**שאלות מכווינות- לינוקס**

**להלן שאלות מכווינות רגע לפני שתתחילו לשחק במכונת לינוקס ולהתנסות בפקודות בעצמכם.**

1. **מה המשמעות של מערכת הפעלה OpenSource?**

מ"ה קוד-פתוח היא סוג מ"ה שהקוד שלה פתוח לציבור (חלק יאמרו שקוד פתוח אמיתי יהיה קוד-פתוח חינם) וכל אחד יכול לצפות, לשנות ולהפיץ את אותו קוד. כאשר הקוד הפתוח הוא קוד של מ"ה, משתמשים מוזמנים לשנות ולהפיץ שינויים שלהם למ"ה, מה שגורם לשיתוף פעולה בין משתמשים וייזום של שיפורים אינדיבידואלים למ"ה.

1. **מה היא הפצה בלינוקס?**

הפצה (distribution) של לינוקס הינה מ"ה הבנויה מסביב לקרנל לינוקס. הפצה של לינוקס בדר"כ תכלול את הקרנל ואוסף חבילות למ"ה. אוסף החבילות יכול לכלול כלי עזר למערכת ולמשתמש ודברים אחרים כמו ספריות, פקודות תוכנות ועוד. הפצות שונות של לינוקס נוצרות (רוב הזמן) ע"י משתמשי לינוקס בשביל משתמשי לינוקס לשימושים שונים וקהלי יעד שונים, לדוגמא ubuntu נוצר לשימוש יום יומי של משתמש עם ידע יחסית נמוך במחשבים, redhat נוצר למשתמשים יותר רציניים, kali linux נוצר בשביל זיהוי ובדיקות חדירה, ועוד…

1. **מה הוא משתמש root בלינוקס? במה הוא שונה משאר המשתמשים?**

משתמש root הוא ה-administrator של מ"ה, משתמש בעל כל ההרשאות שיש למ"ה, ויכול לעשות כל מה שמשתמש יכול לעשות. כל הפעולות שמשתמש הroot יכול לעשות לא ניתנים למשתמשים רגילים מכיוון שמשתמש הroot יכול לעשות דברים כמו לראות את כל המידע שיש על מחשב ולהרוס את המחשב, ויכול להיות שיש מידע קריטי על המחשב שאנחנו לא רוצים שכל משתמש יראה ו/או יהיה לו את האפשרות להרוס. כאשר משתמשים רגילים רוצים לעשות משהו שאין להם את ההרשאה, הם יכולים "לעקוף" את ההרשאה הזאת ע"י sudo, שאומר ל-superuser לעשות את הפקודה שרשומה אחרי (בשביל לעשות את הפקודה הזאת המשתמש כן צריך הרשאה).

1. **מה מציינת פקודת הpwd בלינוקס?**

pwd (print working directory)d הינה פקודה בלינוקס המראה את התיקייה הנוכחית שבה המשתמש נמצא.

1. **איך בנוי מנגנון ניהול ההרשאות בלינוקס?**

בלינוקס לכל קובץ יש שלושה הרשאות:

* Read
* Write
* Execute

ולכל קובץ יש שלושה קבוצות:

* Owner
* Group
* Others

כלומר לכל קבוצה של משתמשים יכול להיות את שלושת ההרשאות הקודמות.

לכתיבת ההרשאות יש קונבנציה, לדוגמא, rwxr--r-x בעצם אומר:

* למנהל יש הרשאות של קריאה, כתיבה והרצה
* לקבוצה יש הרשאות של קריאה בלבד
* לאחרים יש הרשאות של קריאה והרצה בלבד

1. **לפניך הפלט של הפקודה ls -l:** 
   1. **d rwx r-x r-x. 18 Sigit Sigit 4096 2010-01-14 08:20 Documents   
      ציין לגבי כל חלק מה הוא מציין.**

d - סוג קובץ, במקרה הזה זה directory

rwx - הרשאות של המנהל, במקרה הזה read write execute

r-x - הרשאות של הקבוצה, במקרה הזה read execute

r-x - הרשאות של אחרים, במקרה הזה read execute

18 - ה-link count, מציג כמה hard links יש לקובץ.

Sigit - מנהל הקובץ

Sigit - הקבוצה הקשורה לקובץ

4096 - גודל הקובץ בבתים

2010-01-14 08:20 - הזמן האחרון שהקובץ השתנה

Documents - שם הקובץ

-

1. **מהו הservice הראשון שרץ בעליית המחשב בלינוקס? ומתי הוא מסתיים?**

השירות הראשון שרץ בעליית המחשב בלינוקס הוא Init, והוא מסתיים כאשר המערכת נכבת. init הוא גם הancestor של כל שאר התהליכים.

1. **הסבירו בהרחבה מהם הserviceים הבאים:**
   1. **Sshd (SSH Daemon)d** - השירות שרץ ומאפשר למשתמשים אחרים להתחבר למכונה שלך באמצעות ssh. השירות מספק תקשרות מוצפנת ובטוחה בין שני משתמשים. מקשיב לחיבורים מקליינטים על פורט 22.
   2. **Ntpd (Network Time Protocol Daemon)d** - שירות הדואג שזמן מ"ה יהיה מסונכרן עם שרתי NTP מיועדים בעזרת המידע שהמשתמש מקבל מהם, מה שמבטיח שזמן מ"ה הוא נכון.
   3. **Syslog (מה השירות כותב ולאילו קבצים)** - שירות האחראי על ניהול הודעות לוגים המגיעים ממקומות שונים במ"ה. ללוגים בדר"כ יש פורמט של:

* timestamp hostname application msg

והם בדר"כ ב-var/log/syslog או var/log/messages

1. **אתה נמצא מקומית (Console) על שרת לינוקס עם הפצת Red Hat. אתה שם לב שאין GUI אלא רק Shell טקסטואלי, שמכיל את הפלט של הפקודה htop.** 
   1. **מה ההבדל בין GUI ל TUI?**

TUI (Textual User Interface)i - אינטרפייס בשביל משתמש המשתמש בטקסט בשביל לייצג מידע ואלמנטים על המסך ומאפשר אינטראקציה עם אותם אלמנטים.

GUI (Graphical User Interface)i - אינטרפייס בשביל משתמש המשתמש באלמנטים גרפים בשביל לייצג מידע ואלמנטים, כמו אייקונים, כפתורים ועוד…

ההבדלים העיקריים הם:

|  | TUI | GUI |
| --- | --- | --- |
| ויזואליזציה | מראה הכל דרך טקסט, מה שנותן קצת אפשרויות למה להראות ואיך. | יכול תיאורתית להראות הכל בעזרת גרפיקה. |
| לימוד | קשה בהתחלה להבין איך משתמשים, אך בסוף יכול להיות מאוד נוח | מאוד טינטואטיבי למשתמש |
| משאבים | לוקח פחות משאבים | לוקח יותר משאבים |
| שימוש | משתמשים בדר"כ במקומות שהגרפיקה פחות משנה ו/או במקומות עם פחות משאבים, כמו שרתים ובטרמינל. | משתמשים בדר"כ במקומות חברותיים למשתמש ומחשבים אישיים לשימוש אישי כמו מחשב נייח. נשתמש בGUI גם בתוכנות שצריכות אינטרפייס גראפי, כמו פוטושופ. |

* 1. **הסבר מהו runlevel ובאיזה runlevel השרת נמצא?**

runlevel-ים מגדירים תצורות שונות שבא המערכת עובדת, מה הקונפיגורציות שלה ומה רץ בכל תצורה. מכיוון שאין GUI אלה רק shell טקסטואלי אפשר להניך שהrunlevel הוא 3.

* 1. **מה תפקידה של הפקודה htop?**

htop היא מראה תהליכים בצורה אינטרקטיבית. תפקידה היא לצפות בכל התהליכים הרצים על המערכת, להראות מידע על אותם תהליכים ונותנת לעשות דברים על אותם תהליכים כמו להרוג אותם. משתמשים בhtop כמו הפקודה top, רק שhtop זה נוח וקריא יותר.

1. **מה ההבדל בין Interpreter ל-Compiler?**

| Compiler | Interpreter |  |
| --- | --- | --- |
| מתרגם את כל קוד המקור לקוד מכונה לפני הרצה. בדר"כ מביא קובץ ריצה שונה מהקוד שאפשר להריץ. | בודק את הקוד שורה שורה ומריץ אותו בין רגע. מבין את הקוד תוך כדי ריצה. | הרצה |
| בודקים בשביל שגיאות לפני ההרצה. בדר"כ הקומפילציה תפסיק עם נמצא שגיאה ולא יצא קובץ הרצה. | מפסיק את ההרצה מיד לאחר שנמצא שגיאה, מביא משוב מיד. | טיפול בשגיאות |
| יותר קשה לעשות דיבאגינג מכיוון שהשגיאות ידווחו רק לאחר ניסיון הקומפילציה. | יותר קל לעשות דיבאגינג מכיוון שהשגיאות נמצאות כאשר נריץ את הפקודה הרצויה. | דיבאגינג |
| יותר יעיל | פחות יעיל | יעילות |

1. **מהו bash?**

Bourne Again SHell (Bash)i - נכתב ע"י gnu, הBash הוא הShell הנפוץ ביותר לינוקס. Bash מביא CLI בשביל פקודות וסקריפטינג. Bash הוא כלי קריטי בשביל מ"ה לינוקס, מכיוון שבלינוקס אפשר לעשות הכל דרך הshell. אפשר להשתמש בBash גם כinterpreter.

1. **מה מאפשר הכלי putty**

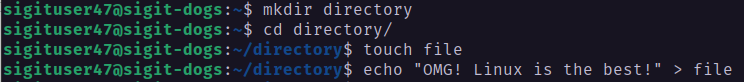
PuTTY הוא אמולטור טרמינל התומך בפרוטוקולים רבים כמו SSH, SCP, Telnet ועוד… התוכנה מאפשרת לניהול מערכות מחשבים מרחוק בשל הכלים הרבים שלה והחיבור הבטוח שמביאה.

1. **הסבר על הפקודות הבאות:**
   1. **Vim** - עורך טקסט, פופולרי בזכות הקיצורים שלו ויכולות הקונפיגורציה שלו. משתמשים בוא בעיקר מפתחים.
   2. **Touch** - משתמשים בזה בעיקר ליצור קבצים חדשים רקים, ואפשר להשתמש בזה גם בשביל לשנות את זמן המודיפיקציה של קובץ לעכשיו.
   3. **Cat** - קיצור של Concatenate, משתמשים בזה בשביל לראות את הטקסט של קובץ מסויים או כמה קבצים. אפשר להשתמש בזה בשביל להכניס את אותו לקובץ אחר.
   4. **Mkdir** - יוצר directories.
   5. **Rm** - מוחק קבצים וdirectories.
   6. **ll** - לראות מידע על כל הקבצים בתיקייה, עם פורמט ספציפי ופרטים מיוחדים כמו ההרשאות
   7. **ifconfig** - מציג מידע על כל ממשקי הרשת של המערכת, ונותן להגדיר אותם.
2. **ענו על השאלות הבאות:**
   1. **מהו הקובץ /etc/resolv.conf?** - קובץ משומש ע"י מ"ה (ספציפית רסולבר הDNS משתמש בא) בשביל למצוא שרתי DNS.
   2. **מהו הקובץ /etc/network?**  - (אצלי זה תיקייה) משתמשים בא בשביל לאחסן קבצי קונפיגורציה הקשורים לרשת. לדוגמא, בתיקיות if-up.d וif-down.d יש סקריפטים שירוצו כאשר רשת עולה ויורדת, בהתאמה, ובinterfaces יש הגדרות לאינטרפייסים של רשת, כמו כתובות IP
   3. .
   4. **הסבר מהם iptables? מתי נשתמש? מה המצב הדיפולטי?** - iptables היא תוכנה המאפשרת למנהל המערכת להגדיר את הפילטרים לקבלת פאקטות של הfirewall של הליבה. הפילטרים מאורגנים לtables, ולכל אחד מהם יש chains שקובעים איך פאקטות יקובלו. המנהל יכול לשנות את הtables הדיפולטים (שהם INPUT, OUTPUT, וFORWARD), או להוסיף עוד ולהגדיר אותם. לtables הדיפולטים אין חוקים שיקבעו איך הפאקטות יעברו. יכול להיות שהפצות שונות החליטו כן להגדיר חוקים דיפולטים. נשתמש בiptables שאנחנו רוצים להגדיר איך פאקטות ינוהלו במערכת לינוקס שלנו בשביל לשלוט בתעבורת הרשת.

תרגול מעשי לינוקס

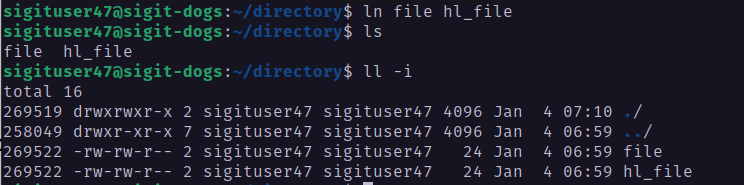
תיקיות וקבצים

1. צור תיקייה חדשה בשם directory.
2. עבור לתיקייה החדשה שיצרת.
3. צור קובץ חדש בשם file.
4. כתוב לקובץ "OMG! Linux is the best!!!".



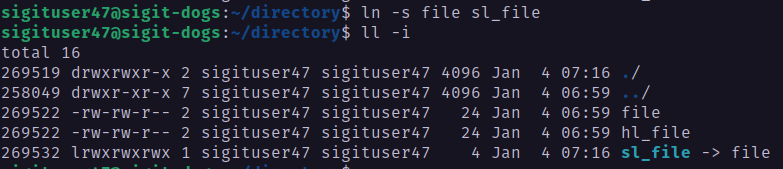
1. הסבר מהו hard link וצור אחד כזה לקובץ file.

מצביע לinode שקיים כבר בדיסק. יכול להיות כמה כאלה, ושנמחק את הקובץ שאליו מצביע, לא יקרה כלום מכיוון שהמקום נשאר תפוס בדיסק.



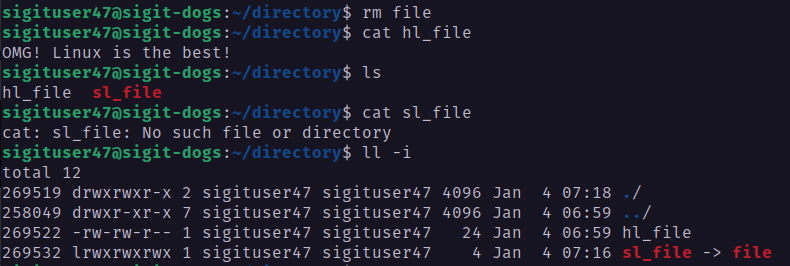
1. הסבר מהו softlink וצור אחד כזה לקובץ file.

קובץ שונה מהמקור שבו הpath לקובץ הרצוי. יש לאותו קובץ inode משלו, ושנמחק את הקובץ שאליו מצביע, הsoftlink גם ימות.



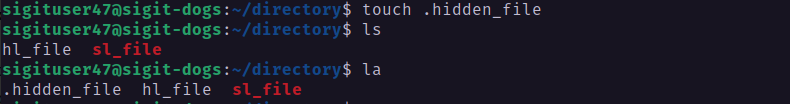
1. מחק את הקובץ file ונסה להגיע אליו דרך כל אחד מהלינקים. הסבר איך מחיקת הקובץ השפיעה על הלינקים.

ה-hard link נשאר כפי שהוא, מכיוון שהוא מצביע לאותו מקום בדיסק כמו file, וה-soft link מת, מכיוון שהקובץ שאליו הצביע נמחק.

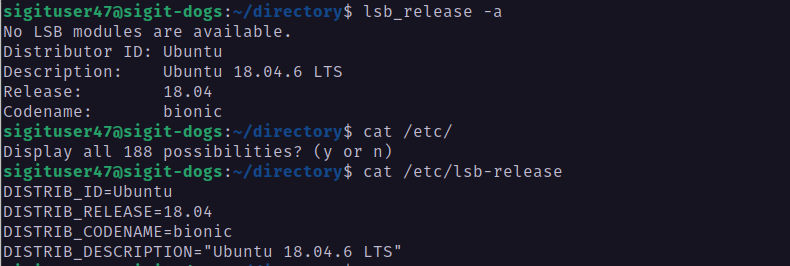


1. הסבר מהו קובץ מוסתר וצור אחד כזה. איך אוכל לראות אותו?

קובץ מוסתר הוא קובץ ששמו מתחיל ב-'.', הם דיפולטים לא נראים כאשר נשתמש בפקודות כמו ls, ומשתמשים בהם בעיקר לקבצים שלא נרצה לשנות אותם לעיתים קרובות.



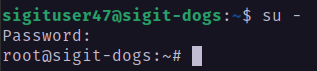
1. מהי מערכת ההפעלה של המכונה? איפה ואיך בדקתם?



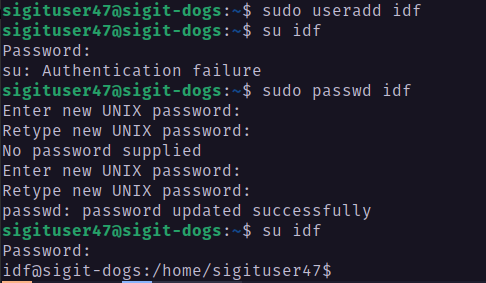
משתמשים והרשאות

1. עברו מsigituser למשתמש root באמצעות הפקודה su -. הסבירו מה עושה הפקודה su ומה המשמעות של ה "-".

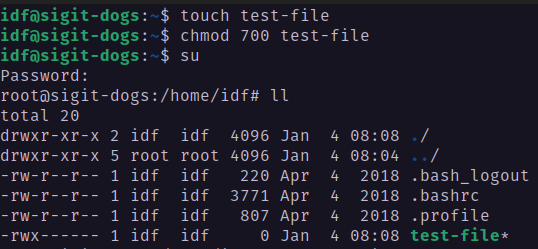
su נותן למשתמשים לשנות את הID שלהם או להפוך לsuperuser. משמעות ה"-" היא לשמור על environment variables כמו שהם היו נכנסים כרגיל.



1. צרו משתמש חדש בשם idf
2. קבעו סיסמא חדשה למשתמש הזה
3. תעברו אל המשתמש הזה



1. צרו קובץ בשם test-file בתיקיית הבית של המשתמש הזה ותעשו שמשתמשים אחרים לא יוכלו לקרוא את הקובץ הזה (other)sud
2. חזרו למשתמש root



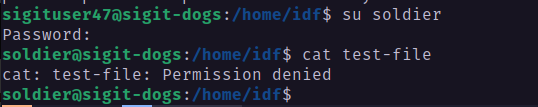
1. פרטו על מצב ההרשאות של הקובץ

למשתמש (idf) יש את כל ההרשאות (read, write, execute) אל הקובץ, ולקבוצה idf ולכל השאר אין שום הרשאות.

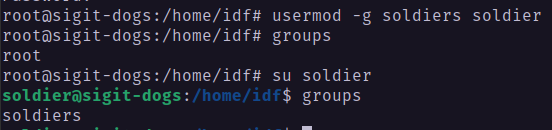
1. צרו עוד משתמש בשם soldier
2. \*/צרו קבוצה בשם soldiers



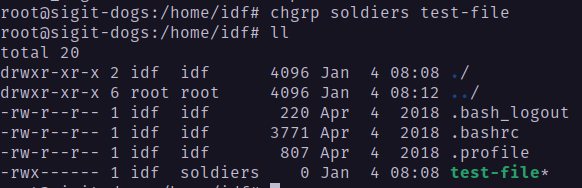
1. עברו למשתמש soldier ותנסו לקרוא את הקובץ הזה



1. שייכו את המשתמש soldier לקבוצה soldiers



1. הגדירו שהקבוצה שתהיה בבעלות על הקובץ היא soldiers

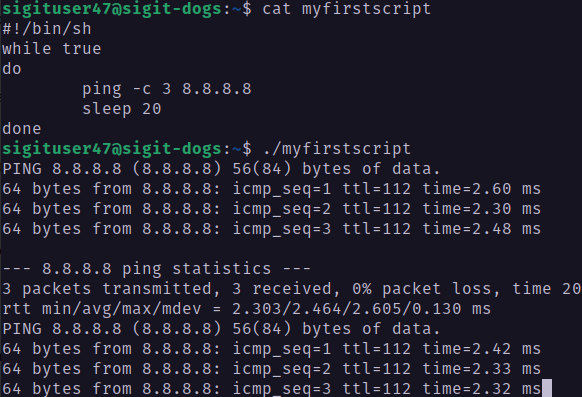


סקריטיפם bash

1. צרו קובץ בשם myfisrtscript ותנו לו הרשאות הרצה



1. כתבו סקריפט שכל עשרים שניות נותן שלושה פינגים (ping) ל8.8.8.8 ותריצו אותו (ללא cron)

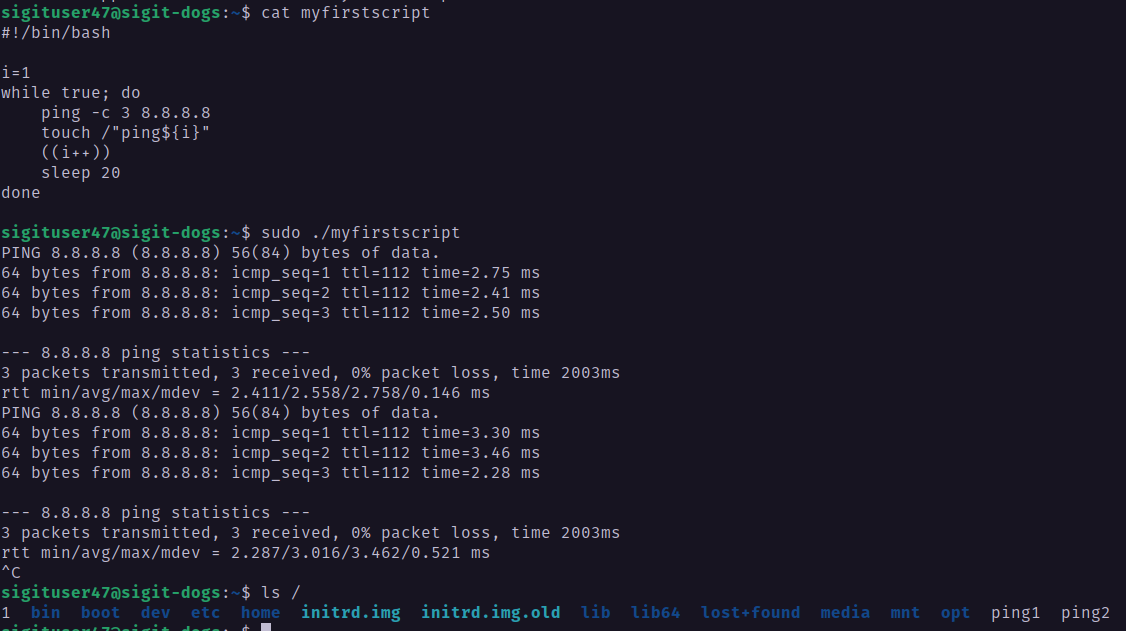


1. כעת, ערכו את הסקריפט כך שכל פינג שמתבצע ייצור קובץ חדש בתיקיית הroot "/" בקונבנציה הבאה -

/ping1

/ping 2

וכן הלאה...

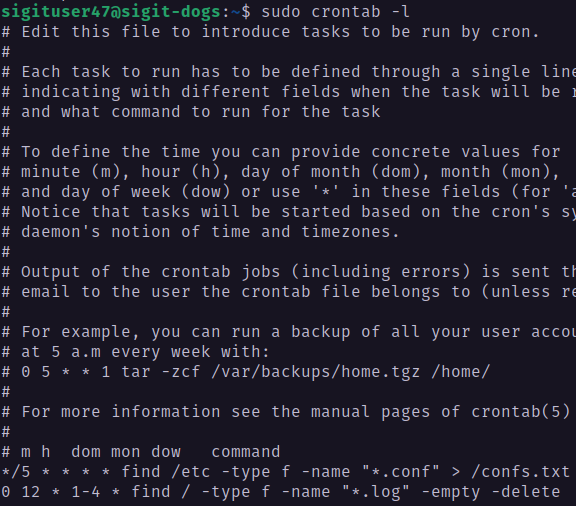


Cron

1. צרו crontab שיוצר רשימה של שמות כל הקבצים עם סיומת conf מהתיקייה /etc ומכניס אותם לקובץ /root/confs.txt. קבעו שהוא ירוץ כל 5 דקות ויעשה overwrite לקובץ.
2. צרו crontab שמוחק את כל הקבצים הריקים בעלי הסיומת log. קבעו שהוא ירוץ כל יום ב12 במשך ארבעת החודשים הבאים.

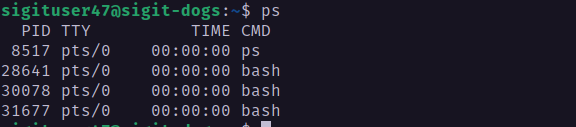


1. הציגו אילו cronים רצים כעת ברקע.



Proccessים:

1. מצא את כל התהליכים שהשרת מריץ.



1. באמצעות איזו פקודה תוכל "להרוג" תהליך.

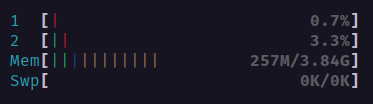
kill

1. מזה PID ואיפה נמצא אותו?

process ID, מחובר אוטומטית לכל תהליך כאשר נוצר, ולכל תהליך יהיה PID שונה.

יהיה אפשר למצוא אותו בעזרת ps … | grep …, או בעזרת pidof

1. מצא את אחוז הניצול משאבים של השרת שלך (CPU ו RAM).



1. בחר 5 תהליכים שרצים על השרת שלך והסבר עליהם.

* /lib/systemd/systemd-timesyncd - שירות המסנכרן את שעון המערכת דרך הרשת בעזרת קליינט SNTP.
* lib/systemd/systemd-networkd - שירות האחראי על קונפיגורציה וניהול הרשת. מנהל דברים כמו אינטרפייסים של הרשת, כתובות IP ועוד… השירות גם מזהה ממשקי רשת המתגלים, ומנהל אותם. 
* lib/systemd/systemd-resolved - מנהל DNS וכתובות IP בשביל ממשקים מקומיים ומביא שמות רשת לממשקים מקומיים.
* usr/sbin/rsyslogd -n - מתחיל את השירות rsyslog, שזה תוכנת קוד פתוח בשביל להעביר לוגים דרך הרשת.
* lib/systemd/systemd-resolved - שירות המנהל התחברויות של משתמשים. השירות עוקב אחרי המשתמשים, התהליכים שלהם, מה קורה שהמשתמש מתנתק ושליחת הודעות קיר (הודעה לכלל המשתמשים).

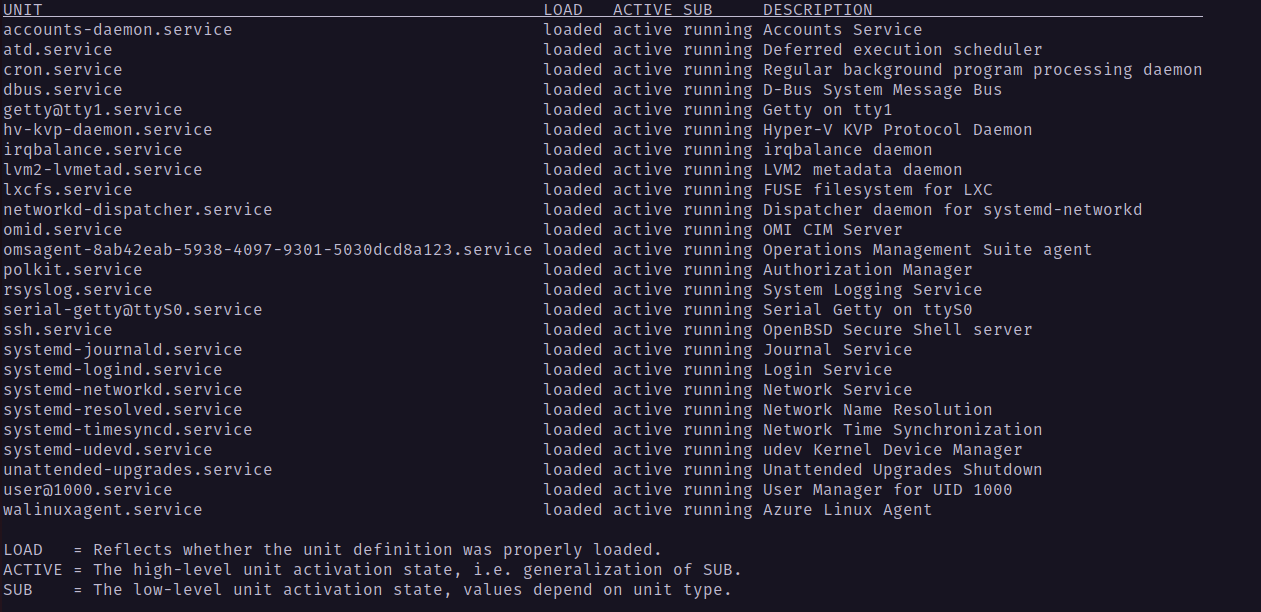
Serviceים:

1. רסט את service התקשורת בשרת שלך (דגש על **ריסוט** ולא כיבוי) מה קורה בריסוט הservice?



לא קרה כלום,אבל זה אמור לעשות ריסטרט לתקשורת.

1. הצג את כל הserviceים שיש על השרת ואת המצב שלהם



בהצלחה!