שיעור 6 – סקאפי

**כותבים את כלי CMD מחדש**

בתרגיל זה נעבוד עם אותם הכלים שתרגלנו בתרגיל הכיתה, אך נשתמש בהם בקובץ פייתון. במהלך התרגיל נשכפל את כלי ה-CMD שכולנו מכירים – nslookup, ping, tracert, והכל נעשה בעצמנו!

**סקאפי בקובץ py – איך מתחילים?**

1. כמו תמיד כאשר אנו עובדים בקבצים, כתיבת הקוד מתבצעת רק לאחר שכבר בדקנו מה אנחנו רוצים לעשות באינטרפטר / בטרמינל של סקאפי.
2. פתחו קובץ פייתון חדש ובצעו ייבוא לסקאפי בעזרת הפקודה:

from scapy.all import \*

**\* שימו לב, בעבר למדנו שייבוא בצורה הזאת import \* הוא מסוכן. שימוש בסקאפי הוא אחת הפעמים היחידות בהם נשתמש בייבוא זה. זאת מכיוון שבשימוש בסקאפי אנו זקוקים להרבה מאוד אובייקטים וקבועים, ואם לא נייבא אותם ישירות הקוד שלנו יתארך מאוד (scapy.IP במקום IP וכו'). עם זאת יש להיזהר כרגיל מליצור משתנים עם שמות שעשויים להיות תפוסים כבר ע"י סקאפי (כמו שמות של פרוטוקולים).**

1. כעת נכתוב סקריפט לדוגמא אשר מבצע שאילתת PING כפי שעשינו בתרגיל הכיתה.
2. צרו את ההודעה השלמה אשר ברצוננו לשלוח כך:

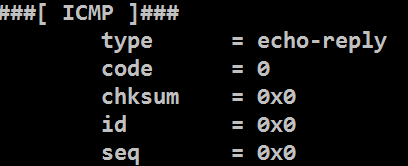
fullmsg = Ether() / IP(dst=”google.com”) / ICMP()

1. כעת שלחו את ההודעה בעזרת srp1:

ans = srp1(fullmsg)

1. אם תרצו שסקאפי לא ידפיס שום דבר עבורכם (עדיף), השתמשו בשורה הבאה:

ans = srp1(fullmsg, verbose=0)

1. כעת הדפיסו את התשובה שקיבלתם למסך בעזרת show().
2. הריצו את הסקריפט ובדקו שאכן הוא מדפיס את תשובת Echo reply למסך כך:
3. זהו, כעת אתם יודעים כיצד להשתמש בסקאפי בקבצים!

**\* אם נתקלתם בבעיות בשליחת / קבלת תשובה, בדקו את הבעיות הנפוצות המופיעות בתרגיל הכיתה.**

1. **כתיבת nslookup**

עליכם לכתוב כלי אשר מבצע שאילתת DNS בדיוק כמו הכלי nslookup. הכלי יבקש מהמשתמש להקיש דומיין מבוקש, וידפיס את ה-IP שהוא מצא.

יש להשתמש באותן פקודות בהן השתמשתם בטרמינל סקאפי בתרגיל הכיתה – הכל זהה!

שימו לב:

* יש להדפיס את ה-IP בלבד (עליכם לגשת לשדה הספציפי שמחזיק את ה-IP).
* יש לשמור את כל הנתונים (כתובת שרת ה-DNS, פורט וכו') בקבועים.
* יש מספר דרכים בפייתון לבצע המרה של דומיין ל-IP, במקרה הזה – אתם נדרשים ליצור ולשלוח הודעת בקשת DNS בסקאפי לשרת DNS.

1. **כתיבת כלי Ping**עליכם לכתוב כלי אשר יבקש מהמשתמש להקיש דומיין נבחר, ויבצע 3 בקשות Ping אליו. הכלי יציג את הזמן של הגעת כל תשובה (במילישניות), ואת הזמן הממוצע של שלושתן.   
   למעשה על הכלי לבצע ICMP Echo-request לאתר נבחר, ולמדוד את הזמן שלוקח לתשובה לחזור. כדי למדוד זמן, עליכם לשמור את השעה לפני שליחת ההודעה והשעה ברגע שהיא חוזרת, ולמצוא את ההפרש ביניהם. חפשו בגוגל מידע על ספריית timedelta.  
   שימו לב:

* על הכלי להיראות דומה לכלי ping האמיתי. יש להדפיס שורה על כל תשובה שחוזרת, זמן החזרה שלה, ובסוף סיכום של ממוצע הזמנים.

1. **כתיבת tracert**עליכם לכתוב כלי אשר מבצע tracert לכתובת מסוימת. אם שכחתם מה זה tracert, סורו חזרה לשיעור 12 בסמסטר א'. המשתמש יקיש דומיין נבחר, והתוכנה תציג את כל התחנות שבדרך לדומיין הזה בצורה כזאת:

**1: 10.0.0.13  
2: 100.10.1.1  
3: 142.12.31.10**

**….**

**\*** לעיתים חלק התחנות בדרך לא מגיבות (זה בסדר). כדי להתמודד עם זה מומלץ לקרוא על הפרמטר timeout שאפשר להעביר לפונקציה srp1.

**יש להגיש את 3 קבצי py בקובץ ZIP בשמות – nslookup.py, tracert.py, ping.py**

1. **בונוס – קבלת פרמטרים דרך argument**

כל כלי CMD לא מבקשים קלט מהמשתמש אלא מקבלים אותו דרך ארגומנטים (Command line arguments), כך שניתן למשל להריץ **tracert google.com** בשורה אחת.

שפרו את כל הסקריפטים כך שיעבדו בצורה הזאת (היעזרו בגוגל כמובן).