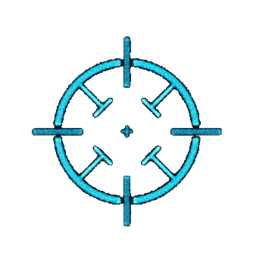
**Office Attack Patterns**



**ניתן לצפות בקוד הפרויקט גם בקישור הבא (הקוד נוהל באמצעות (GitHub,**

**(עם (RafaelsNinjas@gmail.com:**

**https://github.com/SapirC-coder/office-attack-patterns-sapir**

[**סרטון האפליקציה**](https://drive.google.com/drive/folders/1xw2vRyy1_5n_YfRBCnn3noV-60VCh-FO?usp=sharing)

**ספיר כהן**

**213884380**

מסמך תיעוד

**תוכן עניינים:**

[1. מטרות הפרויקט 3](#_Toc71109359)

[2. תיאור האתר 4](#_Toc71109360)

[3. ארכיטקטורת המערכת 6](#_Toc71109361)

[מבנה הפרויקט 6](#_Toc71109362)

[צד שרת 7](#_Toc71109363)

[מסד הנתונים 7](#_Toc71109364)

[נתיבים \ לקיחת מידע 7](#_Toc71109365)

[צד לקוח 8](#_Toc71109366)

[4. טכנולוגיות \ תוכנות צד שלישי 9](#_Toc71109367)

[צד שרת 9](#_Toc71109368)

[צד לקוח 9](#_Toc71109369)

[5. נספח: סיכום אישי 10](#_Toc71109370)

[האתגר 10](#_Toc71109371)

[בניית האפליקציה 10](#_Toc71109372)

[6. נספח: הכנות והרצות 11](#_Toc71109373)

# מטרות הפרויקט

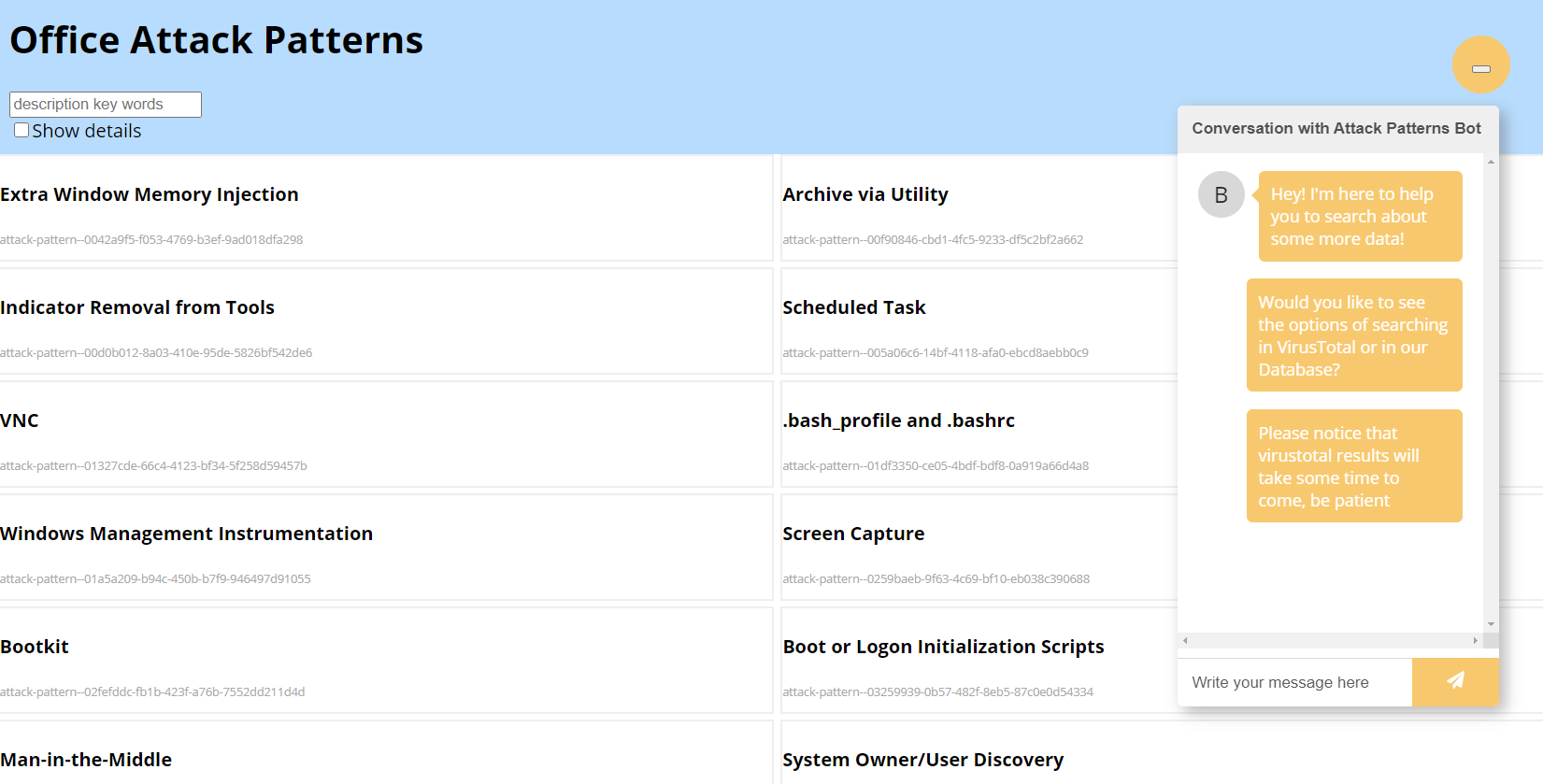
ניטור מתקפות על סביבת Office

על פי מסמך הדרישה שהתקבל להלן תקציר המטרות:

* זיהוי המתקפות הפוטנציאליות שבהן ישתמש תוקף והצגת דרך ההתגוננות.
* הצגת מתקפות בצורה נוחה וברורה.
* חיפוש לפי מילות מפתח ב-Description והחזרת המתקפות הרלוונטיות.
* יצירת מערכת חקירות של קבצים לצורך חקירות (Cuckoo sandbox)
* פיצ'ר העוזר להתמודד עם עומס התקיפות, סייבר בוט:
  + ביצוע חיפוש בDB של האתר והחזרת תוצאות.
  + ביצוע שאילתות מול מאגר חיצוני (VirusTotal) והחזרת תוצאות.
  + חיפוש באמצעות שפה פשוטה.
  + תמיכה בפקודות API של מערכת הCuckoo sandbox

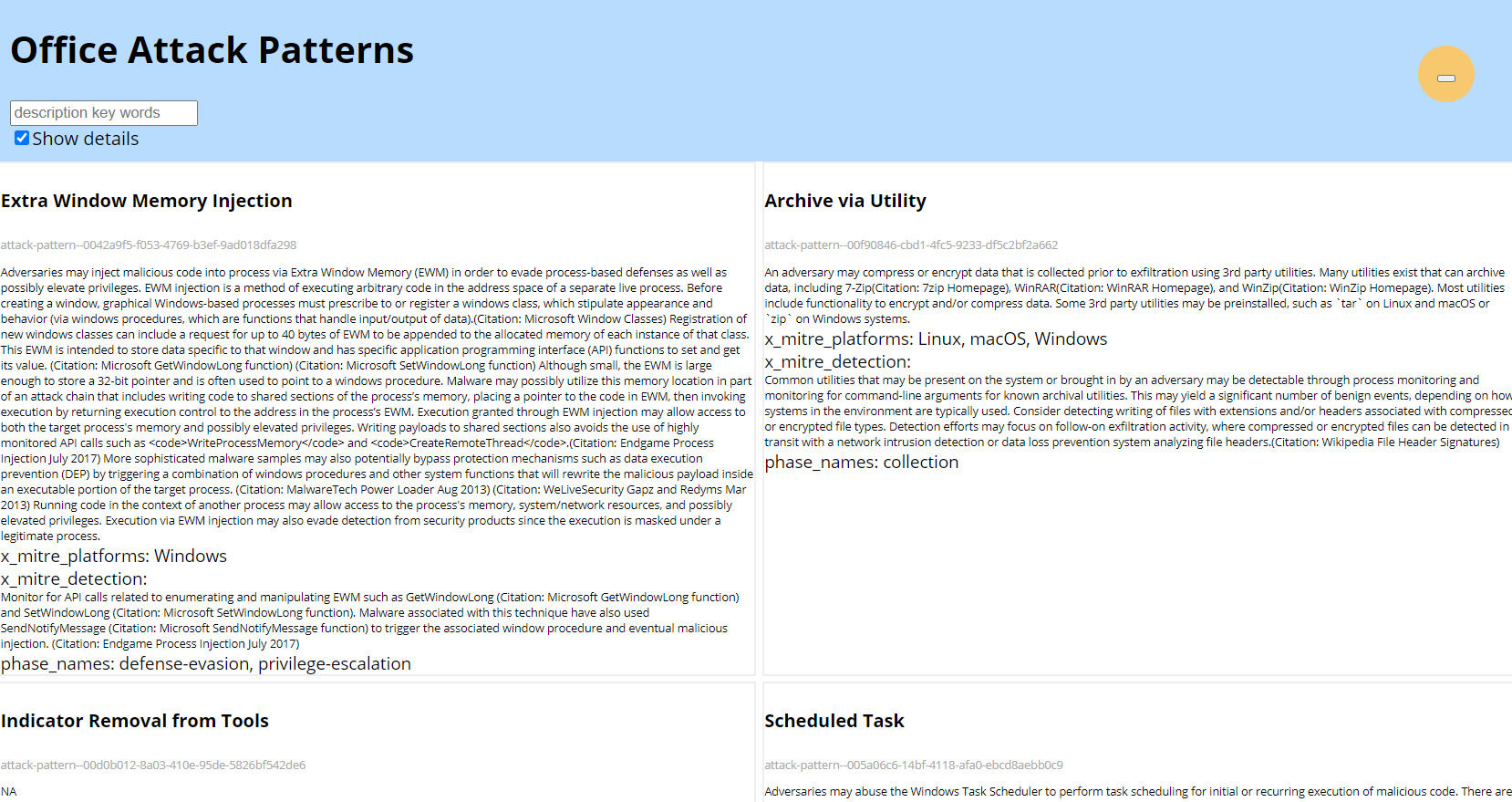
# תיאור האתר

ניתן לצפות במתקפות בצורת רשימה בה רואים רק את שמות המתקפה והמזהה שלהן:



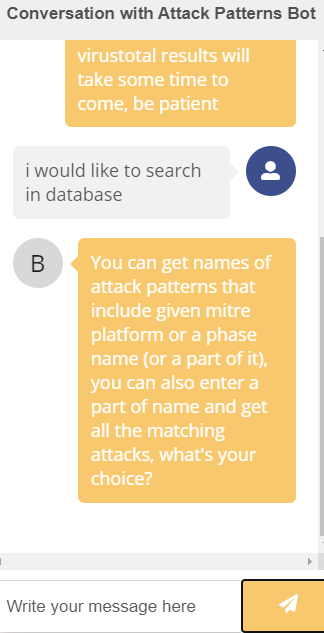
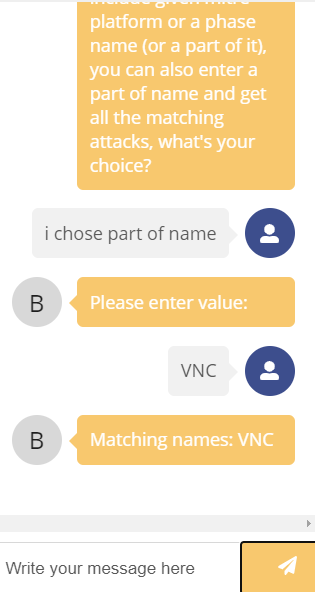
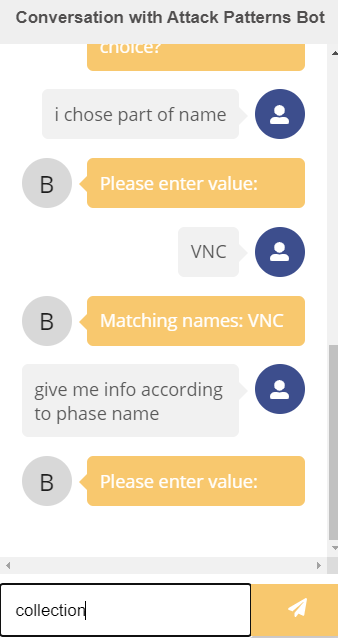
כמו כן, ניתן גם להרחיב את הנתונים ולצפות בעוד מידע רלוונטי על כל מתקפה בעזרת לחיצה על

"Show details" או להזיז את העכבר על המתקפות:

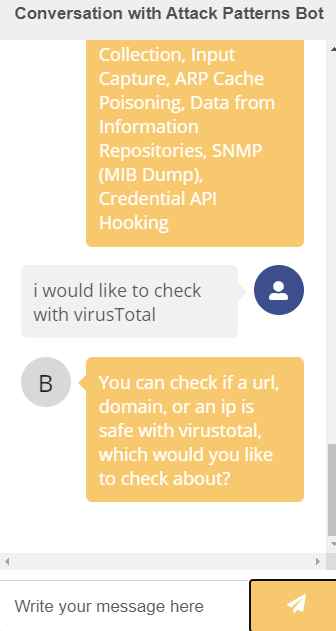
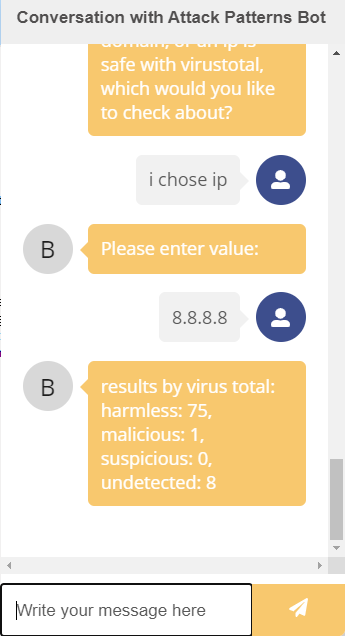


בנוסף, לחיצה על הכפתור הצהוב תסתיר את הצ'אט עם הבוט של האתר.

באמצעות הבוט, ניתן להעזר בו ולחפש עוד שמות של מתקפות רלוונטיות, לפי חלק משם, שלב מתקפה, או מערכות פגיעות הניתנות כחיפוש:



ניתן גם לבדוק IP, DOMAIN או URL באמצעות הבוט, והבוט מחזיר אם מה שניתן בטוח או לא לפי האתר VirusTotal, יש לחכות מספר שניות עד לקבלת התוצאות:



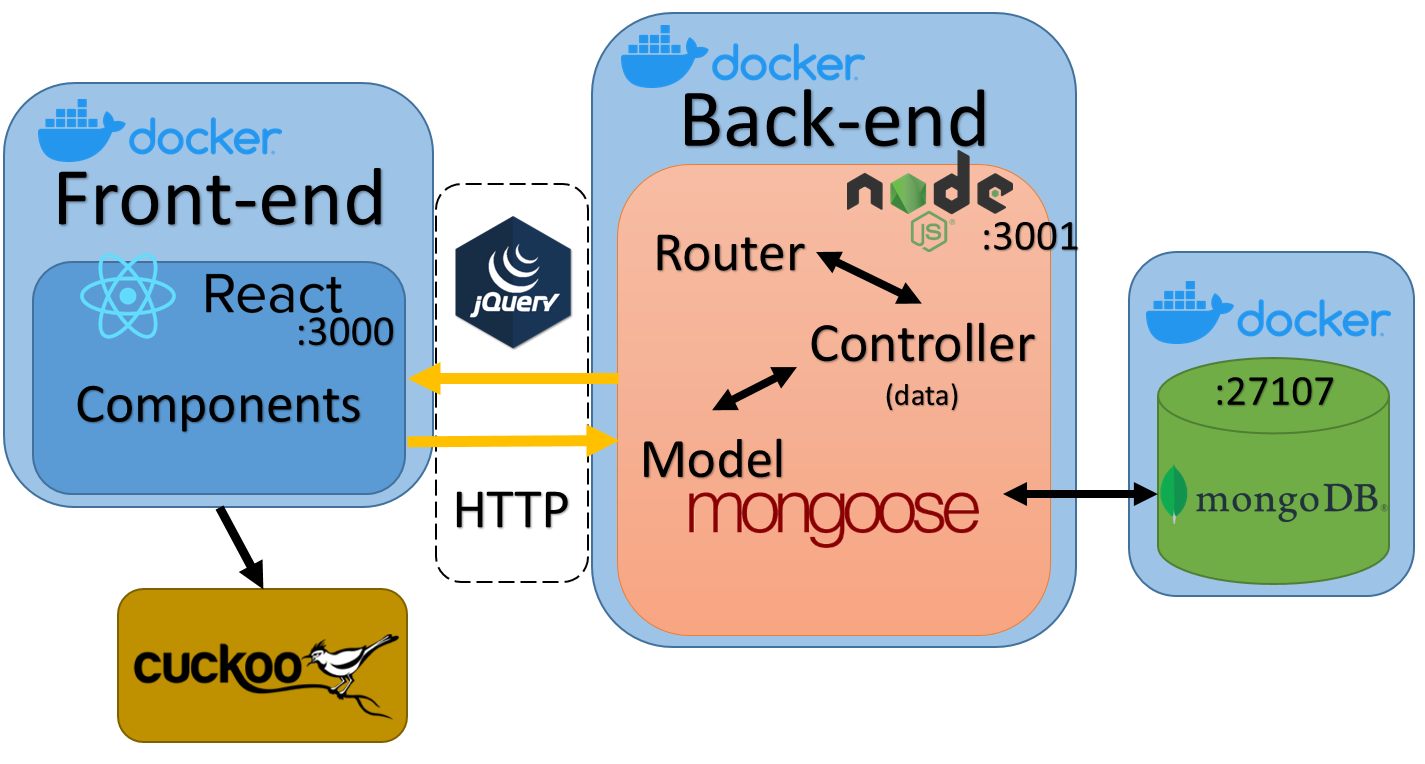
# Be your react js node js developer by Elancmia | Fiverrארכיטקטורת המערכת

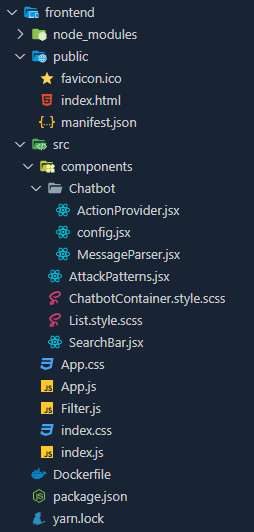
**MERN stack:**

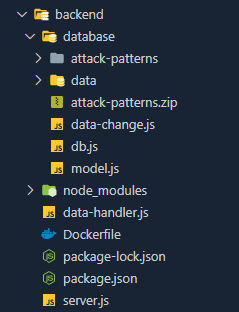
React + Node.js + Express + MongoDB

CRUD Application

## מבנה הפרויקט







## צד שרת

### מסד הנתונים

המערכת מתחברת לשרת של MongoDB: בפורט 27017 בlocalhost ((db.js

Service שפועל ברקע ומתחילים אותו באמצעות:

Linux: systemctl start mongodb

Windows: net start mongodb

לאחר מכן המידע (תיקיית attack-patterns) נשמר בשרת זה ((dataChange.js : initData() , באמצעות מודל בסיסי של (model.js) Mongoose, המכיל את השדות הנדרשים:

* + 1. name – שם המתקפה
    2. description – תיאור המתקפה
    3. \_id – מזהה ייחודי למתקפה
    4. x\_mitre\_platforms – מערכות פגיעות
    5. x\_mitre\_detection – דרכי זיהוי
    6. phase\_name – שלב המתקפה

שליפת הנתונים מתבצעת בrouts, באמצעות הפעולה find() המחזירה את כל הdocuments של המודל בו נשמר כל המידע.

### נתיבים \ לקיחת מידע

באמצעות Router של ExpressJS מוגדרים הנתיבים האפשריים בהם ניתן לקחת מידע נדרש: (data-handler.js)

|  |  |
| --- | --- |
| הפעולה | URLs |
| קבלת כל המידע של מסד הנתונים (במערך JSONs) | **localhost:3001/** |
| פנייה לאתר VirusTotal,  בבקשה לבדוק את הip שניתן והחזרת המידע הרלוונטי | **localhost:3001/vt-ip/:ip** |
| פנייה לאתר VirusTotal,  בבקשה לבדוק את הurl שניתן והחזרת המידע הרלוונטי | **localhost:3001/vt-url/:url** |
| פנייה לאתר VirusTotal,  בבקשה לבדוק את הdomain שניתן והחזרת המידע הרלוונטי | **localhost:3001/vt-domain/:domain** |

## צד לקוח

האתר מורכב מComponent עיקרי, המכיל את המתקפות בצורת רשימה או grid, ומעל מופיעים הכפתור שמגלה את הבוט, והחיפוש לפי מילות מפתח של description

הבוט והחיפוש צריכים לקבל מתקפות רלוונטיות, והם מקבלים זאת באמצעות הפונקציות בFilter.js.

בנוסף לכך, ישנו צורך בלקיחת מידע מסויים מהשרת לפי הנתיבים הרלוונטים, עבור הצגת מידע של virusTotal בבוט, או מידע של הDatabase.

קבלת המידע מהשרת נעשה באמצעות בקשת HTTP GET, עם jQuery.

החיפוש הוא Component נוסף, SearchBar.jsx

הבוט הוא של react-chatbot-kit והוא מורכב מ:

ActionProvider.jsx, MessageParser.jsx, config.jsx

ActionProvider: שולח הודעות למשתמש, ובו יש פניות לשרת לפי מה שמתבקש מהמשתמש, או שימוש בפילטור מידע (לפי מה שמוזכר מעלה, בFilter.js)

MessageParser: מנתח הודעות שהתקבלו מהמשתמש ומשתמש בפונקציות הרלוונטיות בclass ActionProvider.

Config: ההגדרות של הבוט, הודעות ראשניות, איך הוא יראה והשם שלו.

# טכנולוגיות \ תוכנות צד שלישי

**שיטת ה":"deployment**

**דרך פריסת המוצר ומה שמאפשר**

 **מריצים אותו בעזרת:,CI/CD**

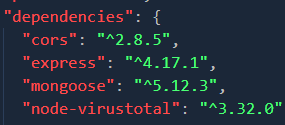
**יש להוריד דוקר לפני)) Docker-compose up**

**באמצעות דוקר מורדות כל החבילות הרלוונטיות לפרויקט,**

**ניתן להפעיל שירותים שונים (לדוגמא נדרש שרת מסד נתונים)**

**חבילות: לפי Package.json:**

## צד שרת

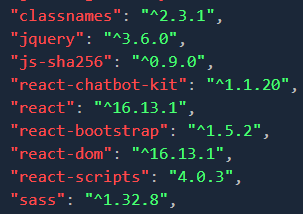


cors– Cross-Origin Resource Sharing

mongoose – ORM שמאפשר התממשקות נוחה מול הDATABASE

node-virustotal –VirusTotal API for Node JS , באמצעות זה נשלח ומקבל המידע מVirusTotal

## צד לקוח



classnames, react-bootstrap, sass – עזרים בשביל Components ועיצוב

jquery – לקיחת מידע מצד השרת

js-sha256 – הפיכה לHASH בשביל הURL הנשלח לVirusTotal

react-chatbot-kit – הבוט

# נספח: סיכום אישי

## האתגר

האתגר העיקרי שעמד בפני היה ללמוד את כל הנושאים בעצמי ולחקור מה הדרך הנכונה ביותר כדי ליישם את האפליקציה.

בהתחלה היה לי קשה לתפוס איך כל דבר מתחבר אחד לשני, עד היום למדתי איך לפתור בעיות בנפרד ובזכות הפרויקט למדתי איך לעשות intergration בצורה טובה.

התחלתי לחקור דברים ולהבין אותם בצורה הפשוטה ביותר, ולאחר מכן אחרי שהבנתי מה הכי נכון לעשות, התעמקתי ולמדתי על עצמי שאני מסוגלת להגיע לתוצרים גדולים.

בעתיד אלמד גם לכתוב testים כדי לייעל את תהליך העבודה והבדיקות.

## בניית האפליקציה

תהליך בניית האפליקציה היה שונה ממה שעשיתי בדרך כלל, בנוסף, למדתי על הקשיים התשתיים שיכולים לעמוד בפניי, למשל בעקבות Cuckoo שדרש מקום רב הlinux שלי הפסיק לעבוד וככה הייתי צריכה לטפל גם בו, יצרתי partition חדש בשביל שיהיה לי עוד מקום והשתמשתי ב system rescue cd כדי להעביר את הכל.

בנוסף, גיליתי שהתיעוד הרשמי של cuckoo לא מעודכן, מה שהקשה עליי להגיע לפתרון.

מאוד נהנתי מבניית האפליקצייה, זה היה דבר שונה מכל קוד אחר שכתבתי ואפילו יותר נהנתי לעשות את הפרויקט הזה מאשר לעשות משהו במגשימים, נהנתי מהתהליך ומהעובדה שהייתי צריכה ללמוד מלא דברים לבד, לחקור מה הדרך הנכונה ביותר, עם מה הכי נכון להשתמש ולהכיר דרכים שונות לפתרון.

אהבתי לעבוד עם המערכת של CUCKOO למרות הקושי שהיה, ולבנות את צד השרת.

# נספח: הכנות והרצות

## Virtualbox of windows

[**לעשות import למה שנמצא בדרייב**](https://drive.google.com/drive/folders/1xw2vRyy1_5n_YfRBCnn3noV-60VCh-FO?usp=sharing)

**או ליצור virtualbox של ווינדווס 7:**

לצרף אליו את הiso המתאים, נדרש במכונה מינימום 20 GB

לאחר ההתקנה יש להוסיף את הדברים הבאים בהגדרות:

בnetwork

להוסיף host only adapter של vboxnet0

בfilesharing

לבחור מקום בו תהיה התיקייה המשותפת

להוסיף בה את הקבצים הבאים ואת agent.py שנמצא בחלק של הcuckoo

[Pyrhon2.7](https://www.python.org/ftp/python/2.7.8/python-2.7.8.amd64.msi)

[Python pillow](https://pypi.python.org/packages/2.7/P/Pillow/Pillow-2.5.3.win-amd64-py2.7.exe#md5=33c3a581ff1538b4f79b4651084090c8)

לאחר מכן כאשר המכונה פועלת, יש לבחור באפשרות בסרגל למעלה בה רשום devices ולבחור באפשרות האחרונה, להוריד את הכל, ללחוץ על computer ולהפעיל את מה שהורדנו ולבסוף לאפשר הפעלה מחדש.

כאשר המכונה נדלקה יש ללחוץ על network ושם אמורה להופיעה התיקייה המשותפת עם הקבצים, יש להריץ את ההתקנה של פייתון, לאחר מכן את ההתקנה השניה ולבסוף לפתוח את agent וליצור snapshot

**Preparations and running**

**Extract attack-patterns.zip in oap-app/backend/database**

Make sure the folder containts files and not a directory (use extract here)

**Please notice!**

**In order to use cuckoo in the chatbot you have to do the following steps:**

**in ubuntu 20.04:**

**Install all requirements:**

**sudo apt install -y net-tools**

**sudo apt-get install uwsgi uwsgi-plugin-python nginx**

**create a host only network adapter:**

**vboxmanage hostonlyif create**

**setting the network of vb:**

**vboxmanage hostonlyif ipconfig vboxnet0 --ip 192.168.56.1**

**running cuckoo:**

**in ve:**

**. cuckoo/bin/activate**

**cuckoo --cwd ~/cuckoo-proj-sapir/cuckoo {~ :all the path before the directory}**

**starting the server in a new terminal:**

cuckoo --cwd ~/projects/cuckoo-proj-sapir/cuckoo web {~ :all the path before the directory}

**After that you got 2 options:**

**Install docker (Recommended)**

[click here for download](https://www.docker.com/get-started)

**Run in oap-app docker-compose up to start running, and change the word start to down incase you want to close the servers of the app**

**change in backend\database\db.js in DB\_IP: 127.0.0.1 to db**

**Or**

**Install mongodb**

[install community server](https://www.mongodb.com/try/download/community)

**Open a service of mongo:**

**In linux run: systemctl start mongodb**

**In windows run: net start MongoDB**

**change the word start to stop incase you want to stop the service**

**In backend**

**Run: npm install**

Only on the first run

**Run: node server.js**

**In frontend**

**Run: yarn install**

Only on the first run

**Run: npm start**