עבודה עם ספריית CheckPointAPI

## רעיון כללי של הספריה

הספרייה עצמה מבוססת על ממשק ה API שצ'קפוינט מציעים החל מגרסת R80 שלהם. בהתאם לכך הספרייה לא תעבוד עם גרסאות נמוכות יותר (ואו גבוהות יותר בהתאם למה שאנשי צ'קפוינט היקרים יחליטו לעשות בעדכון הבא).

הרעיון הכללי שלה היא להיות כמה שיותר מובנת, לספק פונקציות שימושיות להשגת מידע מהCMA ולהיות מודולרית, הספרייה לחלוטין לא שלמה ויש עוד הרבה דברים שניתן להוסיף אליה, איך היא ממחישה את הרעיון הבסיסי של לקיחת נתונים והוצאתם דרך ממשק תכנותי כזה או אחר לשימוש אישי של הצוות.

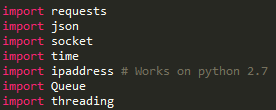
אם יבוא היום ומגן סייבר אמיץ או כל אדם אחר ייראה לנכון את הצורך להוסיף לה פונקציונליות, GO FOR IT. רק תכתוב דוקומנטציה ברורה !

**דברים שחשוב לדעת !**

דאגתי לכתוב לכל הספרייה דוקומנטציה ברמת הקוד. זה לקח הרבה זמן, אבל זה היה שווה את זה. ניתן לגמרי לעשות help על כל פונקציה בקוד ולמצוא הסבר סביר לדברים שהיא עושה (פלוס מינוס טעוית כתיב של ישראלי ממוצע). מקווה שבעזרת המדריך הזה והדוקומנטציה אצליח לתת מענה על רוב הבעיות (חוץ מהבאגים שעוד לא גיליתי)

**חלוקה מבחינת הקוד**

ספריות :



**Requests** – מאפשר לי לבצע קריאות http לכיוון הCMA בצורה נוחה. חלק מהפצת אנקונדה (שלדעתי יש לעמית בתיקייה האישית)

**Json** – פרסור של json שמגיע מכיוון הCMA

**Socket** – כשכתבתי את המדריך נזכרתי שאני לא צריך באמת את הספרייה הזאת והורדתי אותה, אבל לשימוש עתידי כזה או אחר היא מאפשרת לבצע שאילתות DNS (שווה ערך לnslookup) ולכן היא לחלוטין שימושית.

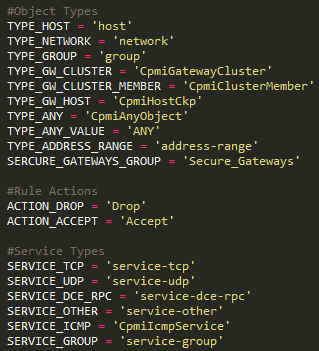
**Time** – what time is it ? ITS ADVENTURE TIME . מאפשר לי לעצור את הזמן או יותר נכון את הthread של התוכנה ובכך לעצור לכמה שניות במקומות מסויימים שיש בדבר צורך.

**Ipaddresss** – מאפשרת לי לבצע ניתוחים של כתובות ip השוואות ובדיקות כאלו ואחרות

Queue – מבנה נתונים (יענו תור) שהשימוש העיקרי שלו זה לביצוע פעולות אסינכרוניות בצורה נוחה

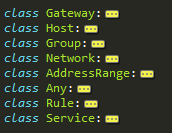
**Threading** – מי אמר אסינכרוני ולא קיבל ?

**קבועים (זה חשוב !!!)**



רוב הקבועים הם פרטים מזהים אשר לקחתי מהממשק של Checkpoint כשבדקתי את הפלטים שהוא מחזיר. הרשימה לא שלמה ואני בטוח שעם עוד התעסקות עם הממשק יהיה אפשר להכניס עוד הרבה כאלו, רובם דיי מסבירים את עצמם ובחיפוש קצר בקוד ניתן להבין מה השימוש שלהם.

**מחלקות של מבני נתונים בCMA**



בצורה מאוד הגיונית, המחלקות מתחקות אחרי מבני התונים שCheckpoint מציע. רוב הפונקציות של הספרייה יחזירו את אחד ממבני הנתונים האלה אם לא רשימה שלו, היחידי שאני מוצא לנכון להסביר אותו זה Any והוא נועד לשמש מקרים שיש Any בחוק מסויים ורציתי לייצג אותו בצורה נוחה יותר לעין מבחינת טקסט ומבחינת מידע.

**מודולים פונקציונלים**

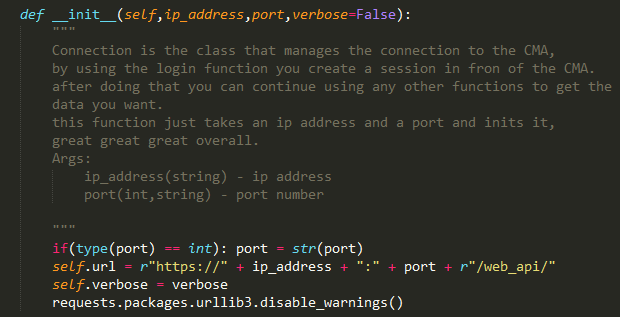
**C:\Users\cyber21\Desktop\modules.PNG**

זוג המילים הכי מפציץ שכתבתי עד עכשו שבסה''כ בא להגיד ששני המחלקות האלה מנהלות לי את כל הפונקציות הכיפיות שאתם כנראה רוצים לממש, ואני לא רוצה שייכאב לכם לממש.

**ספריית Connection ((**

**שלושת הפונקציות הקסומות שאחראיות על כך שהמחלקה תעבוד בסופו של יום**

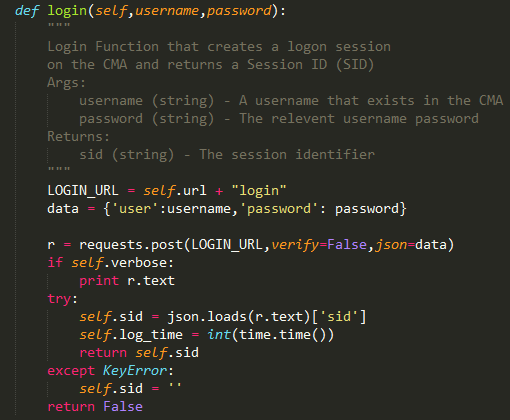
**Init**

****

יצירת המחלקה היא החלק החשוב ביותר !

הדבר היחידי שהיא צריכה זה את כתובת הIP של ה CMA , ואת הפורט עליו רץ ממשק הAPI. אם זה אצלנו ברשת אז רוב הסיכויים שזה 4434

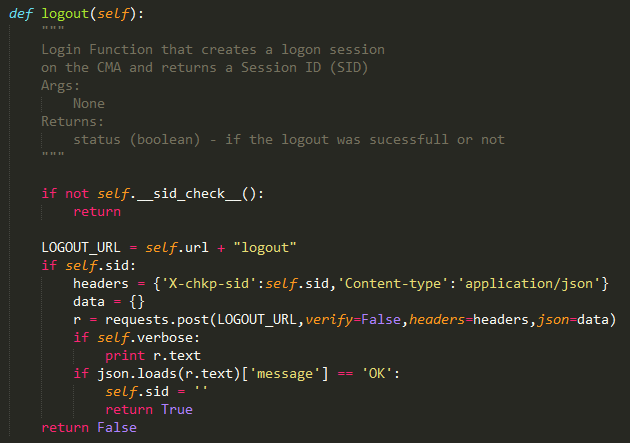
**Login**



הפונקציה לוקחת שם משתמש וסיסמה (אני מעדיף להשתמש במשתמש עם הרשאות read) ויוצרת את הסשן אל מול הCMA בכתובת והפורט שניתנו מקודם. הפונקציה מחזירה את הsid שנוצר כשתוצאה מקריאה זאת.

לפני שקראנו לפונקציה זאת והתחברנו לCMA לא נוכל לבצע את שאר הפונקציות של הספרייה ולבצע קריאות אל מול ה CMA בשביל להוציא ממנו מידע.

**Logout**



פונקציה זאתי דואגת שהסשן שנפתח ייסגר, לא קריטית אך הייתי עד למקרים שפתחתי כל כך הרבה סשנים על הCMA שזה גרם לו להאטה רצינית. (במקרה שטעיתם ועשיתם את השטות שאני עשיתי, אפשר דרך הממשק של צ'קפוינט לנתק את הסשנים ידנית)

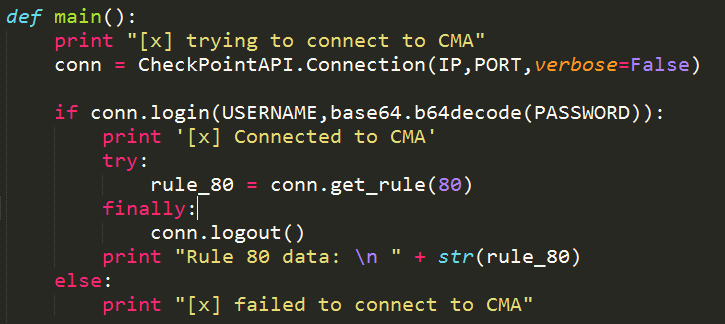
חשוב לקרוא לו בסיום כל ריצה אחרי שביצענו login והרצנו את הפונקציות שרצינו.

**דוגמאות לשימוש בספרייה**

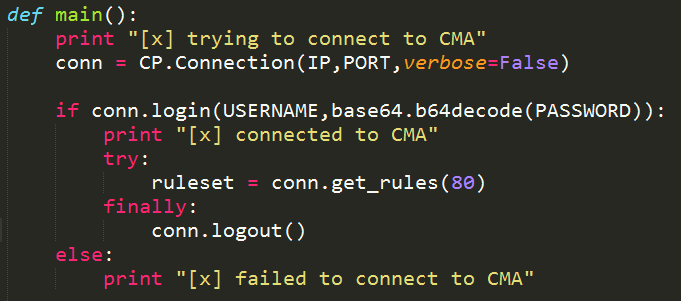
אני מצרף לתיקייה זאת לצד המדריך הזה גם סקריפט שכתבתי שמממש כמה מהיכולות שלה, בעיקר לקיחת מידע ואנליזה עליו בצורה נוחה.

ברוב המימושים ההתחלה תראה זהה לחלוטין אנחנו נממש את אובייקט Connection ע''י כך שנספק לו כתובת IP ופורט איתו אנחנו עובדים מול ה CMA

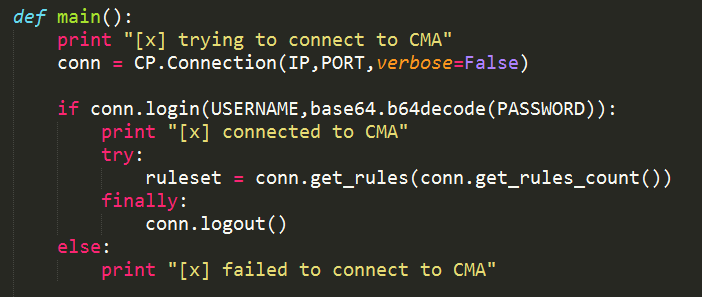
נעשה LOGIN בשביל ליצור סשן ואז נוכל להתחיל לקרוא לשאר הפונקציות שהמחלקה מספקת לנו

**הוצאת חוק יחיד** 

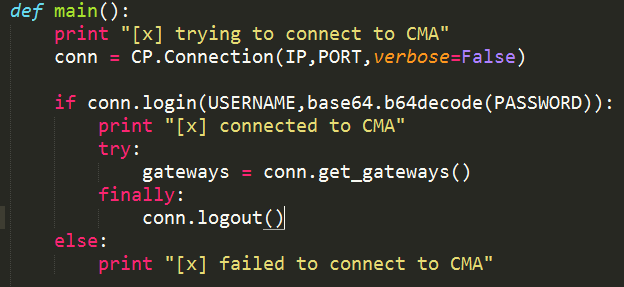
**הוצאת 80 חוקים ראשונים**



**הוצאת כל החוקים**



**הוצאת כל הקלאסטרים ברשת**



**בדיקה האם משהו יעבור את החוקה או לא**

השתמשתי במקרה הזה ב80 החוקים הראשונים (אפשר לעשות את זה על כל מספר חוקים, או על כל החוקה, הזמן שייקח לסקריפט לרוץ ייתארך בהתאם)

