספים. הצורך בחיפוש בקבצי טקסט ומציאת תוכן הינו הכרחי וכאן הביטוי הרגולרי נכנס לתמונה.

ביטוי רגולרי הינו תבנית המאפשרת לנו לאתר תוכן מסוים בקובץ המתאים לתבנית שהגדרנו.

התבניות מורכבות מהתווים הבאים:

- () – הגדרת תבנית כקבוצה, שרשור תבניות.
- {} – כמות הפעמים יופיע רצף מסוים שמוגדר באמצעות סוגריים עגולים ().
- [] – מספר אופציות לתו בודד
- . – תו כלשהו (דומה ל-? ב-wildcards)
- * – התו הקודם מופיע 0 או יותר פעמים כך שהצירוף *. מציין כל דבר כמו (* ב-wildcards)
- + – התו הקודם מופיע פעם 1 או יותר, דומה מאוד ל-*
- ^ – השורה צריכה להתחיל בתו מסוים, ואם התו מופיע בתוך סוגריים מרובעים הוא מסמן היפוך.
- $ – השורה צריכה להסתיים בתו מסוים

הפקודה grep:

- egrep/-E – הפעלת extended regex, ללא הפקודה לא ניתן להשתמש בחלק מהאפשרויות כגון [].
- o- – להציג רק את ההתאמה לתבנית אחרת כל השורה תוצג.
- r- – חיפוש רקורסיבי בתוך כל הקבצים בתיקייה ובתתי תיקיות.
- n – הצגת מספר השורה.

- i- – ביטול תווים רגישים (case sensitive).

לדוגמה על מנת להציג את הכתובת ה-IP של המחשב שלכם השתמשו בפקודה:

```
ifconfig > ip
grep -Eo "([0-9]{1,3}\.){3}[0-9]{1,3}" ip
```

שימו לב שאם תכתבו את הפקודה בלי o– כל השורה תוצג, לפעמים זה בדיוק מה שאנחנו רוצים לדוגמה אם נרצה להציג את כל התיקיות נשתמש בפקודה הבאה:

```
ls -l | grep ^d
```

תרגיל

1. הציגו את כל כתובות ה-IP בקובץ /var/log/messages ושימרו אותם בקובץ /tmp/ip_log.
2. הציגו את כל הקבצים בתיקייה הנוכחים באמצעות ls ו-grep.
3. הציגו את כל כתובות ה-Mac Address (במידה ואינכם מכירים את התבנית יש לחפש בגוגל) בקובץ /var/log/messages.
4. הציגו את כל השורות שמופיע בהם המילה error או Error בקובץ /var/log/messages יש להציג גם את מספר השורה.