המחלקה למערכות מידע ניהוליות

פרוייקט גמר תשף 2020

מעבדה לזיהוי התקפות ברשת ה-LAN באמצעות למידת מכונה ללא תיוג של פעולות אנומליות

NETWORK ATTACK DETECTION SYSTEM USING AN UNSUPERVISED MACHINE LEARNING ALGORITHM

LAB

מסמך זה מהווה חלק מדרישות לקבלת תואר ראשון במערכות מידע BSC

מכללת אחווה

שמות המבצעים: אלמוג אליאס, קובי תורג'מן וכנרת כהן.

שם המנחה: ד"ר יצחק רוזן.

**הצהרה**

הפרויקט נעשה בהנחיית ד"ר צחי רוזן

המכללה האקדמית אחווה- מחלקת מערכות מידע ניהוליות

מכיוון שהפרויקט הנו בלמידת מכונה ושימוש בתחום איתור התקפות בתעבורת הרשת על בסיס אנומליה, הוא חדשני יחסית, דרוש מחקר רב על מנת לבצע את הפרויקט בנוסף למימוש עצמו.

הפרויקט מציג את עבודתנו האישית ומהווה חלק מקבלת תואר ראשון במערכות מידע ניהוליות.

תוכן עניינים

[1. תיאור הארגון 1](#_Toc41577581)

[1.1. רקע כללי 1](#_Toc41577582)

[1.2. לקוח 1](#_Toc41577583)

[1.3. מומחי היישום 1](#_Toc41577584)

[1.4. משתמשים 1](#_Toc41577585)

[2. מטרות ויעדי הארגון 1](#_Toc41577586)

[2.1. מטרת הארגון 1](#_Toc41577587)

[2.2. עקרי ענף המערכת 1](#_Toc41577588)

[2.3. הצגת מטרות המערכת 2](#_Toc41577589)

[2.4. תיאור הבעיה 2](#_Toc41577590)

[3. סקירת ספרות 3](#_Toc41577591)

[*3.1.* *אנומליה* 3](#_Toc41577592)

[*3.2. מה זה נורמה?* 4](#_Toc41577593)

[*3.3.* *זיהוי אנומליה על ידי איתור מרחק* 4](#_Toc41577594)

[*3.4.* *מדדי דמיון Similarity Measures* 4](#_Toc41577595)

[*3.5.* *מידע חריג במרחב חד מימד* 4](#_Toc41577596)

[3.6. מידע חריג במרחבים רב מימדיים 6](#_Toc41577597)

[3.7. Isolation Forest 6](#_Toc41577598)

[4. תיאור המערכת העתידית 8](#_Toc41577599)

[4.1. High-Level 8](#_Toc41577600)

[4.2. Low Level 8](#_Toc41577601)

[5. בחירת Features 9](#_Toc41577602)

[5.1. כמות חבילות ליחידת זמן 9](#_Toc41577603)

[5.2. כמות הודעות ל- Flow 11](#_Toc41577604)

[5.3. גודל (bytes) חבילה ליחידת זמן 14](#_Toc41577605)

[5.4. הפרש הזמנים בין יצירת הקובץ לשימוש בקובץ. 16](#_Toc41577606)

[5.5. סיכום Features 17](#_Toc41577607)

[6. דרישות פונקציונאליות - כללי 18](#_Toc41577608)

[6.1. איסוף המידע 18](#_Toc41577609)

[6.2. עיבוד הנתונים 18](#_Toc41577610)

[6.3. ממשק WEB- UI 20](#_Toc41577611)

[6.4. זיהוי חריגות 21](#_Toc41577612)

[7. דרישות פונקציונאליות - מפורט 21](#_Toc41577613)

[7.1. איסוף המידע 21](#_Toc41577614)

[7.2. עיבוד הנתונים 22](#_Toc41577615)

[7.2.1. Feature Packets in 10 seconds 22](#_Toc41577616)

[7.2.2. חישוב מספרי בFLOWS 23](#_Toc41577617)

[7.2.3. מציאת קבצי הרצה 24](#_Toc41577618)

[7.2.4. חישוב מטריקות 25](#_Toc41577619)

[7.3. ממשק WEB - UI 27](#_Toc41577620)

[7.3.1. מערכת הWEB 27](#_Toc41577621)

[7.3.2. עצירת/ משיכת תזרים מידע לWEB 28](#_Toc41577622)

[7.4. זיהוי חריגות 29](#_Toc41577623)

[7.5. סימולטור התקפה 30](#_Toc41577624)

[8. תיאור ארכיטקטורה הכללית של המערכת 34](#_Toc41577625)

[9. הדרישות הלא פונקציונאליות 35](#_Toc41577626)

[10. פעולות שיתבצעו באופן שוטף 35](#_Toc41577627)

[10.1. האזנה 35](#_Toc41577628)

[10.2. תצוגת Dashboard 35](#_Toc41577629)

[10.3. מציאת חריגות 36](#_Toc41577630)

[10.4. הצגת נתונים בממשק WEB - UI 37](#_Toc41577631)

[10.5. עצירה / משיכה של תזרים מידע למסך הDashboard 37](#_Toc41577632)

[10.6. Web extension 38](#_Toc41577633)

[10.7. התקפת סייבר 39](#_Toc41577634)

[11. מילון נתונים 39](#_Toc41577635)

[12. עיצוב מנשקי עץ התפריטים ומסכי קלט פלט 42](#_Toc41577636)

[12.1. סימולטור התקפה 42](#_Toc41577637)

[12.2. ממשק WEB - UI 42](#_Toc41577638)

[13. בסיס הנתונים 43](#_Toc41577639)

[14. ארכיטקטורת המערכת והמודלים העיקריים 43](#_Toc41577640)

[15. תיאור לוגי של המערכת 44](#_Toc41577641)

[16. Class Diagram 45](#_Toc41577642)

[16.1. האזנה 45](#_Toc41577643)

[16.2. תצוגת Dashboard 46](#_Toc41577644)

[16.3. סימולטור התקפה 47](#_Toc41577645)

[17. Sequence Diagram 48](#_Toc41577646)

[17.1. האזנה 48](#_Toc41577647)

[17.2. ממשק WEB - UI 49](#_Toc41577648)

[17.3. סימולטור התקפה 50](#_Toc41577649)

[18. חלוקת תפקידים 51](#_Toc41577650)

[19. לוחות זמנים 52](#_Toc41577651)

[20. ביבליוגרפיה 53](#_Toc41577652)

# 

# תיאור הארגון

## רקע כללי

רשת מקומית LAN מאפשרת למחשב אחד המחובר לרשת להשתמש במשאבים ממחשבים אחרים. שימושים נוספים הם הרצת יישומים ממחשב אחד על מחשב אחר.

חסרונותיה של העבודה ברשת הם סיכון לפרטיות המידע וריבוי מקורות לתוכנות זדוניות. המחשבים שברשת יכולים להתחבר ביניהם בכבלים או בקשר אלחוטי ויש לוודא שאין פעילות חריגה על הרשת, ושרק אנשים מורשים יוכלו להשתמש במחשב שכבר מחובר אל הרשת. מעבר לזאת יש להגביל את הגישה של מי שכן מורשה כך שתוכנם של מסמכים יהיה גלוי רק לבעלי סמכות רלוונטית עבור תוכנו של כל מסמך או תוכנה.

## לקוח

לקוחות המערכת הם חוקרי אבטחת המידע. הלקוח מגדיר לנו מה לעשות ובהתאם לזה נבנה כנדרש. המערכת תשמש בעיקר את משתמשי הרשת המקומית LAN. הגדרנו את קובי להיות המייצג של הלקוח, אשר יהיה מסוגל להגדיר את הדרישות של הלקוח.

## מומחי היישום

מומחי היישום הינם חוקרי אבטחת מידע אשר יטמיעו את המערכת.

ההחלטה לגבי הפיתוחים תהיה על ידי צוות היגוי שתורכב מראש צוות האפיון, ומומחה אבטחת מידע.

## משתמשים

חוקרי אבטחת מידע ישתמשו במערכת כדי להגיע למסקנות ולתוצאות במחקרם.

# מטרות ויעדי הארגון

## מטרת הארגון

הטכנולוגיה חשופה לשלל פרצות מידע, ויש צורך תמידי לשכלל ולהתמודד עם באגים, אשר הולכים והופכים למתוחכמים יותר .ישנם שיקולים שחייבים להילקח בחשבון כשעוסקים בחיבור מחשבים לרשת. הנחה היא שאי אפשר לעבוד ללא תקשורת בין מחשבים ולכן המטרה היא לזהות פעילות חריגה עויינת של תוקפים למניעת התפשטות ברשת המקומית.

## עקרי ענף המערכת

הענף בו עוסק הפרויקט הינו תחום הסייבר על ידי שיטות ואלגוריתמים של למידת מכונה.  
קהל היעד של הפרויקט הנו:

1. משתמשי רשת מקומית.
2. חוקרי אבטחת מידע.

## הצגת מטרות המערכת

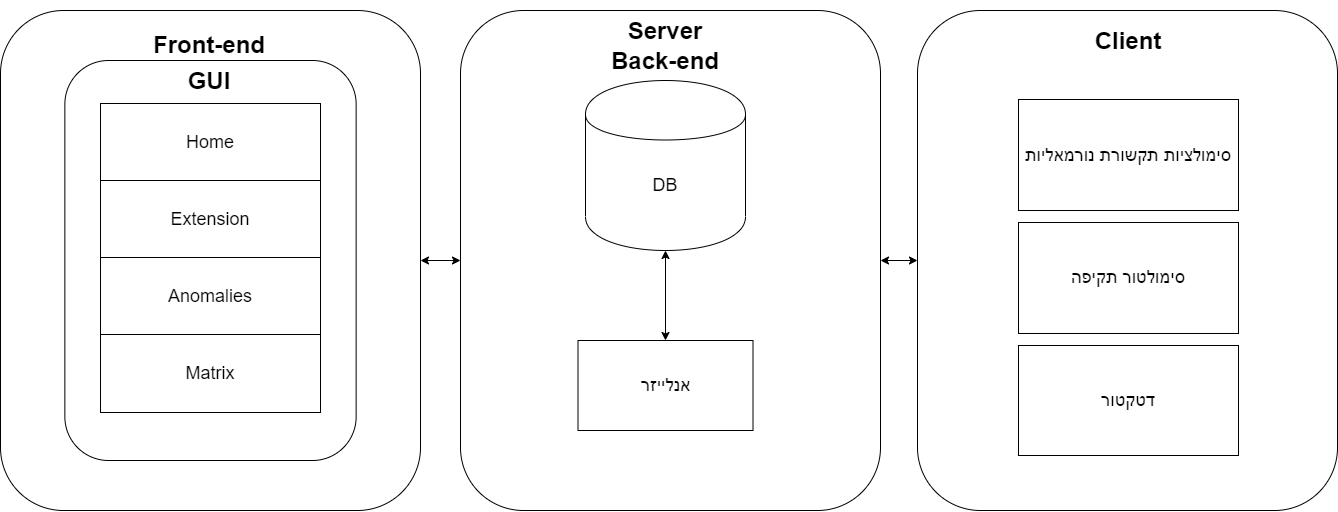
הגדרת הפרויקט היא בניית תשתית מעבדה שתאפשר לחקור תופעה אנומלית.

התשתית תכיל רכיב תקשורת, עמדות קצה אשר יתקשרו בינהם ושרת אשר יאסוף את המידע.

כל רכיב יפעל באופן עצמאי, אשר ידמה מצב אמיתי ברשת ה - LAN.

המערכת תאפשר לחוקר להשתמש בכלי תקיפה ולדמות התקפות באמצעות הסימולטור ובכך יוכל להגדיר את ההתנהגות שלהן באמצעות אלגוריתם של למידת מכונה.

בכדי להציג את המידע והממצאים תוקם מערכת WEB המציגה את המידע.



**Client** – עמדת קצה בסביבת המעבדה אשר תשמש להעברת חבילות מידע לעמדה אחרת במעבדה ולזיהוי התעבורה.

* **סימולציות תקשרת נורמאליות** – העברת חבילות תקשורת נורמאליים בין מחשבים.
* **סימולטור תקיפה** – תפקיד מודול זה הוא להזרים תקשורת לא נורמאלית בסביבת המעבדה.
* **דטקטור** – אוסף נתונים מהמעבדה ורושם אותם לDatabase.

**Server Back-End** – שרת מארח את ה-Database ואת האנלייזר.

* **DB** – מנגנון שמירת המידע אשר ימומש בשרת.
* **אנלייזר** – מנתח את הנתונים שנאספו על ידי Clustering ובודק את איכות האותות.

**Front-End** **GUI**– ממשק WEB UI.

* **Home**- הדף הראשי של האתר אשר מציג מידע בזמן אמת.
* **Extension**- הרחבה של הדף הראשי אשר מציג מידע סטטי ברגע נתון.
* **Anomalies**- דף המציג את האנומליות שנמצאו ברגע נתון.
* **Matrix**- דף המציג את הסטטיסטיקות של המערכת.

## תיאור הבעיה

כמות המידע שגוברת עם הזמן והתממשקות מול רכיבים נוספים גורמת לסיכון גבוה יותר לתקיפת סייבר בעולם וברשת הפנימית בפרט.

כיום ישנם חברות קיימות אשר משלבות מוצר חומרתי (נתב) או חוצץ בין הרשתות כדי לזהות אנומליות אך לא בהכרח כל התעבורה עוברת דרך הנתב.

אף על פי כן, מוצרים אלו מאתרים איומים והתקפות בתעבורה ע"י חיפוש תבניות ספציפיות או ע"י הכרת דפוס הפעולה של הקובץ הזדוני, כך שלכל מתקפה קיימת תבנית ייחודית עבורה.

כאשר נחשף איום חדש יוצרים תבנית תואמת לאיום ולפי התבנית המערכת מזהה את האיום לאחר מכן.

ההנחה היא שהתוקף כבר נמצא ברשת וע"י למידת וחקירת הרשת הוא מזהה מטרות תקיפה.

**הבעיה** היא זיהוי פעילות בשלב  Active Reconnaissance של תוקף וזמן התגובה לאיום זה מהרגע שבו הוא מזוהה עד לבניית התבנית עבורו, כיוון שכבר נוצרה פגיעה במערכת. היום בעולם המודרני ישנה עלייה בכמות המידע העוברת ברשת ובעקבות כך נוצר ריבוי בסכנות וסיכוי גבוה יותר לפריצות ע"י תוקפים פוטנציאלים.

מדובר על **מערכת מידע מורכבת** שלא ניתנת לחיזוי מתקפות סייבר עתידיות.

# סקירת ספרות

הוחלט להתבסס על הספר "Anomaly Detection Principles and Algorithms" ועל המחקר "IsolationForest" מאת Fei Tony Liu, Kai Ming Ting Gippsland School of Information Technology Monash University, Victoria, Australia

## *3.1. אנומליה*

בכריית מידע, זיהוי אנומליות Anomaly Detection (לעיתים מכונה גם גילוי אנומליות או זיהוי חריגים) מתייחס לאיתור תצפיות או אירועים אשר אינם תואמים לדפוס הצפוי או ליתר התצפיות בבסיס הנתונים.

לרוב, התצפיות החריגות מעידות על בעיה. אנומליה בתעבורת רשת תקשורת עשויה להעיד על מתקפת סייבר על רשת התקשורת.

ניתן לחלק את הטכניקות לזיהוי אנומליות לשלושה סוגים:

טכניקות לא מונחות (unsupervised), אשר מניחות כי רוב התצפיות בבסיס הנתונים מייצגות מקרים נורמליים או תקינים. לשם כך ניתן, למשל, להשתמש בטכניקות של ניתוח אשכולות כדי לאפיין את התצפיות הנורמליות. תצפית שאינה שייכת לאף אשכול מוגדרת כאנומליה.

טכניקות מונחות (supervised), אשר בהן התצפיות בבסיס הנתונים סווגו מראש לתצפיות "נורמליות" או לתצפיות "לא נורמליות". במקרה כזה, ניתן להשתמש בשיטות של למידה חישובית לאימון מסַווגים, המאפשרים סיווג של מקרים חדשים שאותם לא פגשנו בתהליך הלמידה.

טכניקות מונחות למחצה (semi-supervised), אשר בונות מודל המייצג את דפוס ההתנהגות הרגיל מתוך אוסף של תצפיות נורמליות. לאחר מכן בוחנים באמצעות המודל את הנראות של תצפיות חדשה.

כאשר מיושמים אלגוריתם לאיתור חריגות, יש לקחת בחשבון ארבעה מקרים אפשריים:

1. Correct Detection: חריגות שנמצאו בנתונים אכן תואמות בדיוק את החריגות בתהליך.

2. False Positive: התהליך ממשיך להיות תקין, אך נצפים ערכי נתונים לא צפויים, למשל בגלל רעש מערכת פנימי.

3. False Negative: התהליך הופך להיות חריג, אך ההשלכות אינן רשומות כנתונים הלא תקינים, למשל, בגלל שהאות החריגה חזקה באופן לא מדויק לעומת הרעש במערכת.

4. True Negative: לא נמצאו חריגות בנתונים אשר גם לא נמצאו בתהליך.

על פי המחקר, אנומליה מוגדרת על פי:

* ישנם קצת נקודות כאלה.
* שונות מהנקודות הרגילות ושונות אחת מהשניה.

## *3.2. מה זה נורמה?*

באנליזה מתמטית, נורמה היא פונקציה ממשית המוגדרת על מרחב וקטורי, ומתאימה לכל וקטור ערך ממשי, באופן שמתמלאים מספר תנאים. תנאים אלו מבוססים על התכונות היסודיות של האורך המוכר במרחב האוקלידי. מרחב וקטורי שמוגדרת עליו נורמה נקרא מרחב נורמי.

האורך במרחב האוקלידי מקיים את הדרישות הטבעיות הבאות:

* אורך הוא תמיד חיובי, חוץ מאורכו של וקטור האפס, שהוא אפס.
* מתיחה של הווקטור בסקלר מכפילה גם את האורך בערכו המוחלט של אותו סקלר.
* מתקיים אי שוויון המשולש.

## *זיהוי אנומליה על ידי איתור מרחק*

**Distance based anomaly detection approaches**

*בהגדרה, זיהוי אנומליה כרוך בכך שנתון מסויים "שונה" מאחרות. הגדרה זו מוגדרת בהכרח על ידי מערך הנתונים שעליו משווים את נקודת הנתונים.*

*חלק זה מתמקד באלגוריתמים לאיתור חריגות הנשענים על חישובי מרחק בין נקודות שונות לסק נקודות. הסמל D משמש לייצוג מערך הנתונים ה – N ממדי, המשוער כי נמצא במרחב רציף ללא הערכה אמיתית Rn. נקודות נתונים ב-D מסומנות על ידי תווים p,q. סמלים בעלי אותיות גדולות כמו P משמשים לציון קבוצות נקודות, כלומר . המרחק בין 2 נקודות מסומן על ידי d(p,q)*

1. *מדידה: עד כמה נקודה נתונה באופן לא נורמלי? זה דורש הפיכת נקודות לסולם חד ממדי.*
2. *אבסולוטי: האם נקודה נתונה היא אנומלית? זה דורש threshold אשר אומר שנקודה*

*היא אנומאלית אם .*

1. *רלטיבי: האם נקודה אחת יותר אנומלית מנקודה אחרת? זה מצריך השוואה בין נקודות. אז*

*אם נקודה יותר אנומאלית מנקודה אחרת .*

*נוכל לסמן את ההקשר עם הסימן אם .*

## *מדדי דמיון Similarity Measures*

*לעתים קרובות אנו מוצאים דמיון במונחים של מידת מרחק. נשים לב שניתן למדוד דמיון על ידי השגת 'עקרון' בהתחשב ב'הפוך' של מידת מרחק. אמצעי דמיון נפוצים המיועדים לכלול כוללים אמצעים ישירים כמו האוקלידים, מינקובסקי ומהלאנוביס. משתמשים גם באמצעים נוספים, כמו דמיון קוסינוס ומדד ג'קרד; לעתים קרובות אמצעים עקיפים, כמו השכן המשותף הקרוב ביותר (SNN - מספר הנקודות הנפוצות בין k שכנים של שתי נקודות מייצג את הדמיון הרצוי בין הנקודות).*

## *מידע חריג במרחב חד מימד*

כל נקודת מידע היא מספר יחיד. ישנם התפלגויות שונות לחריגים במרחב חד מימד:

Uniform Distribution – התפלגות אחידה

כאשר נתונים מתפלגים באופן אחיד על טווח סופי, הממוצע וסטיית התקן רק מאפיינים את טווח הערכים. אם בקרבת איזור של כל נקודת נתונים היא עשירה בנקודות כמו כל נקודה אחרת, אז אין חריגות בתנאי שהגבולות התחתונים והעליוניים של התחום רחוקים יחסית מהממוצע. אפשרות לאנומליה יכולה לקרות כשאיזור קטן מכיל פחות או יותר נקודות נתונים אשר מצופה בהתפלגות האחידה.

Normal Distribution - התפלגות רגילה

כאשר נתונים מתפלגים נורמאלית, צפיפות הנקודות יורד משמעותית ככל שאנחנו מתרחקים מהממוצע. כ 0.1% מהנקודות יותר מ- (3 סטיות סטאנדרטיות) מהממוצע, ורק % *מהנקודות אשר יותר מ6 סטיות סטאנדרטיות מהממוצע.*

נקודות אשר נמצאות מעבר למרחק מהממוצע נחשבים לחריגים. לעומת זאת קיומם של מספר נקודות הרחוקות מהממוצע זהו צירוף מקרים להתפלגות נורמאלית, מכאן ניתן לטעון כי אלה נקודות חריגות במידה והמספר שלהם גבוה מהמספר המצופה כשהנתונים מתפלגים נורמלית. לדוגמה אם 2% מנתוני המידע נמצאים מעל 3 סטיות סטאנדרטיות.

Other Unimodal Distributions – התפלגויות Unimodal אחרות

רבים מהתפלגויות Unimodal לא נורמאליים, כאשר יש גבול תחתון לטווח ערכי הנתונים, לדוגמה התפלגויות המכילות Log-normal ו- Gamma. אם מאפייני התפלגות ידועים, אחת מהדרכים לנסות למצוא Thresholds beyond אשר הוא מספר קטן (כ-1%) של נקודות נתונים. שוב, ניתן לטעון כי אוסף הנתונים הללו הינו אנומאלי אם מספרם גדול מחיזוי הסטטיסטיקה.

Multimodal Distributions – התפלגות מולטימליות

ההתפלגות לכמה מערכי נתונים כוללים מספר רב מצבים, שיתגלו רק כאשר הנתונים נבדקים מקרוב. היוריסטיקות כגון חוק אינו מועיל בהתפלגויות כאלה. ולכן, צריך חשוב על הנתונים אשר מורכבים מאוסף אשכולות של נקודות נתונים.

Clustering מהווה חלק חשוב בזיהוי אנומליות. נקודות אשר לא שייכות להתאגדות מסויימת של נקודות יכולות להיות מסומנות כנקודות חריגות. למעשה, נקודות אשר רחוקות מהתאגדויות שכנות נחשבות לוגית לאנומליות. הדבר החשוב במודל מסוג Clustering הוא ההבהרה מתי אוסף של נקודות צריך להיחשב כהתאגדות, ומה המשמעות של נקודה אשר נמצאת רחוק מספיק מהתאגדות זו (או מכמה התאגדויות).

ההגדרה הבלתי פורמלית של אשכול היא שמדובר באוסף של נקודות קרובות זה לזה.

Density-based cluster – התאגדות מבוססת צפיפות

זיהויים פופולארים מאוד בזיהוי מידע אנומאלי. אם המספר היחסי של נקודות (ליחידת מרחק) גדול משמעותית באיזור קטן מכל מערך הנתונים. ניתן לקחת בחשבון את הנקודות באיזור הזה כהתאגדות של נקודות. זו עדיין לא הגדרה מתימטית, מכיוון שהביטויים "איזור קטן" ו-"גבוה משמעותית" לא מוגדרות.

ההתפלגות של הצפיפות ניתנת לניתוח, (אם איננו מודאלי) נוכל לזהות את צפיפות הסף העליון כשיש צפיפות גבוהה אשר מספיקה לשקול איזור זה כהתאגדות של נקדות.

נשים לב שאותו מערך הנתונים יכול להכיל איזור אחד עם צפיפות גבוהה ואיזור אחר עם צפיפות נמוכה שגם נחשיב אותו להתאגדות.

Compare Intra & Inter groups distance – השוואת מרחקים פנים קבוצתיים עם בין קבוצתיים.

אספקט נוסף, הינו השוואה בין מרחקים בפנים קבוצתיים עם מרחקים בין קבוצתיים, ובכך לטעון שאוסף נקודות מהתאגדות קרובים יותר אחד לשני (יותר מהממוצע) מאשר מרחקים מהנקודות נמצאות מחוץ להתאגדות. לעומת זאת, על פי הגדרה זאת התפלגות המידע עלולה להוביל להתאגדות טרוויאלית אשר מאכלסת כמעט את כל הנקודות בהתאגדות אחת.

חלק מאלגוריתמי ה-Clustering מקצים כל נקודה להתאגדות מסויימת אך לא בהכרח.

אחרים, מניחים שמספר ההתאגדויות קבוע אך זה גם לא הכרחי.

כאשר משתמשים בגישות מבוססות התאגדויות לגילוי אנומליות, נקודות בעל גודל מינימאלי שבתוך ההתאגדויות בדרך כלל לא נחשבים לאנומליים. "גודל מינימאלי" הוא פרמטר שצוין חיצונית כגון, הסף שמבוסס על התפלגות גדלי ההתאגדויות במערך נתונים.

לא כל הנקודות שמחוץ להתאגדויות צריכים להיחשב כאנומאלים. נקודה אחת יכולה להיחשב "יותר אנומאלית" מנקודה אחרת אם היא רחוקה יותר מההתאגדות הקרובה ביותר – אשר מרחק זה מוערך בדרכים שונות.

## מידע חריג במרחבים רב מימדיים

הגדלת מימדיות גורמת לסיבוכים נוספים. הסיבוכים נובעים מבחירת מידת המרחק. ייתכן שמדד המרחק האוקלידי אינו מתאים כיוון שהוא עשוי לשלב מספרים ממימדים שונים, אבל הבחירה בגורם קנה המידה המשותף אינו וודאי לכל המימדים. בניסיון להעריך איזו משתי נקודות היא קרובה יותר להתאגדות, למשל מידת המרחק שנבחר עשויה לקבוע את המנצח.

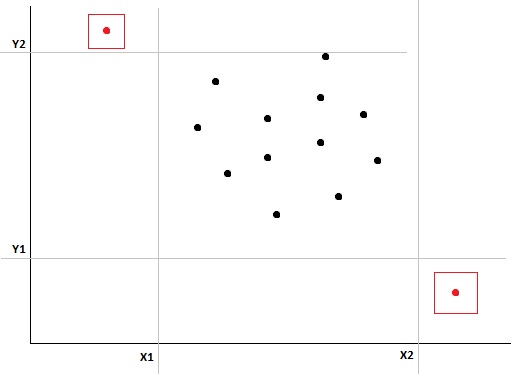
נורמליזאציה, לדוגמה, קנה מידה בהינתן מימדים שונים ככלי התאמה באותו התחום, נחוץ לעיתים, אבל מעלה את השאלה איזה סוג נורמליזציה הוא הטוב ביותר. ערכים קיצוניים עשויים לעוות את תוצאות הגישה לנורמליזציה.

## Isolation Forest

על פי המחקר נגדיר כל אירוע אבטחת מידע יוגדר כנקודה במרחב רב מימדי.

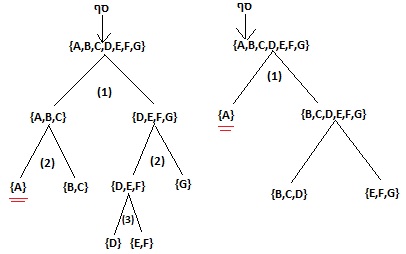
כל מרחב מייצג Feather אחר בכל משמעות שונה מהאחרת.

ניתן לראות זאת באיור הבא:



האיור מתאר מרחב דו מימדי אשר מייצג 2 פיצ'רים כגון: גודל חבילה, כמות חבילות. למעשה, כל מישור מתייחס לכל פיצ'ר שונה.

לאחר מכן, ממקמים את האירועים על עץ הפרדה בינארי מלא כפי שניתן לראות האיור הבא:



נבנה עץ הפרדה על פי השלבים הבאים:

* בוחרים פיצ'ר (מישור) מקרי.
* בוחרים ערך סף מקרי, אשר יקבל ערך בתחום מוגדר וסופי במישור שנבחר.

חוזרים חזרה על אותו התהליך עד שנוצרות קבוצות הומוגניות.

בסוף התהליך, כל עלה יחשב כקבוצה הומוגנית ולא נוכל עוד לבנות עצי הפרדה.

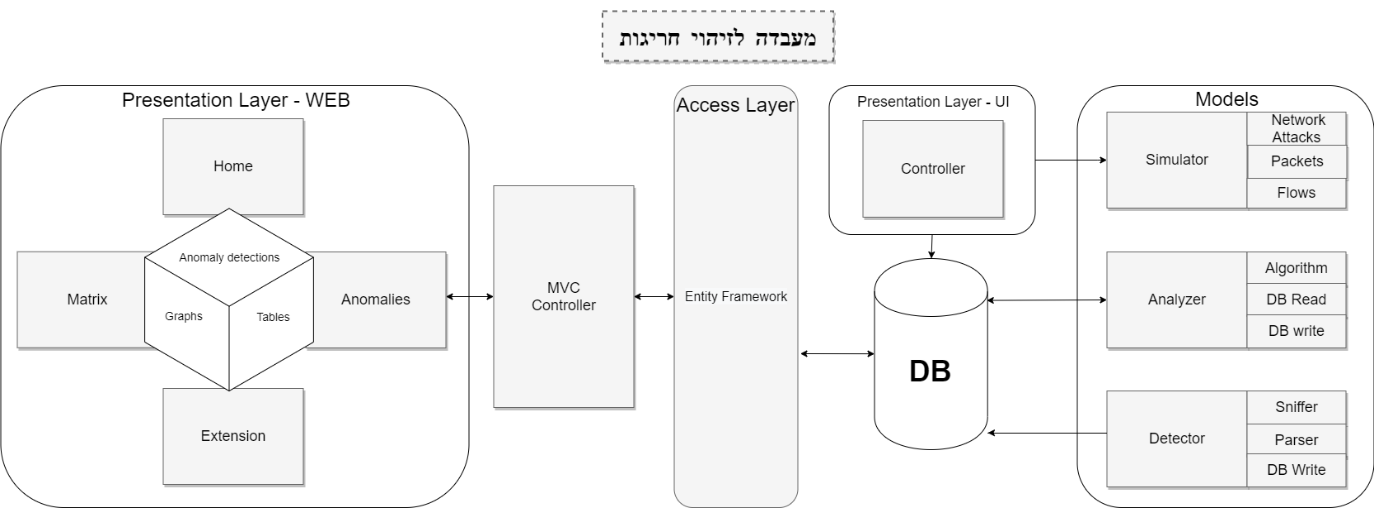
יחושב ממוצע של מספר ההפרדות לכל הנקודות ההומוגניות בהתאם כך שלכל נקודה בעץ יהיה את ממוצע מספר ההפרדות שנעשו לו.

התהליך לעיל יבוצע מחדש על מספר רב של עצים.

מסקנה: בהפרדה מקרית בממוצע הנקודות האנומליות יופרדו מהר.

# תיאור המערכת העתידית

## 4.1. High-Level



* Models – מערך המודלים במערכת: Detector, Analyzer, Simulator.

Simulator - מדמה התקפות תקשורת.

Detector - איסוף המידע התנועות ברשת - גשש שאוסף את התנועות ברשת בזמן אמת ומתעד אותם - ושליפה של ערכים להגדרת מודל לאנליזה עליו נעבוד.

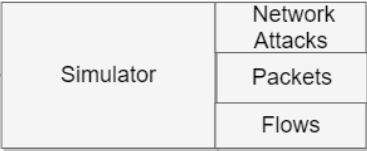
Analyzer – מציאת אנומליות ברשת על ידי שימוש אלגוריתם של למידת מכונה.

* Controller – ממשק UI להפעלת Simulator ההתקפה.
* Access Layer – שכבת קישור בין ה- DB למערכת הWEB.
* MVC Controller - תפקידו לעבד ולהגיב לאירועים המתרחשים בתצוגה, לרוב, כתגובה לפעולה של המשתמש.
* DB – 2 שכבות: שכבת RAW DATA, שכבת עיבוד המידע.
* Presentation Layer - WEB – ממשק WEB UI לצורך הצגת הנתונים.

## 4.2. Low Level

**סימולטור התקפות תקשורת**

העברת קבצים בגדלים שונים ובתדירויות שונות.

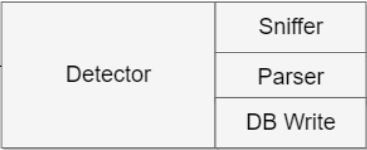
כמויות חבילות לFLOW.

סריקת פורטים.

התקפת SynFlood.

העברת חבילות גדולות לאורך זמן.

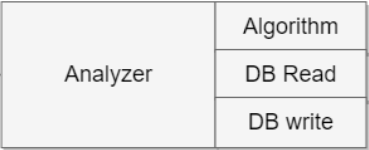
הפרש הזמנים בין יצירת הקובץ לשימוש בקובץ.

**איסוף המידע**

הסנפת חבילות בעמדת קצה.

שליחת המידע למאגר המידע ב – DB.

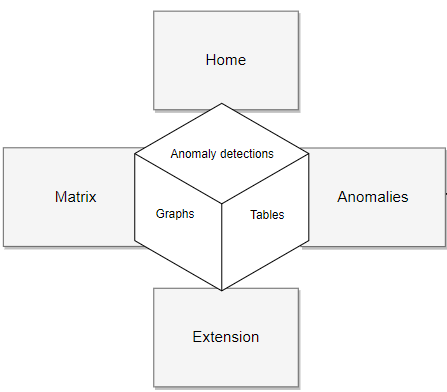
**עיבוד הנתונים**



רישום הנתונים לתוך המאגרים השונים.

עיבוד המידע לפי האלגוריתם ללימדת מכונה.

סיווג מידע חריג בהתאם למודל שנבנה.



**Presentation layer**

יצירת ממשק WEB ויזואלי לגראפים על סמך מאגרי הFeatures.

שימוש במידע שנאסף במאגרי המידע השונים.

יצירת טבלאות המציגות את החריגות שנמצאו על ידי ה - Analyzer.

# בחירת Features

## 5.1. כמות חבילות ליחידת זמן

כמות גדולה של חבילות ליחידת זמן עשויה להעיד על פעילות חריגה.

**מודל – נתונים נאספים:**

* רשימה של זמני החבילות.
* כמויות של חבילות לפי פרק זמן.

**סימולציה – אוסף נתונים:**

הרצת תעבורה בפרוטוקול: Microsoft – ds אשר משתמש בפורט 445.

פרוטוקול זה משמש להעברת קבצים בין עמדות קצה

סימולציה נורמלית:

העברת קבצים בזמן רנדומלי בין 10-120 שניות באופן תמידי.

כל קובץ ינוע בגודל שבין 10-100 Bytes.

סימולציות אנומליות:

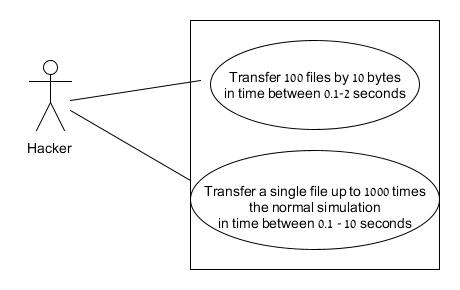
1. העברת קובץ יחיד בנפח שגדול פי 1000 מהסימולציה הנורמאלית.
2. העברת 100 קבצים בגודל של 10 Bytes בפרק זמן בין 0.1-2 שניות.

**למה זו יכולה להיחשב חריגה**

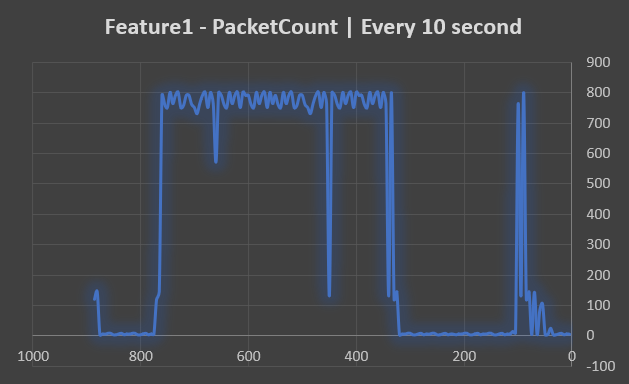
* כמות של חבילות למשך זמן ארוך מהממוצע (מכיוון שהקובץ גדול אז תהינה כמות רבה של חבילות לאורך זמן) נחשבת לחריגה מכיוון שזמן הפעילות גבוה מהזמן הממוצע.
* כמות גבוהה של חבילות בזמן קצר נחשבת לחריגה מכיוון שהכמות גבוהה מהממוצע.

**על מה היא מעידה?**

* מעידה על עומס ברשת – שיכול לגרום לאיטיות – בדרך כלל אין פעילות מסיבית, ולכן זאת תופעה פוטנציאלית שעלולה לקרות בזמן של חדירה.



חיזוי גרפי:



## 5.2. כמות הודעות ל- Flow

כמות קטנה/גדולה של חבילות בשיחה ליחידת זמן עשויה להעיד על פעילות חריגה.

**מודל – נתונים נאספים**זמן התחלת השיחה, יום התחלת השיחה, כתובת יוזם השיחה, כתובת יעד השיחה, פורט מקור, פור יעד, פרוטוקול שירות (Service), כמות החבילות הכללי בשיחה, כמות ה-Syn Flag בשיחה, כמות ה-Ack Flag בשיחה, כמות ה-Payload בשיחה, כמות ה - Fin בשיחה.

**סימולציה – אוסף נתונים**

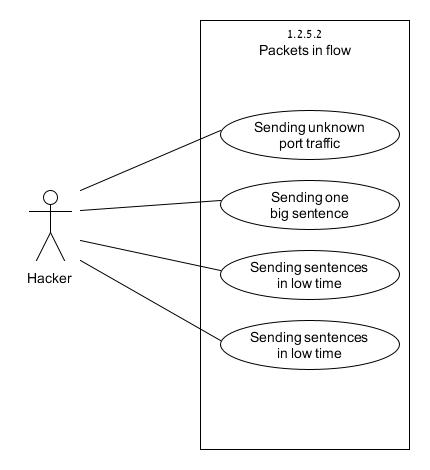
סימולציה נורמלית:

הרצת תעבורה בפרוטוקולים : 80,3389,21,23.

העברת רצפי מחרוזות בגודל 2-10 מילים אשר בגודל 1-15 אותיות בזמן רנדומלי בין 1-10 שניות באופן תמידי.

סימולציות אנומליות:

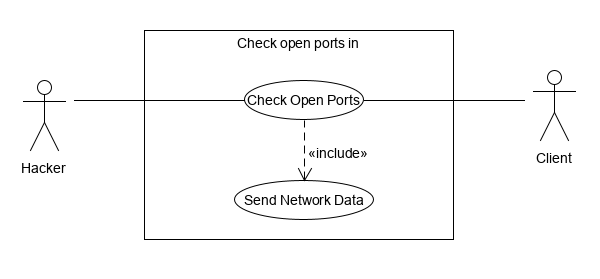
1. הרצת תעבורה בפורט שלא היה קיים קודם לכן לדוגמה פורט מספר 4444.
2. העברת רצף מחרוזות יחיד בגודל של פי 10 מהסימולציה הנורמלית.
3. העברת 100 רצפי מחרוזות בגודל של מילה יחידה בפרק זמן בין 0.1-2 שניות.
4. שליחת רצף מחרוזת יחיד בגודל של אות בודדת.

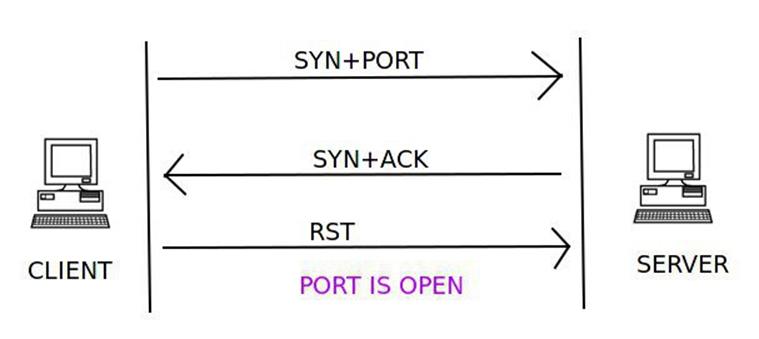


**סריקת פורטים**

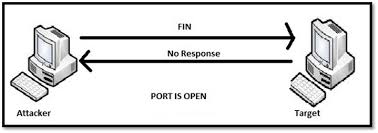
כלי שמשמש את האקר בשלב איסוף המידע על מחשב היעד לבירור אילו פורטים פתוחים וכך יכול התוקף להסיק אילו שירותי רשת רצים על המחשב ובהתאם לכך להפעיל התקפות מתאימות.

סריקת SYN – בסריקה זו המחשב הסורק מתחיל ביצירת קשר עם הפורט אבל לא מסיים את תהליך יצירת הקשר. הדבר מבוצע על ידי שליחת חבילות TCP/IP מסוג SYN, אל הפורטים הנסרקים. המחשב הנסרק משיב בחבילת SYN|ACK אם הפורט פתוח ווב־RST אם הפורט סגור. במקרה והפורט פתוח התוכנה תשלח חבילת RST על מנת שלא יוקם קשר.





סריקת FIN – בסריקה זו נשלחים אל הפורטים הנסרקים חבילות מסוג FIN. אם הפורט סגור המחשב הנסרק צפוי להחזיר חבילת RST, ואם הפורט פתוח המחשב הנסרק לא צפוי להחזיר דבר.

[](https://www.google.com/imgres?imgurl=https%3A%2F%2Fmk0resourcesinfm536w.kinstacdn.com%2Fwp-content%2Fuploads%2F052416_2153_PortScanner7.jpg&imgrefurl=https%3A%2F%2Fresources.infosecinstitute.com%2Fport-scanners%2F&docid=S28zJH-8P-M5FM&tbnid=ld169mugsmg34M%3A&vet=10ahUKEwjXkYzWk5rnAhUqUBUIHTKSDZsQMwhDKAEwAQ..i&w=719&h=254&bih=524&biw=1093&q=Fin%20Scan&ved=0ahUKEwjXkYzWk5rnAhUqUBUIHTKSDZsQMwhDKAEwAQ&iact=mrc&uact=8)

**למה זו יכולה להיחשב חריגה**

1. שימוש בפורט שלא היה בו שימוש קודם לכן עלול להיחשב כחריג מכיוון שהוא חדש.
2. העברת מחרוזת גדולה נחשבת לחריגה מכיוון שתכיל חבילה בגודל אשר גדול מהממוצע.
3. כמות גדולה של מחרוזות ליחידת זמן עשויה להעיד על פעילות חריגה מכיוון ששיחה תכיל מספר רב של חבילות.
4. כמות קטנה של חבילות עם Payload בשיחה נחשבת לחריגה מכיוון שכמות החבילות שמכילות תוכן קטנה מהממוצע.

**על מה היא מעידה**

תוקף ינסה להסוות את פעילותו על ידי שימוש מופחת/מופרז בשיחות.

חיזוי גרפי:

## 5.3. גודל (bytes) חבילה ליחידת זמן

העברת חבילות גדולות לאורך זמן עשויה להעיד על פעילות חריגה.

**מודל – נתונים נאספים**גודל החבילה ויחידת זמן.

**סימולציה – אוסף נתונים**

סימולציה נורמלית

בכל פרק זמן רנדומלי בין 10-120 שניות תעבור חבילה בגודל 10-100 bytes בין המחשבים.

סימולציות אנומליות

1. העברת חבילות בנפח שגדול פי 100 מהסימולציה הנורמאלית.
2. העברת 100 חבילות בגודל בפרק זמן בין 0.1-4 שניות אשר תתבצע באמצעות על ידי מתקפת מניעת שירות SYN Flood שמנצלת פרצת אבטחה במנגנון הקמת קשר של הTCP.

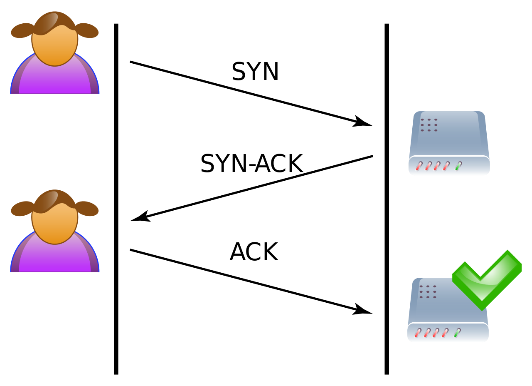
שיטת הפעולה של ההתקפה

על פי מנגנון הקמת הקשר של הTCP אשר שזהו מנגנון שנמצא בשכבת הNetwork במודל השכבות TCP\IP. משמעותו של הפרוטוקול הוא יצירת קשר אמין בין שרת ללקוח.

התקפת SYN FLOOD מנצלת פרצת אבטחה במנגנון זה.

על פי המנגנון כל לקוח אשר פונה לשרת מתחיל בחבילה המכילה את הדגל SYN דלוק.

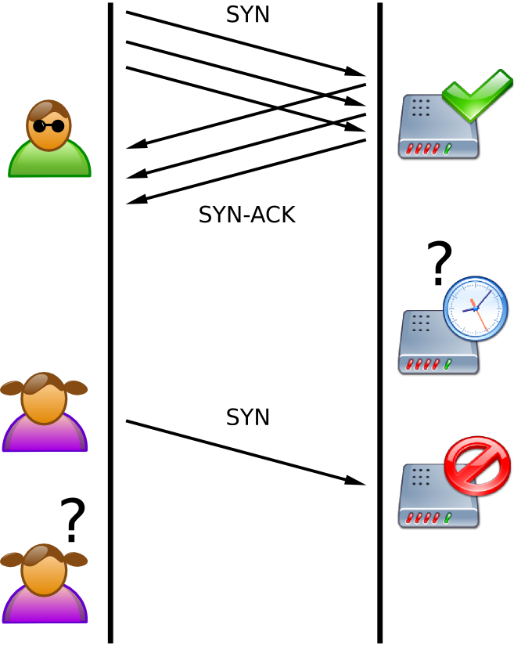
השרת עונה ללקוח בחבילה המכילה את הדגל SYN-ACK וממתין לאישור הלקוח באמצעות חבילת ACK נוספת. טרם שליחה וקבלה של שלוש החבילות האלו לא ניתן להתחיל להעביר מידע.



מתקפת SYN FLOOD מנצלת למעשה את ההמתנה שהשרת מחויב לה עד לקבלת אישור מהלקוח.

באמצעות שליחת חבילות SYN רבות , התוקף מאלץ את השרת להשאיר קשרים פתוחים רבים.

לאחרת פתיחת קשרים רבים בו-זמנית ,השרת מאבד מיכולתו להעניק שירות משום שהוא מוגבל במשאבים.



**למה זו יכולה להיחשב חריגה**

* העברת חבילות גדולות נחשבת לחריגה מכיוון שבדרך כלל לא מעבירים חבילות מלאות למשך זמן.
* כמות גבוהה של חבילות בזמן קצר נחשבת לחריגה מכיוון שהכמות החבילות תיהינה גבוהה מהממוצע.

**על מה היא מעידה?**

* מעידה על עומס ברשת – שיכול לגרום לאיטיות – בדרך כלל אין פעילות מסיבית, ולכן זאת תופעה פוטנציאלית שעלולה לקרות בזמן של חדירה.

חיזוי גרפי:

## הפרש הזמנים בין יצירת הקובץ לשימוש בקובץ.

הפרש זמנים קטן בין יצירת הקובץ עלול להעיד על קובץ חדש שנכנס למערכת ויזם תקשורת עם רכיב נוסף ברשת המקומית

**מודל – נתונים נאספים**הפרש הזמנים בין יצירת הקובץ לשימושו (בזמן מנורמל), למספר זיהוי חבילה.

**סימולציה – אוסף נתונים**

סימולציה נורמלית

הרצת תעבורה בתוכנות:

* NormalChatClient22.exe
* NormalChatClient23.exe
* NormalChatClient25.exe
* NormalChatClient80.exe

בכל 1-10 שניות ירוצו הקבצים לעיל.

סימולציות אנומליות

הרצת קובץ חדש שנרשם למערכת ומשתמש בפורט מספר 4445 להעברת מידע למחשב השני.

שם הקובץ יהיה MaskJPG.jpg אשר מכיל קובץ הרצה EXE.

**למה זו יכולה להיחשב חריגה**

* הפרש הזמנים בין יצירת הקובץ לשימוש בקובץ יכול להיחשב כחריג על סמך קטריון זמן אשר נמוך מהממוצע.

**על מה היא מעידה?**

* קובץ חדש שנוצר במחשב ויוצר תקשורת עם מחשבים אחרים ברשת, דבר אשר עלול להעיד על ניסיון התפשטות של נוזקה.

לצורך הדוגמה: תוקף או נוזקה שמנסה להשתמש בשיטת Lateral movement כדי להגיע לעוד רכיבים שונים.

חיזוי גרפי

## סיכום Features

|  |  |
| --- | --- |
| **קטגוריות** | **Features** |
| כמות חבילות ליחידת זמן | העברת חבילות בנפח שגדול פי 100 מהסימולציה הנורמאלית. |
| מתקפת SynFlood - העברת 100 חבילות בגודל בפרק זמן בין 0.1-4 שניות. |
| כמות הודעות ל- Flow | הרצת תעבורה בפורט שלא היה קיים קודם לכן לדוגמה פורט מספר 4444. |
| העברת רצף מחרוזות יחיד בגודל של פי 10 מהסימולציה הנורמלית. |
| העברת 100 רצפי מחרוזות בגודל של מילה יחידה בפרק זמן בין 0.1-2 msec |
| שליחת רצף מחרוזת יחיד בגודל של אות בודדת. |
| סריקת פורטים - שונים בין 4440-4540 |
| גודל (bytes) חבילה ליחידת זמן | העברת קובץ יחיד בנפח גדול. |
| העברת 100 קבצים בגודל של 10 Bytes בפרק זמן בין 0.1-2 שניות. |
| הפרש הזמנים בין יצירת הקובץ לשימוש בקובץ | הרצת קובץ חדש שנרשם למערכת ומשתמש בפורט מספר 4445 |

# דרישות פונקציונאליות - כללי

## איסוף המידע

**תיאור פונקציונאלי כללי:** התהליך מתרחש באופן תמידי על ידי סוכן שיושב בעמדת הקצה.

ניטור המידע יהווה האזנה לרכיב התקשורת הפועל ברשת המקומית, על ידי בדיקת חיבורים (Local Area Network).

כל חבילה מנוטרת מתפרסרת למידע המתאים למערכת, ונרשמת לאחר עיבוד למאגרי המידע "packetsCountTenSecond" ו"Flows" בשרת ה-DB. בנוסף כל חבילה נרשמת לקובץ Log File.

הקושי במימוש : הינו האזנה לתקשורת הפנימית ברשת ובחירת המידע הרלוונטי.

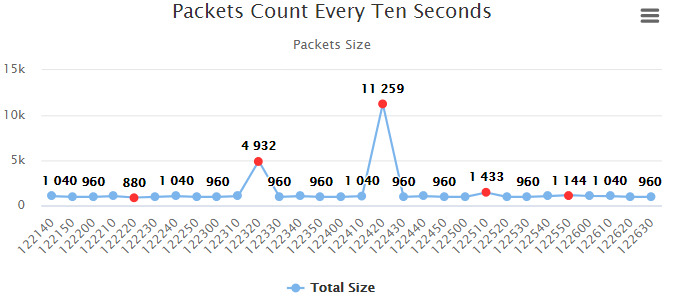
דוגמה לLog File:

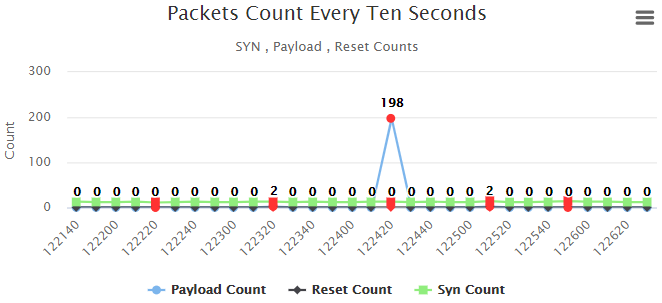
{'unixTime': 1576011466.362082, 'srcMac': '00:0c:29:9e:a3:bc', 'dstMac': '00:0c:29:42:a7:be', 'EthType': 2048, 'srcIpAddr': '192.168.222.100', 'dstIpAddr': '192.168.222.101', 'IPLen': 52, 'ID': 10450, 'Ttl': 128, 'Proto': 6, 'IPChksum': 0, 'ipProtocol': 'TCP', 'srcPort': 49816, 'dstPort': 4444, 'service': 'None', 'Seq': 3368279892, 'Ack': 0, 'dataOfs': 8, 'Flags': 'S', 'Window': 8192, 'TcpChksum': 15938, 'PID': '700', 'fileName': 'MaskJPG.exe', 'filePath': 'C:\\Windows\\Temp\\MaskJPG.exe', 'deltaTime': 0.3356020450592041}

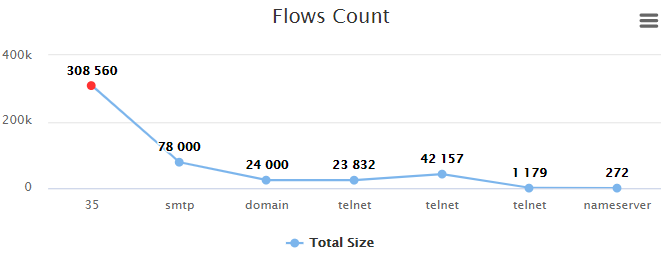
## עיבוד הנתונים

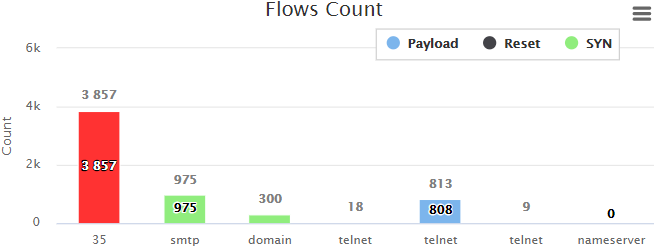
**תיאור פונקציונאלי HighLevel:** התהליך מתרחש בכל פרק זמן בשרת, על ידי הMVC Controller- אשר מעבד את הנתונים שנמצאים בתוך מאגרי המידע ה"Flows" ו""packetsCountTenSecond לתוך קבצי JSON.

הקושי במימוש: הינו הפקת DATASET מתאים בהתאם למאגרי המידע השונים, נוסף על כך יצירת Flow מתוך איסוף החבילות.





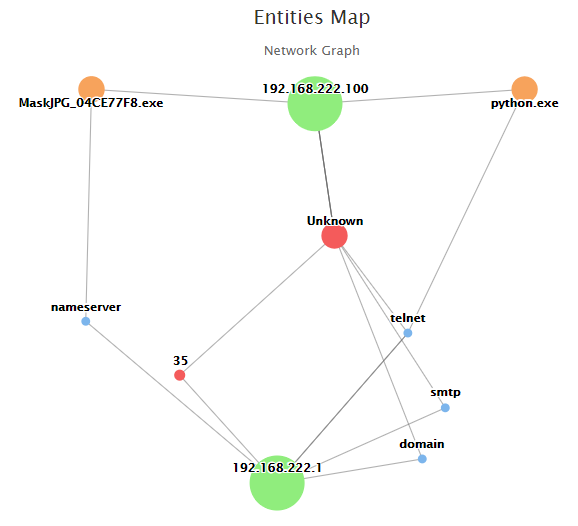




## ממשק WEB- UI

**תיאור פונקציונאלי כללי:** התהליך מתרחש בעת הכניסה למערכת אשר יהווה את מסך ה - Home.

מסך זה יציג את הDashboard הכללי של המערכת אשר יכיל את מסך היישויות, את מאגרי ה"Flows" ו""packetsCountTenSecond בצורות גרפיות. המידע יתעדכן בזמן אמת ויראה את השינויים במסכים השונים.



דוגמה לגרף היישויות:

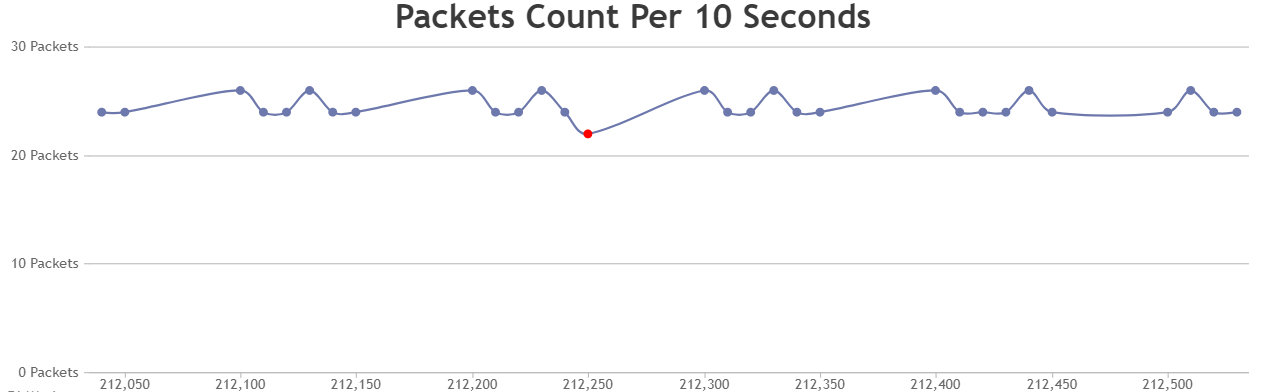
יישות ירוקה –תחנת קצה אשר מזוהה על ידי כתובת IP.

יישות כתומה – קובץ הרצה אשר שלח חבילות תקשורת לתחנה אחרת.

יישות כחולה – פרוטוקול התקשורת אשר באמצעותו נוצרה תקשורת בין תחנות קצה.

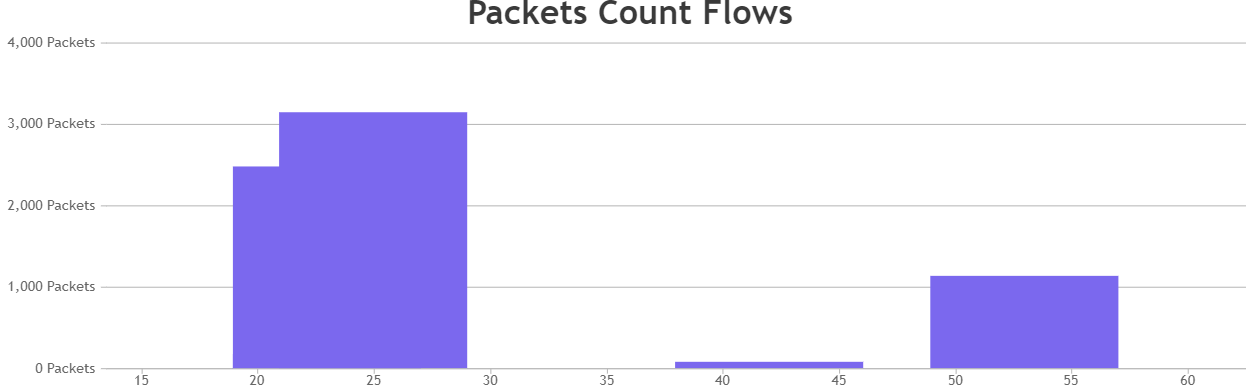
יישות אדומה – יישות המהווה אנומליה ברשת

דוגמה לגרף מספור החבילות ב10 שניות:



גרף זה מציג את כמות החבילות ליחידת זמן (בין 00:00-23:59). הנקודות הכחולות מייצגות תקשורת נורמלית לעומת הנקודות המסומנות באדום אשר מייצגות את האנומליות שנמצאו ברשת.

דוגמה לגרף מספור החבילות לFlows:



גרף זה מציג את כמות החבילות לFlow (מספר פרוטוקול תקשורת). בגרף זה יכולים להיות עמודות בשתי צבעים שונים: כחולות, אדומות. הכחולות מייצגות תקשורת נורמלית לעומת האדומות שמייצגות את האנומליות ברשת.

בתחתית המסך יהיה כפתור אשר יאפשר לעצור את סנכרון המידע לאתר, בעת לחיצתו בשנית יסונכרן המידע מחדש.

הקושי במימוש: יצירת ממשק Web אשר מושך את המידע מהמאגרים.

## זיהוי חריגות

**תיאור פונקציונאלי כללי:** התהליך מתרחש לאחר הזנת הנתונים למאגרי ה"Flows" ו""packetsCountTenSecond. שימוש באלגוריתם אשר יבדיל בין הפעולות הנורמליות לפעולות החריגות. רישום במאגרים שהוזכרו לעיל את הנתונים שהתקבלו מהאלגוריתם לזיהוי החריגות.

הקושי במימוש : בחירת מודל מתאים בהתאם לFeatures.

# דרישות פונקציונאליות - מפורט

## איסוף המידע

**תיאור פונקציונאלי כללי:** התהליך מתרחש באופן ייזום על ידי המערכת בעת עליית מערכת ההפעלה על ידי סוכן שנמצא בעמדה.

ניטור המידע יהווה האזנה לרכיב התקשורת הפועל ברשת המקומית, על ידי בדיקת חיבורים (Local Area Network).

כל חבילה מנוטרת מתפרסרת למידע המתאים למערכת, נרשמת למאגר בזיכרון אשר מחשב בכל 10 שניות את כמות החבילות, וכמות הFlags, נרשמת בזיכרון למאגר Flows ובתום 30 שניות למאגרי המידע "packetsCountTenSecond" ו"Flows" בשרת ה-DB.

בנוסף כל חבילה נרשמת לקובץ לוג file בשם ייחודי (Logfile[Unix-Time]) בעמדת הקצה.

הקושי במימוש : הינו האזנה לתקשורת הפנימית ברשת ובחירת המידע הרלוונטי.

**משתמשי התהליך:** אחראי הרשת המקומית.

**הטריגר לתהליך:** עליית מערכת ההפעלה

**הקלטים של התהליך:**

1. עליית מערכת ההפעלה
2. הסנפת המידע הכללי מהחיבור הזמין – Local Area Network.
3. פרסור המידע לפי השדות להלן:

|unixTimeMicrosec | Time | Date | Day | TimePeriod | srcIpAddr | dstIpAddr | IpLen | ID | Ttl | IPChksum | srcPort | dstPort | Service | Seq | Ack | dataOfs | Flags | Window | TcpChksum |

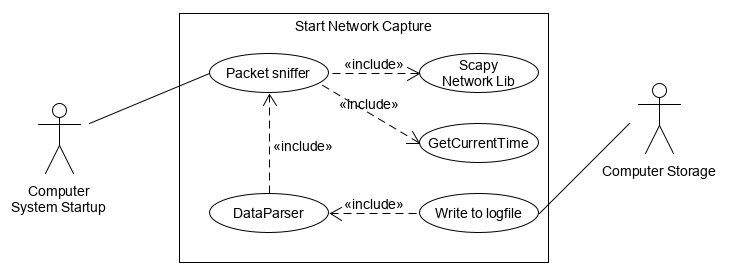
**הפלטים של התהליך:**

1. פרסור חבילות.
2. יצירת קובץ log file.
3. נרשמים כל הרשומות לקובץ לוג file בשם ייחודי (Logfile[Unix-Time]) בעמדת קצה.

**הנתונים המעורבים בתהליך:**

כתיבת נתונים למאגרי מידע רלוונטים.

UML Diagram



## עיבוד הנתונים

### Feature Packets in 10 seconds

**תיאור פונקציונאלי כללי:** התהליך מתרחש כאשר מתווספת חבילה חדשה, כתוצאה מכך מעבדים את החבילה לתוך מאגר ""packetsCountTenSecond ומונים את כמות החבילות, את הFlags השונים ואת גודל החבילות.

**משתמשי התהליך:** שרת הDB

**הטריגר לתהליך:** התהליך מתחיל כל 10 שניות שנרשמות חבילות.

**הקלטים של התהליך:**

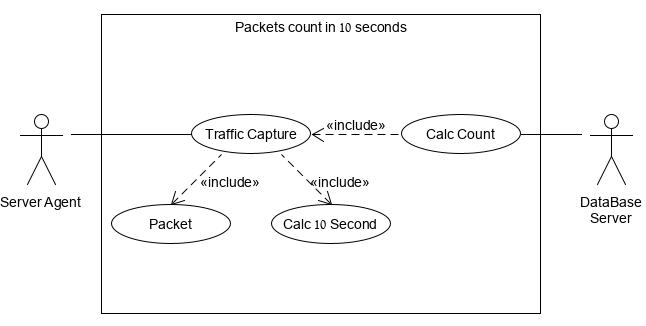
1. רישום חבילה במערכת.
2. עיבוד המידע.
3. רישום למאגר הרלוונטי.

Feature 1 – packetsCountTenSecond | 10 second priod

(PacketCountTenID | TimePeriod | startTimeUnixMillisec | endTimeUnixMillisec | Day | packetCount | counterOfSyn | counterOfAck | counterOfPa | counterOfR | counterOfRA | counterOfFin | packetsTotalSize)

**הפלטים של התהליך:**

כתיבת נתונים למאגרי מידע רלוונטים.

UML Diagram

### חישוב מספרי בFLOWS

**תיאור פונקציונאלי כללי:** התהליך מתרחש בכל חבילה שעוברת כאשר מונים ומעדכנים את הFlow בהתאם לפרוטוקול תקשורת, ליוזם השיחה ולפורטים הרלוונטים בשיחה. אם מאגר הנתונים Flows ריק, נוסיף אליו את החבילה האחרונה שהתקבלה.

אחרת, נאסוף את החבילות הקשורות לFlow נמנה את כמויותהם:

הSynFlag,AckFlag,PayloadFlag,FinFlag ואת גודל הFlow, ולאחר החבילה האחרונה במאגר ה - Flows נוסיף את כמות החבילות.

זיהוי הFlow יתבצע על ידי בדיקת יוזם השיחה – srcIpAddr, הSynFlag ,סיום השיחה תתבצע על ידי בדיקת FinFlag כאשר הSrcPort שווה לSrcPort או לdstPort שהיה קודם לכן.

* כאשר מתקבלת חבילה עם SynFlag – סימן שהתחיל ה – Flow בהינתן srcPort שונה.
* כאשר מתקבלת חבילה עם FinFlag – סימן שה - Flow הסתיים בהינתן srcPort שונה.

כל תאריך שבו מתבצע ניטור הרשת נרשם למאגר הימים והשעות לצורך מעקב.

**משתמשי התהליך:** שרת ה DB

**הטריגר לתהליך:** בכל 30 שניות שחבילה חדשה שמתקבלת במערכת.

**הקלטים של התהליך:**

1. רצף חבילות.
2. עיבוד המידע לפי המאגרים שלהלן:

Feature 2 – ServiceName | CountOfPackets and Flags counting

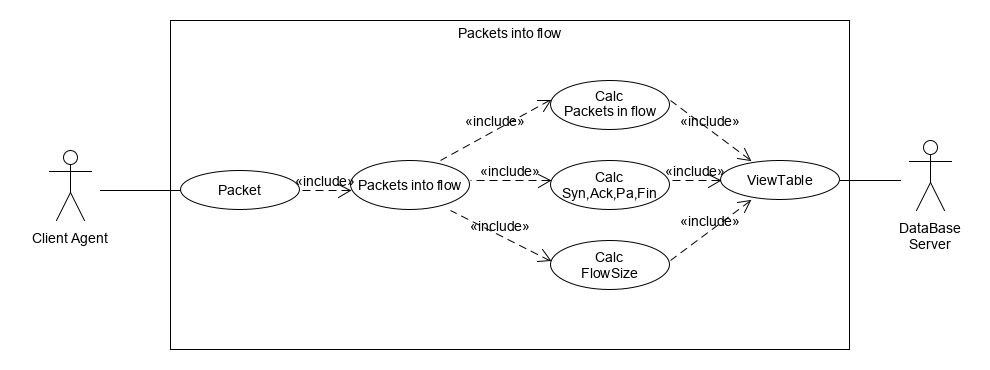
(FlowID | UnixStartTimeMillisec | UnixEndTimeMillisec | Time | Day | TimePeriod | srcIpAddr | dstIpAddr | Service| srcPort | dstPort | counterOfPackets | counterOfSyn | counterOfPa | counterOfR | counterOfRA | counterOfFin | packetsTotalSize)

**הפלטים של התהליך:**

כתיבת נתונים למאגרי מידע רלוונטים.

**הנתונים המעורבים בתהליך:**

המידע המפורסר של החבילה.

UML Diagram

### מציאת קבצי הרצה

**תיאור פונקציונאלי כללי:** התהליך מתרחש באופן ייזום על ידי המערכת בעת עליית מערכת ההפעלה על ידי סוכן שנמצא בעמדה.

לכל חבילה מנוטרת, מזהים אם ישנה שיחה במצב ESTABLISHED לפי מספר srcPort ו- dstPort וגם בהתאם כתובות ה srcIpAddr,dstIpAddr. לכל שיחה כזו שולפים את הPID ולפיו מקבלים את שם הקובץ שהורץ ואת ניתובו בעמדת הקצה.

בנוסף לכך, מחשבים את הפרש הזמנים בין זמן יצירת הקובץ לזמן ההרצה של הקובץ.

**משתמשי התהליך:** שרת הDB

**הטריגר לתהליך:** עליית מערכת ההפעלה

**הקלטים של התהליך:**

פרסור המידע לפי השדות להלן:

| FileID | TimePeriod | startTimeUnixMillisec | endTimeUnixMillisec | Day | packetCount | counterOfSyn | counterOfAck | counterOfPa | counterOfR | counterOfRA | counterOfFin | packetsTotalSize | PID | filename | filePath | deltaTimeMillisec|

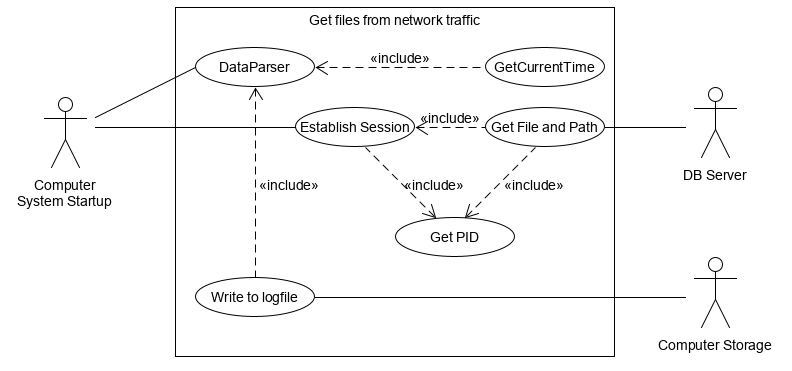
**הפלטים של התהליך:**

1. פרסור חבילות.
2. רשימה לקובץ log file.
3. נרשמים כל הרשומות לקובץ לוג file בשם ייחודי (Logfile[Unix-Time]) בעמדת קצה.

**הנתונים המעורבים בתהליך:**

כתיבת נתונים למאגרי מידע רלוונטים.

UML Diagram



### חישוב מטריקות

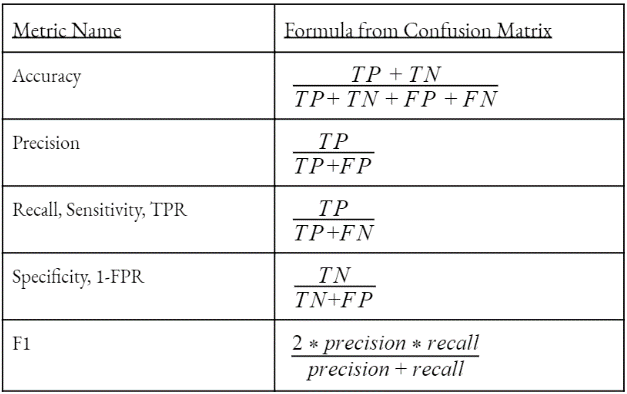
**תיאור פונקציונאלי כללי:** התהליך מתרחש בעת כניסה לדף ה-[Matrix](http://localhost:56482/Home/Matrix) בממשק ה – Web UI. דף זה יכיל 2 טבלאות: Confusion Matrix, Performance Measure.

Confusion Matrix – טבלה המציגה בצורה כמותית את הדיוק והשגיאות במערכת על ידי הפרמטים הבאים:

|  |  |  |
| --- | --- | --- |
|  | SimulatorAttack + | SimulatorAttack - |
| Anomaly Detection + | TP | FP |
| Anomaly Detection - | FN | TN |

* TP - True Positive: חריגות שנמצאו בנתונים אכן תואמות בדיוק את החריגות בתהליך.
* FP - False Positive: התהליך ממשיך להיות תקין, אך נצפים ערכי נתונים לא צפויים, למשל המערכת זיהתה חדירה כשלמעשה אין חריגה אמיתית.
* FN - False Negative: התהליך הופך להיות חריג, אך ההשלכות אינן רשומות כנתונים הלא תקינים, למשל, הייתה חריגה במערכת אך המערכת לא זיהתה את החריגה.
* TN -: לא נמצאו חריגות בנתונים אשר גם לא נמצאו בתהליך.

Performance Measure – טבלה המציגה באחוזים את הנכונות והדיוק במערכת על ידי הפרמטרים הבאים:



הקושי במימוש: שליפת מידע ממאגרי המידע השונים, וחישובם על פי הנוסחאות.

**משתמשי התהליך:** משתמש המערכת.

**הטריגר לתהליך:** התהליך מתחיל בעת כניסה לדף ה-[Matrix](http://localhost:56482/Home/Matrix).

**הקלטים של התהליך:**

מידע לפי המאגרים שלהלן:

Simulator Attack, Anomaly בטבלת כמות חבילות ליחידת זמן, ובטבלת כמות הודעות ל- Flow.

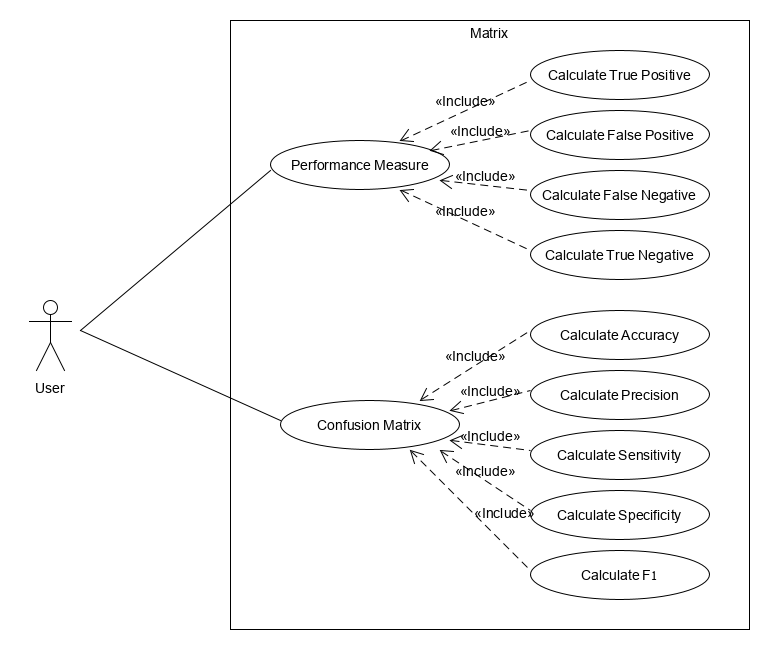
**הפלטים של התהליך:**

הצגת טבלאות רלוונטיות : Confusion Matrix, Performance Measure.

**הנתונים המעורבים בתהליך:**

מאגרי מידע : packetsCountTenSecond , Flows.

UML Diagram



## ממשק WEB - UI

### מערכת הWEB

**תיאור פונקציונאלי כללי:** התהליך מתרחש בעת כניסה לממשק WEB של המערכת.

מסך הDashboard יכיל את מאגרי ה ה"Flows" ו""packetsCountTenSecond בצורות גרפיות.

בנוסף לכך, מסך הDashboard יכיל את טבלת ה"חריגים"

לצד המסך יהיה כפתור למסך כלל הגרפים במערכת, בעת לחיצתו יפתח מסך הגרפים אשר ישקף את הFeature-ים שנבחרו.

הקושי במימוש: יצירת ממשק Web שמושך את המידע מהמאגרים השונים.

**משתמשי התהליך:** אחראי הרשת המקומית.

**הטריגר לתהליך:** התהליך מתחיל בעת כניסה למערכת.

**הקלטים של התהליך:**

מידע לפי המאגרים שלהלן:

כמות חבילות ליחידת זמן, כמות הודעות ל- Flow , גודל (bytes) חבילה ליחידת זמן, הפרש הזמנים בין יצירת הקובץ לשימוש בקובץ.

**הפלטים של התהליך:**

הצגת מאגרי מידע רלוונטים.

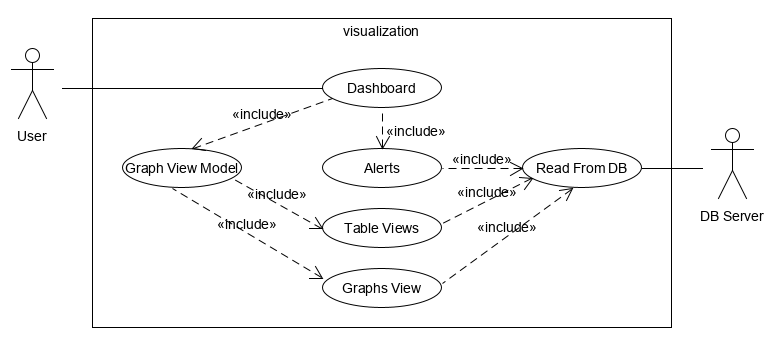
גרפים מופקים על ידי המאגרים.

הצגת טבלת "חריגים" על ידי שימוש בטבלת ה-"Flows" וה-""packetsCountTenSecond

**הנתונים המעורבים בתהליך:**

מאגרי : packetsCountTenSecond , Flows.

UML Diagram



### עצירת/ משיכת תזרים מידע לWEB

**תיאור פונקציונאלי כללי:** התהליך מתרחש בעת לחיצה על כפתור "Pause Traffic"/"Resume Traffic" בממשק ה-WEB של המערכת.

בעליית הממשק יאותחל resumeFlag על ידי המספר 0 אשר מייצג שלא התבצעה לחיצה על לחצן ה-"Pause Traffic" ובכך ממשיכים למשוך מידע לממשק ה-WEB והגרפים בדף התעדכנו בהתאם. בעת לחיצה על כפתור ה – "Pause Traffic", resumeFlag יקבל את הספרה 1 ובכך נעצר תזרים המידע בממשק ה-WEB והגרפים יפסיקו להתעדכן, התוכן של הכפתור ישתנה בהתאם ל "Resume Traffic".

הקושי במימוש: יצירת לחצן שמעדכן בהתאם את הגרפים השונים.

**משתמשי התהליך:** אחראי הרשת המקומית.

**הטריגר לתהליך:** התהליך מתחיל בעת לחיצה על כפתור "Pause Traffic".

**הקלטים של התהליך:**

מידע לפי המאגרים שלהלן:

כמות חבילות ליחידת זמן, כמות הודעות ל- Flow , גודל (bytes) חבילה ליחידת זמן.

**הפלטים של התהליך:**

הצגת גרף ישויות לפי מאגרי המידע השונים.

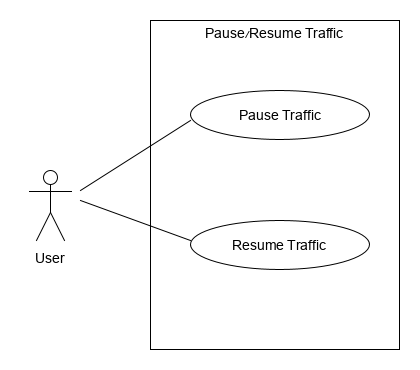
הצגת מאגרי מידע רלוונטים.

גרפים מופקים על ידי המאגרים.

**הנתונים המעורבים בתהליך:**

מאגרי : packetsCountTenSecond , Flows.

UML Diagram



## זיהוי חריגות

**תיאור פונקציונאלי כללי:** התהליך מתרחש בכל פרק זמן קבוע של 30 שניות בצד השרת. התהליך יסמן את הערך "1" בכל רשומה שבדק כדי שבכל פעם שירוץ הוא יעבור רק על רשומות עם שדות ריקים - Null, בנוסף התהליך יסווג כל רשומה בשדה "Anomaly" אם היא חריגה או לא לפי המודל שנבחר ועל סמך ה - Features שנבחרו לעיל. במידה והאלגוריתם יזהה התקפה יסומן הערך "1" בשדה "Anomaly". הסוכן יעשה שימוש באלגוריתם אשר יבדיל בין הפעולות הנורמליות לפעולות החריגות. לבסוף ירשום במאגר ה"Flows" ו""packetsCountTenSecond את הנתונים שהתקבלו מהתהליך לזיהוי החריגות.

**משתמשי התהליך:** סוכן בצד השרת.

**הטריגר לתהליך:** בכל פרק זמן של 30 שניות.

**הקלטים של התהליך:**

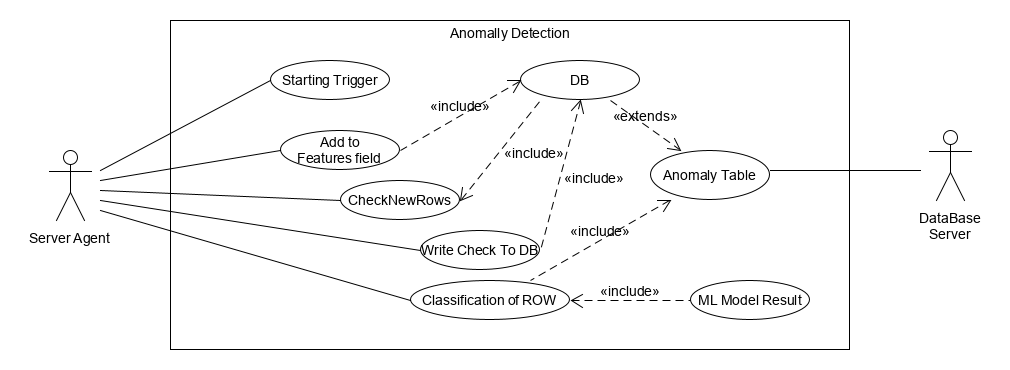
1. נתונים ממאגרי packetsCountTenSecond
2. נתונים ממאגרי Flows
3. נתונים ממאגרי הלמידה

**הפלטים של התהליך:**

כתיבת נתונים למאגר ה"Flows" ו""packetsCountTenSecond.

**הנתונים המעורבים בתהליך:**

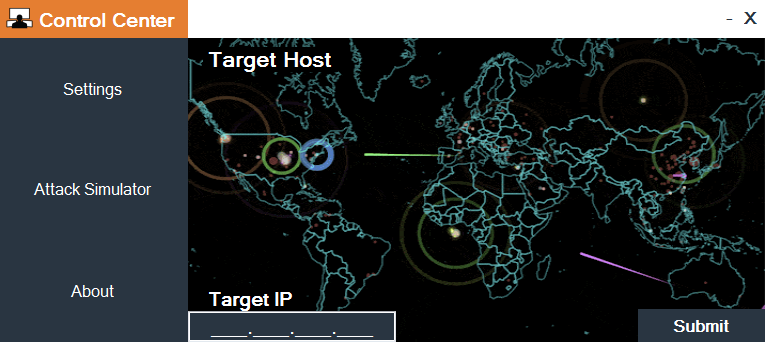
1. נתונים ממאגרי packetsCountTenSecond
2. נתונים ממאגרי Flows
3. נתונים ממאגרי הלמידה

UML Diagram

## סימולטור התקפה

**תיאור פונקציונאלי כללי:** התהליך מתרחש בעמדת קצה. הסימולטור יהווה ממשק GUI אשר יכיל בתוכו 3 חלקים שונים:

* Settings: פאנל ההגדרות של הסימולטור אשר המשתמש מגדיר כתובת IP כיעד תקיפה.



* Attack Simulator: פאנל המכיל התקפות שונות המבוססות על ה – Features שנבחרו.

Syn Flood – התקפת מניעת שירות:

**הצפת SYN** היא סוג של התקפת מניעת שירות המנצלת פרצת אבטחה במנגנון הקמת הקשר של ה- TCP. במסגרת ההתקפה התוקף שולח רצף של חבילותSYN  למחשב המותקף ובכך מאלץ אותו לפתוח במקביל חיבורים רבים עד שלא נותרים לו משאבים לקבלת חיבורים חדשים. בכך התוקף מונע מהמחשב המותקף להעניק שירות למשתמשים אחרים ומשיג את מבוקשו.

Port Scan – כלי לסריקת פורטים:

היא תוכנת אבטחת מידע, המשמשת לבירור אילו פורטים פתוחים במחשב מסוים המחובר לרשת מחשבים. הסריקה מבוצעת בדרך כלל על ידי האקר בשלב איסוף המידע על מחשב היעד או איש אבטחת מידע כדי לדאוג לסגור פורטים פתוחים. באמצעות רשימת הפורטים הפתוחים יכול התוקף להסיק אילו שירותי רשת רצים על המחשב הנסרק ובהתאם לכך להפעיל התקפות מתאימות.

File Transfer – העברת מספר קבצים גדולים על ידי פרוטוקול SMB:

הפרוטוקול SMB‏(Server Message Block) פועל בשכבת היישום השכבה העליונה של מודלOSI ומשמש בעיקר כדי לספק גישה משותפת אל קבצים, מדפסות, יציאות טוריות, ותקשורת בין מחשבים ברשת. הפרוטוקול מספק גם תקשורת בין תהליכית עם מנגנון הרשאות המאפשר את ירושתן. רוב השימוש של SMB, הוא במחשבים המריצים חלונות, שבו הוא ידוע לעיתים קרובות כ"שכנים ברשת".

Unknown Port – העברת חבילות בפורט שנקבע על פי Value

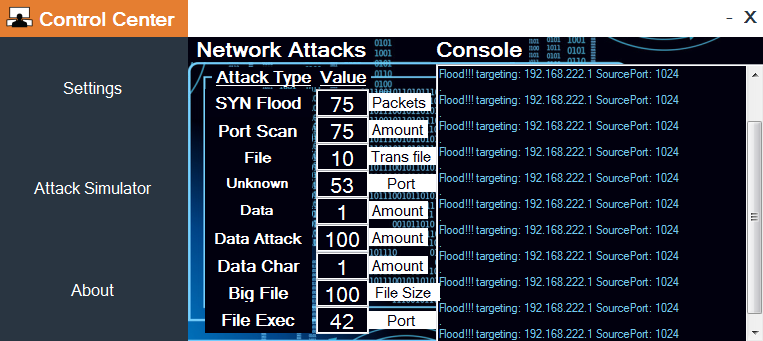
Data Payload – העברת חבילה עם מידע רב

Data Attack – העברת חבילות עם מידע על פי ה Valueשנקבע

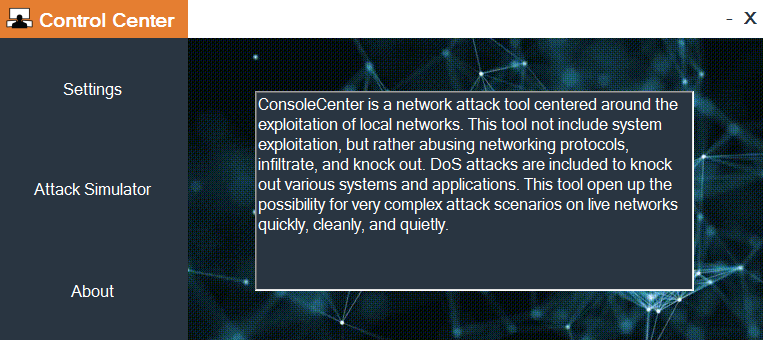
Data Char – העברת חבילה עם מספר אותיות על פי שנקבע בValue

Big File – העברת קובץ גדול

File Exec – יצירת קובץ והרצתו אשר שולח חבילות לכתובת יעד



* About: פאנל הסבר על הסימולטור:



כל התקפה שנמצאת בסימולטור תוקפת את כתובת היעד שנשמרה בSettings. לכל מתקפה יש ערך שהמשתמש ממלא. בעת לחיצה על כפתור התקפה, תרוץ ההתקפה בהתאם מאחורי הקלעים עם הערך שמילא המשתמש. לאחר ההתקפה הסימולטור יסמן את הערך "1" בכל רשומה שהייתה בזמני התקיפה תחת שדה בשם "SimulatorAttack" במאגרי המידע השונים - ה"Flows" ו""packetsCountTenSecond.

**משתמשי התהליך:** משתמש המערכת.

**הטריגר לתהליך:** בעת לחיצה על לחצן ההתקפה.

**הקלטים של התהליך:**

1. נתונים ממאגרי packetsCountTenSecond
2. נתונים ממאגרי Flows

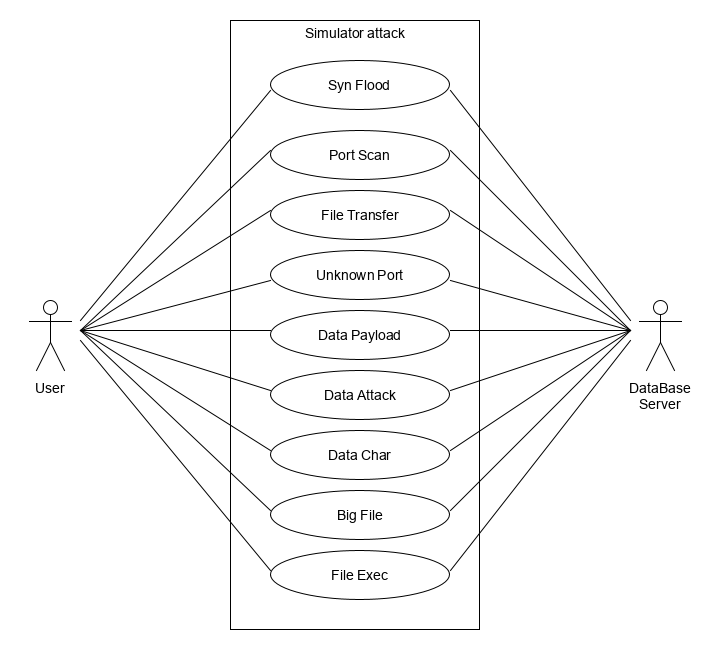
**הפלטים של התהליך:**

1. ייצוג נתונים על מסך ה – Simulator.
2. כתיבת נתונים למאגר ה"Flows" ו""packetsCountTenSecond.

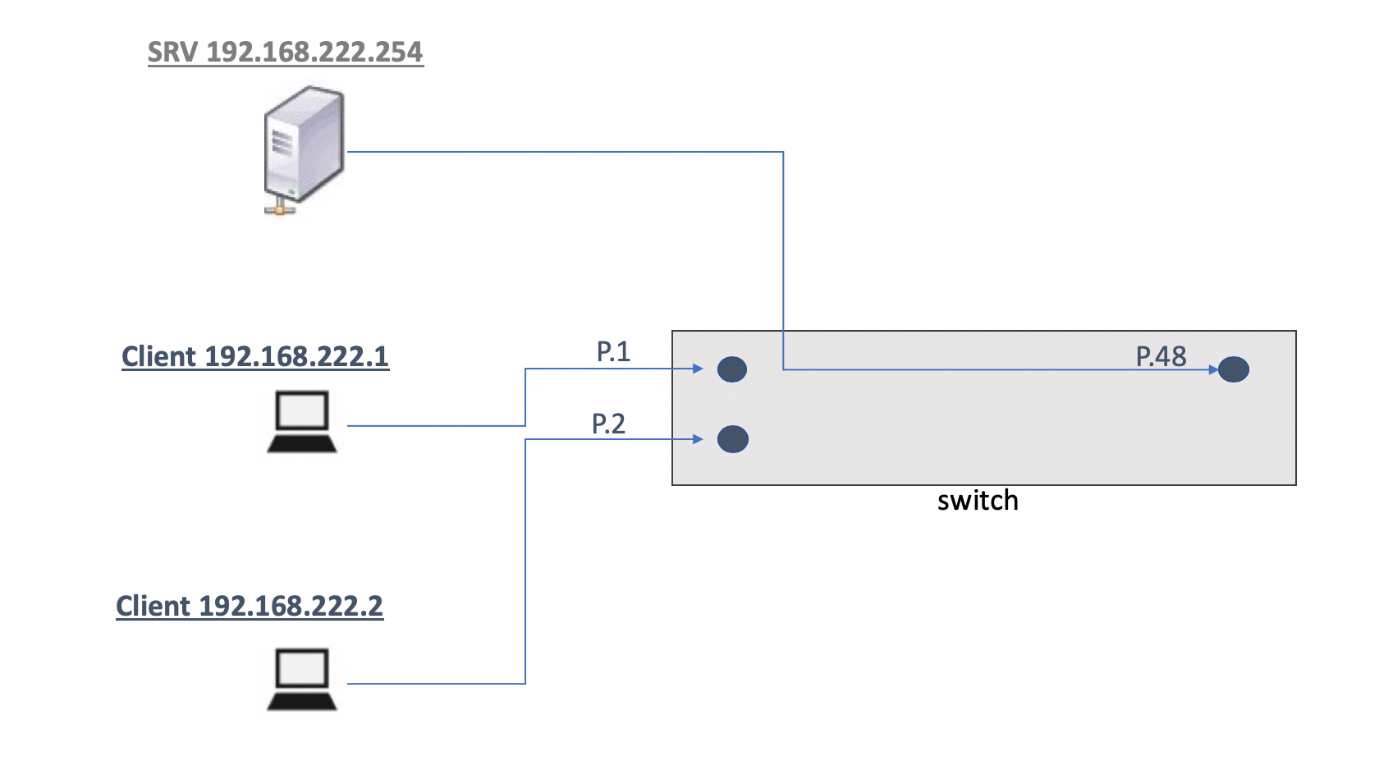
**הנתונים המעורבים בתהליך:**

1. נתונים ממאגרי packetsCountTenSecond.
2. נתונים ממאגרי Flows.

UML Diagram



# תיאור ארכיטקטורה הכללית של המערכת



המערכת ממוקמת על שרת Windows Server 2019 עם Microsoft sql server 2014 ותתבצע תקשורת עם עמדות הקצה הממוקמות אצל הלקוח דרך הרשת הפנימית (LAN).

צד לקוח:

תהליך איסוף המידע יתחיל בצד הלקוח על ידי הסנפת חבילות בזמן אמת.

יצירת סימולציות (נורמליות, לא נורמליות) יבוצעו בצד הלקוח.

תהליך איסוף המידע יבוצע בשפת Python על ידי ספרייה שנקראתScapy .

צד שרת:

יכיל DB עם מאגרי מידע בהתאם לדרישות הפונקציונאליות ויאכלס בתוכו את המידע שנאסף על ידי הלקוח.

בנוסף, השרת יאפשר שירות WEB לויזואליות של המערכת.

שפת התכנות שבה נשתמש בשרת היא C# בטכנולוגית MVC, Python והDB שנשתמש בו יהיהMSSQL 2014.



# הדרישות הלא פונקציונאליות

רכיבי מחשוב:

* 2 עמדות לצורך ביצוע הסימולציות.
* שרת אשר מסוגל לעבד נתונים בצורה מהירה.
* מתג תקשורת בכדי לאפשר תעבורה בין כלל הרכיבים של המערכת.

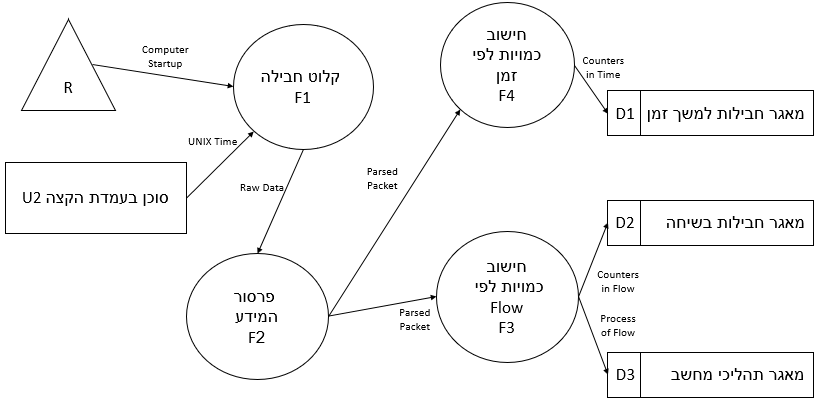
זמני תגובה:

* המערכת תצטרך לעמוד בכמות מידע מאסיבי.
* יכולות עיבוד גבוהות על מנת לעמוד בעומס החבילות.

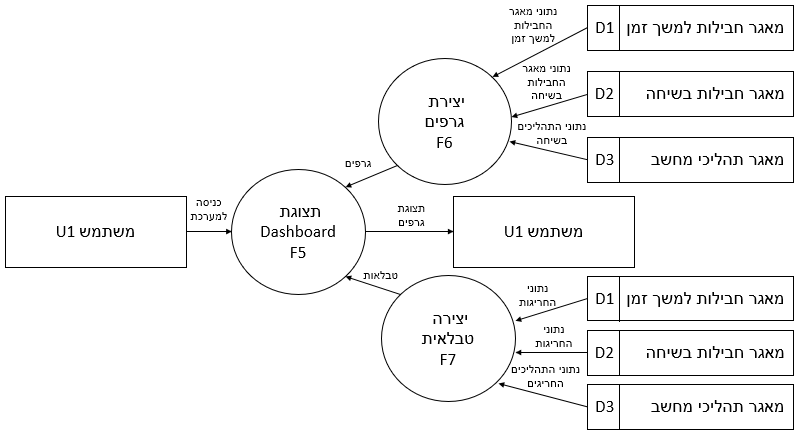
# פעולות שיתבצעו באופן שוטף

## האזנה

* + 1. בעת עליית המחשב ירוץ סוכן אשר יקלוט חבילות היוצאות ונכנסות אל המחשב.
    2. קליטה החבילה תעבור תהליך של פרסור המידע הגולמי למידע רלוונטי.
    3. בכל 10 שניות יחושבו כמויות של חבילות וירשם למאגר החבילות למשך זמן D1.
    4. כל חבילה שנכנסת לתהליך הקליטה תתמיין לFLOW, יחושבו מספרי החבילות השייכות לכל FLOW וירשמו למאגר חבילות בשיחה D2.
    5. כל חבילה ששייכת לFLOW תרשם למאגר תהליכי המחשב D3.

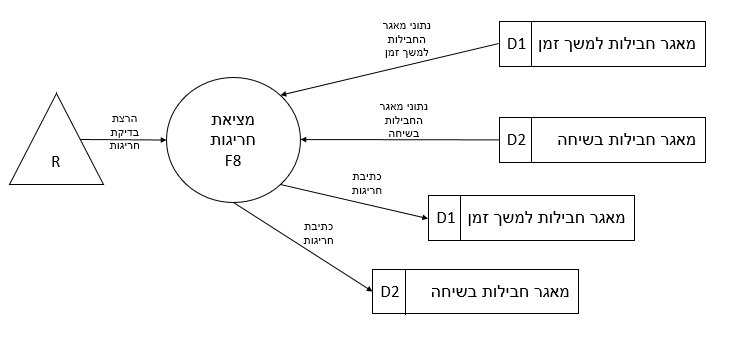


* 1. תצוגת Dashboard
     1. משתמש U1 נכנס לממשק המערכת.
     2. המערכת תעלה בממשק הDashboard F5.
     3. במסך הDashboard תוצג תצורת גרפים:
        1. שליפת נתונים מתוך מאגר חבילות למשך זמן D1.
        2. שליפת נתונים מתוך מאגר חבילות בשיחה D2.
        3. שליפת נתוני התהליכים מתוך מאגר תהליכי המחשב D3
     4. במסך הDashboard תוצג טבלת החריגות:
        1. שליפת חריגות מתוך מאגר חבילות למשך זמן D1.
        2. שליפת חריגות מתוך מאגר חבילות בשיחה D2.
        3. שליפת נתוני התהליכים החריגים ממאגר D3.



## מציאת חריגות

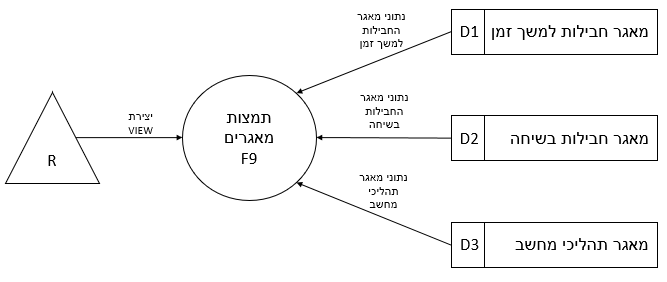
* + 1. הרצת אלגוריתם למציאת חריגות.
    2. משיכת נתונים ממאגר החבילות למשך זמן D1.
    3. משיכת נתונים ממאגר החבילות בשיחה D2.
    4. רישום תוצאות במאגר החבילות למשך זמן D1.
    5. רישום תוצאות במאגר החבילות בשיחה D2.



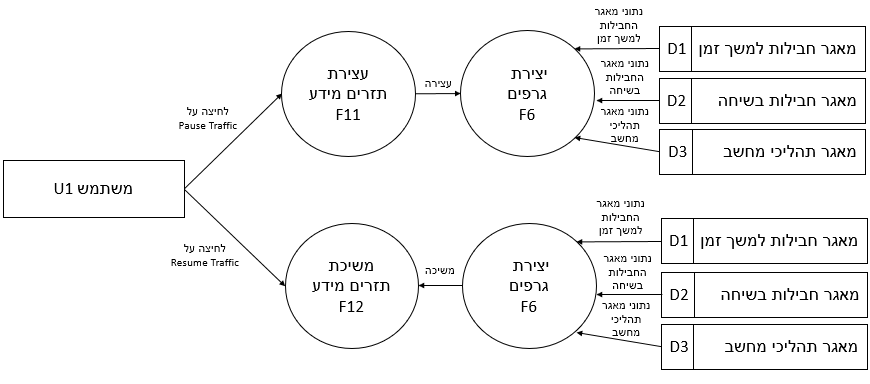
## הצגת נתונים בממשק WEB - UI

תמצות מאגרי מידע להצגה גרפית

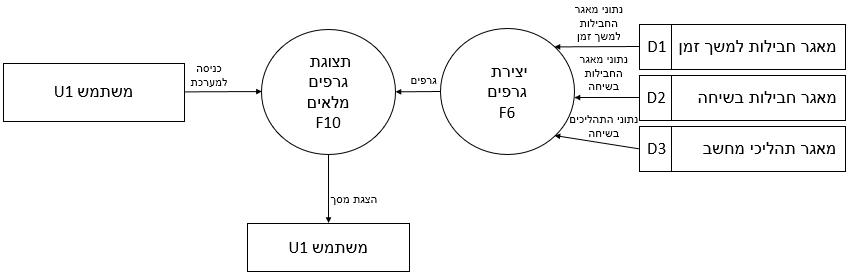
* + 1. שולפים ומתמצתים את הנתונים ממאגר החבילות למשך זמן D1 לפי זמן ביום.
    2. שולפים ומתמצתים את הנתונים ממאגר החבילות בשיחה D2 לפי זמן ביום.
    3. שולפים ומתמצתים את הנתונים ממאגר תהליכי המחשב D3.



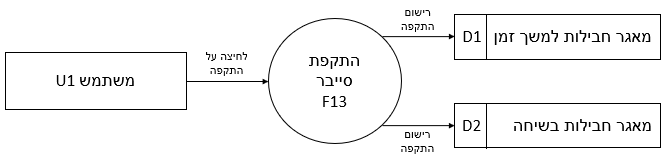
* 1. עצירה / משיכה של תזרים מידע למסך הDashboard
     1. משתמש U1 לוחץ על לחצן ה"Pause Traffic":
        1. עצירה של תזרים מידע מתוך מאגר מידע D1.
        2. עצירה של תזרים מידע מתוך מאגר מידע D2.
        3. עצירה של תזרים מידע מתוך מאגר מידע D3.
     2. משתמש U1 לוחץ על לחצן ה"Resume Traffic":
        1. משיכה של תזרים מידע מתוך מאגר מידע D1.
        2. משיכה של תזרים מידע מתוך מאגר מידע D2.
        3. משיכה של תזרים מידע מתוך מאגר מידע D3.



* 1. Web extension
     1. משתמש U1 נכנס לממשק ה- Extension.
     2. יצירת גרפים לפי הנתונים:
        1. שליפת נתונים מתוך מאגר חבילות למשך זמן D1.
        2. שליפת נתונים מתוך מאגר חבילות בשיחה D2.
        3. שליפת נתונים מתוך מאגר תהליכי מחשב D3.
     3. הצגת מסכים למשתמש המערכת U1.



* 1. התקפת סייבר
     1. משתמש U1 נכנס לממשק הסימולטור התקפות.
     2. לחיצה על אחד מהכפתורים בסימולטור יתקוף את כתובת היעד בהתאם לבחירה.
     3. לאחר ההתקפה ירשם המידע:
        1. במאגר חבילות למשך זמן D1.
        2. במאגר חבילות בשיחה D2.



# מילון נתונים

**טבלת טרנזקציות**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' | שם | סוג | תיאור מילולי |
| 9.1 | האזנה | זמן אמת | הטרזנקציה מתארת את תהליך איסוף הנתונים מהעמדות קצה. |
|
| 9.2 | תצוגת Dashboard | משתמש | הטרנזקציה מתארת את תהליך הצגת המידע המלא במסך הראשי של המערכת. |
|
| 9.3 | מציאת חריגות | זמן אמת | הטרנזקציה מתארת את תהליך מציאת החריגות על סמך הנתונים שנאספו. |
| 9.4 | הצגת נתונים בממשק WEB - UI | משתמש | הטרנזקציה מתארת את תהליך הנפקת מידע רלוונטי על סמך הנתונים שנאספו. |
| 9.5 | עצירה / משיכה של תזרים מידע למסך הDashboard | משתמש | הטרנזקציה מתארת את תהליך עצירת / משיכת המידע במסך הראשי של המערכת. |
| 9.6 | Web extension | משתמש | הטרנזקציה מתארת את תהליך הצגת המידע המלא במסך המשני של המערכת. |
| 9.7 | התקפת סייבר | משתמש | הטרנזקציה מתארת את תהליך ההתקפה על כתובת היעד. |

**טבלת פונקציות**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' פונקציה | מס' טרנזקציה | שם פונקציה | תיאור קצר של הפונקציה |
| 1 | **9.1** | קלוט חבילה | פונקציה שמריצה סוכן אשר קולט חבילות שיוצאות ונכנסות אל המחשב. |
| 2 | **9.1** | פרסור המידע | פרסור המידע הגולמי בחבילה למידע הרלוונטי |
| 3 | **9.1** | חישוב כמויות לפי flow | פונקציה אשר מחשבת כמויות של חבילות בכל 10 שניות. |
| 4 | **9.1** | חישוב כמויות לפי זמן | כל חבילה שנכנסת לתהליך הקליטה תתמיין לflow ,יחושבו מספרי החבילות השייכות לכל flow. |
| 5 | **9.2** | תצוגת dashboard | המערכת תעלה בממשק הdashboard |
| 6 | **9.2, 9.5, 9.6** | יצירת גרפים | במסך הdashboard תוצג תצורת גרפים |
| 7 | **9.2** | יצירת טבלאות | שליפת נתונים מתוך מאגר החריגות למסך הdashboard |
| 8 | **9.3** | מציאת חריגות | פונקציה שמושכת נתונים ממאגר חבילות למשך זמן וממאגר חבילות בשיחה ורישום תוצאות חריגות. |
| 9 | **9.4** | תמצות מאגרים | כתיבת נתונים מתומצתת ממאגר חבילות למשך זמן ומאגר חבילות בשיחה. |
| 10 | **9.5** | תצוגת גרפים מלאה | פונקציה שמציגה מסכים למשתמש המערכת. |
| 11 | **9.6** | עצירת תזרים מידע | עצירת שליפת מידע למסך הDashboard |
| 12 | **9.6** | משיכת תזרים מידע | שליפת מידע למסך הDashboard |
| 13 | **9.7** | התקפת סייבר | ביצוע התקפות סייבר על כתובת יעד |

**טבלת מאגרי נתונים בטרנזקציות**

|  |  |  |
| --- | --- | --- |
| מס׳ מאגר | מס' טרנזקציה | תפקיד המאגר בטרנזקציה |
| 1 | **9.1** | חישוב כמויות חבילות לפי זמן |
| 2 | **9.1** | מאגר שבו כל חבילה שנכנסת תתמיין לflow ויחושב מס׳ חבילות בכל flow |
| 3 | **9.1** | מאגר שמכיל את התהליכים שביצעו את העברת המידע ברשת |
| 1 | **9.2** | משיכה ורישום נתוני החבילות למשך זמן |
| 2 | **9.2** | משיכה ורישום נתוני מאגר החבילות בשיחה |
| 3 | **9.2** | משיכה ורישום נתוני התהליכים בFLOW |
| 1 | **9.3** | משיכה ורישום לצורך מציאת חריגות |
| 2 | **9.3** | משיכה ורישום לצורך מציאת חריגות |
| 1 | **9.4** | תמצית מאגר |
| 2 | **9.4** | תמצית מאגר |
| 3 | **9.4** | תמצית מאגר |
| 1 | **9.5** | שליפת נתונים מתוך מאגר חבילות למשך זמן אליו תוצג תצורת גרפים במסך הdashboard. |
| 2 | **9.5** | שליפת נתונים מתוך מאגר חבילות בשיחה אליו תוצג תצורת גרפים במסך הdashboard. |
| 3 | **9.5** | שליפת נתונים מתוך מאגר תהליכי מחשב אליו תוצג תצורת גרפים במסך הDASHBOARD. |
| 1 | **9.6** | יצירת גרף מתוך הנתונים ממאגר החבילות למשך זמן. |
| 2 | **9.6** | יצירת גרף מנתוני מאגר החבילות בשיחה. |
| 3 | **9.6** | יצירת גרף מנתוני מאגר תהליכי המחשב. |
| 1 | **9.7** | רישום התקפות סייבר |
| 2 | **9.7** | רישום התקפות סייבר |

**טבלת ישויות חיצוניות בטרנזקציות**

|  |  |  |
| --- | --- | --- |
| מס' ישות | מס' טרנזקציה | תפקיד הישות בטרנזקציה |
| 2 | **9.1** | תפקיד הסוכן בעמדת הקצה הוא לקלוט חבילות הנכנסות ויוצאות אל המחשב. |
| 1 | **9.2** | תפקיד המשתמש הוא להיכנס לממשק המערכת, המערכת תעלה בממשק הdashboard. |
| 1 | **9.5** | לאחר יצירת הגרפים ותצוגתם המלאה יוצגו המסכים למשתמש המערכת. |
| 1 | **9.6** | תפקיד המשתמש הוא לעצור / להחזיר את הזרמת המידע למסך הdashboard. |
| 1 | **9.7** | המשתמש בוחר התקפה על הכתובת יעד שהוגדרה. |

# עיצוב מנשקי עץ התפריטים ומסכי קלט פלט

## סימולטור התקפה

1.SYN Flood

2.Anomaly File Transfer

3.Port Scan

4.Chat client to server

Chat

4.1 הזנת פורט

4.2 הזנת IP

4.3 העברת מחרוזת בגודל של פי 10 מהנורמל

4.4 העברת 100 רצפי מחרוזות בגודל של מילה יחידה

4.5 שליחת רצף מחרוזות יחיד בגודל אות בודדת

Port Scan

3.1 הזנת IP

3.2 הזנת פורט

3.2 שליחת חבילות SYN וגם FIN

3.3 הצגה של פורטים פתוחים\סגורים

SYN Flood

* 1. הזנת IP יעד
  2. הזנת פורט
  3. שליחת חבילות SYN

2.1 העברת קובץ יחיד גדול

2.2 העברת 100 קבצים קטנים בגודל 10 ביט

## ממשק WEB - UI

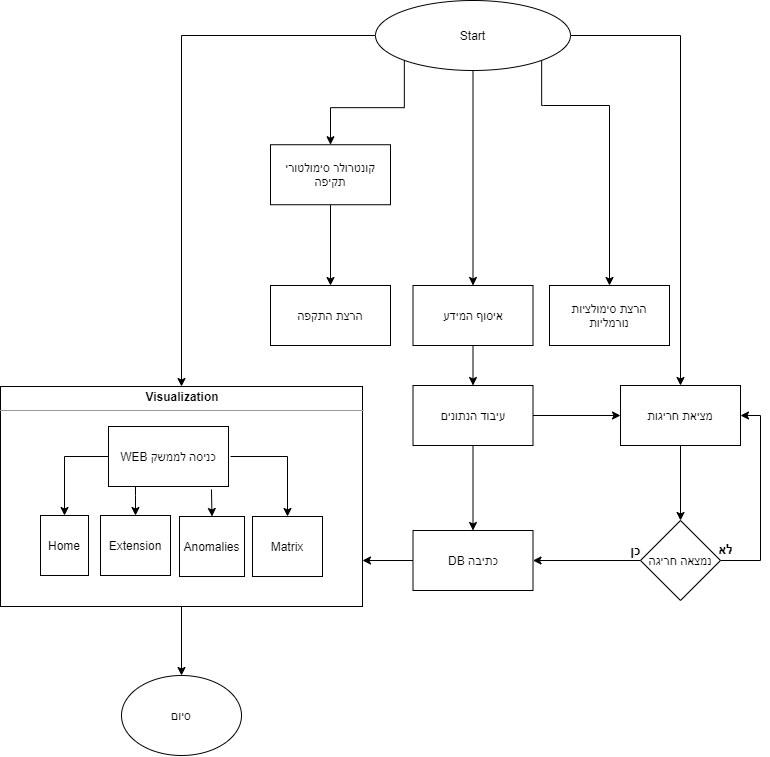
# בסיס הנתונים



# ארכיטקטורת המערכת והמודלים העיקריים

שפת התכנות שנבחרה לשמש בפרויקט זה כצד לקוח (client side) הינו Python בגרסה 3.8 עקב יציבותה הרבה ויתרונותיה הרבות על פני מתחרותיה. בנוסף היא שפת תכנות דינמית מהנפוצות ביותר. פייתון תוכננה תוך שימת דגש על קריאות הקוד, וכוללת מבנים המיועדים לאפשר ביטוי של תוכניות מורכבות בדרך קצרה וברורה. אחד המאפיינים הבולטים בתחביר השפה הוא השימוש בהזחה להגדרת בלוקים של קוד (ללא שימוש בסוגריים או במילים שמורות לצורך כך, כמו ברוב השפות הנפוצות). פייתון היא שפה מרובת־פרדיגמות, המאפשרת תכנות מונחה-עצמים, תכנות פרוצדורלי, ובמידה מסוימת גם תכנות פונקציונלי.   
לפיתוח צד שרת נגיע באמצעות שימוש בשפת Python וארכיטקטורות MVC וASP.NET בשפת C#.  
ASP.NET הינה טכנולוגית קוד פתוח ליישומי רשת לצד שרת (server side)  
והינו פיתוח של Microsoft והינה שדרוג תפסתי חדש לASP המתיישן והיווה מענה לטכנולוגיית JSP מחברת SUN ASP.NET מספק כלים לפיתוח אתרים דינמיים. מאגרי המידע ישמרו בMSSQL מבית מיקרוסופט אשר משתמשת בשפת הפיתוח שבאמצעותה מועברות הפקודות למערכת (על ידי כתיבת פקודות או שימוש בממשק גרפי ששולח פקודות באופן סמוי) היא TRANSACT-SQL ‏(TSQL) שהיא מימוש תקן ANSI של שפת SQL משמשת לתשאול וטיפול בנתונים יצירת טבלאות והיחסים ביניהן ותחזוקת המערכת תוך שימוש בתוכניות שירות שונות.

# תיאור לוגי של המערכת



# Class Diagram

## האזנה

TrafficCapture – תפקידה להפעיל את מודל האזנה באמצעות שאר המחלקות.

adapterInfo – תפקידה לתת את הכתובת IP לפי התקן תקשורת פעיל.

Scapy – תפקידה לייצר תעבורה ברשת.

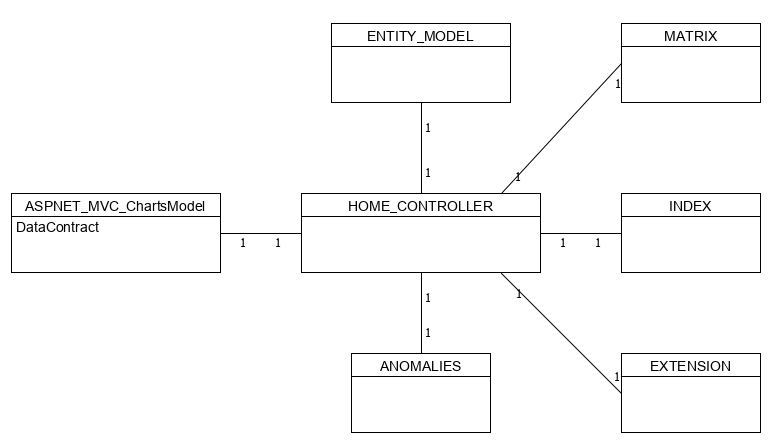
Parsing RawData – תפקידה לפרסר את המידע הגולמי.

PacketCounting – תפקידה למנות את כמות החבילות ב10 שניות.

dbWrite – תפקידה לכתוב לתוך מאגרי הנתונים ב-DB.

Flow – תפקידה למנות ולשמור את החבילות לפי פרוטוקול תקשורת.

## תצוגת Dashboard



Home Controller – תפקידו לנהל את כל ממשק ה – WEB.

Anomalies – תפקידו לזהות את האנומליות על פי הנתונים ב-DB.

ASPNET\_MVC\_ChartsModel – תפקידה למדל את כל מבני הנתונים במערכת.

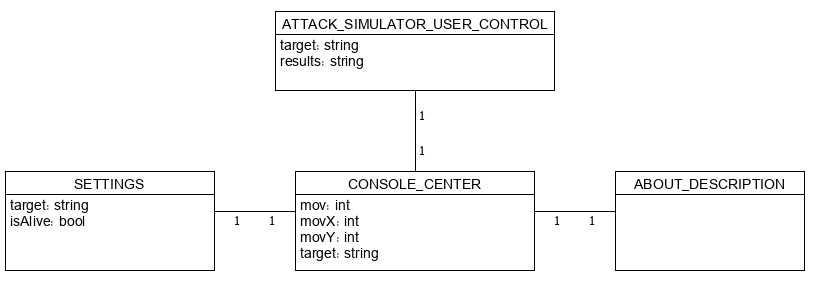
ENTITY\_MODEL – תפקידה למדל את היישויות במערכת.

MATRIX – מחלקה לחישובי שגיאות ודיוק של המודל לזיהוי חריגות.

INDEX – הדף הראשי של המערכת.

EXTENSION – הרחבת מידע לדף הראשי.

## סימולטור התקפה



CONSOLE\_CENTER – ממשק ראשי לשליטה על סימולטור ההתקפה.

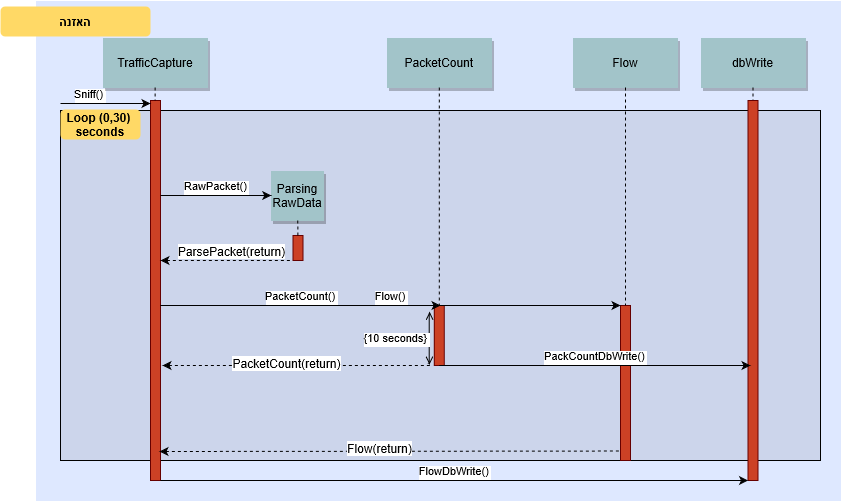
SETTINGS – תפקידה להגדיר את כתובת היעד להתקפה.

ATTACK\_SIMULATOR\_USER\_CONTROL – מחלקה לשימוש בהתקפות תקשורת.

ABOUT\_DESCRIPTION – ממשק הסבר על הConsole.

# Sequence Diagram

## האזנה



**Flow**

כל שיחה חדשה נרשמת לטבלה (ללא קשר לזמן).

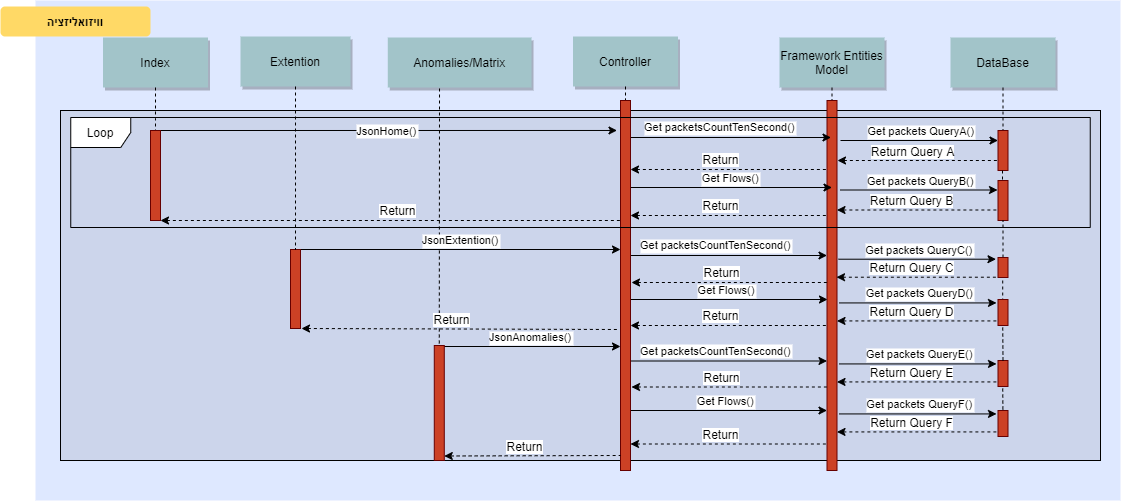
כל עוד לא מצאנו flow חדש, הם ישמרו בזיכרון של המחשב ולאחר 30 שניות של הרצת התוכנית כותבים את ה – flow שהתקבלו אל מאגר הFlows ב-Database.

**PacketCount**

רשומה חדשה לטבלה כל 10 שניות, כלומר הוא כותב לטבלה בזמן ההרצה.

המשמעות של הטבלה היא כמות חבילות לפרק זמן של 10 שניות כלומר שכל 10 שניות רושמים רשומה לטבלה שמונה את כמויות החבילות ,ה- flags השונים וגודל כלל החבילות ב10 שניות.

## ממשק WEB - UI



Index – הדף הראשי של האתר אשר מתעדכן בצורה דינאמית בכל 10 שניות על ידי הController.

Extension – דף הרחבה לדף הראשי והינו סטאטי.

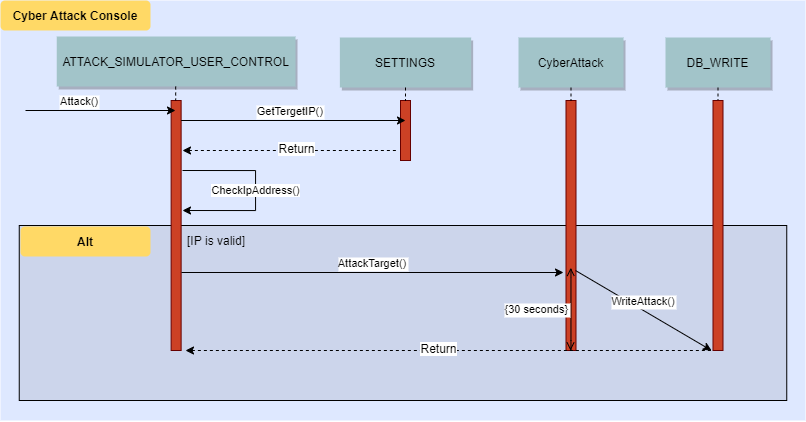
Anomalies/Matrix – דף המציג את האנומליות ברשת ואת השגיאות והדיוקים של המערכת.

Controller – מנוע שמושך נתונים באופן קבוע מתוך שכבת ה- Framework Entities Model בעת כניסה לדף במערכת.

Framework Entities Model – שכבת קישור בין המערכת לDB.

DataBase – משמש לשליפת מידע במערכת.

## סימולטור התקפה



ATTACK\_SIMULATOR\_USER\_CONTROL – ממשק הניהול לסימולטור ההתקפה. המחלקה מקבלת כתובת IP, בודקת שהכתובת נכונה ויש תקשורת אליה, ולאחר מכן מבצעת מתקפה על פי הבחירה של המשתמש.

SETTINGS – מחלקה להזנת כתובת היעד אותה המשתמש ירצה לתקוף.

CYBER\_ATTACK – מחקלה המכילה את ההתקפות השונות. מבצעת את ההתקפה ולאחר 30 שניות רושמת את ההתקפה לDATABASE.

DB\_WRITE – מחקלה האחראית על כתיבת רשומות למאגרי המידע לשדה "Simulator Attack" לפי הזמנים בהם התחילה ההתקפה.

# חלוקת תפקידים

**הקמת מעבדה**

אלמוג, קובי וכנרת בעזרת פלד מנהל מערכות מידע במכללה.

**בחירת Features**

1. בחירת Features תוך שימוש במחקרים קיימים: קובי אלמוג וד"ר צחי.

**יצירת מודל**

1. יצירת מודל המבוסס על בחירת ה Features: אלמוג וד"ר צחי.

**צד לקוח**

* פיתוח סוכן מנהל את התהליך כולו- קובי
* פיתוח הסנפת/האזנה פקטות-אלמוג
* פיתוח כתיבה לDB – כנרת

בונוס- פיתוח הריגת תהליך וחסימה של תעבורה ספציפית.

**צד שרת**

* אפיון DB בטבלאות- כנרת ואלמוג
* יצירת פיתוח תקשורת מול סוכנים-קובי
* אינטראקציית אדם מחשב- אלמוג, קובי וכנרת.

# לוחות זמנים

הספקת גרסאת אלפא למערכת

על הסטודנטים לעמוד בלוח זמנים צפוף אשר תהא תואמת את לוחות הזמנים הסימסטריאלים והלו"ז שנקבע על ידי מנחה הפרוייקט והינם כ-שלושה חודשים מתחילת העבודה   
וכן לעמוד בלוז הפורט להל"ן

## 

|  |  |
| --- | --- |
| שלב | זמן |
| מסמך דרישות מפורט | 9 שבעות |
| מסמך עיצוב תכנון | שבוע 12-13 |
| מסמך מפורט | תחילת סמסטר ב |
| PILOT | שבוע רביעי סמסטר שני |
| מוצר סופי | סוף שנת הלימודים |

# ביבליוגרפיה

1. Dey, . (May 24, 2019). **Evaluating Classification Models**. נדלה ב 26 במאי 2020 from https://towardsdatascience.com/hackcvilleds-4636c6c1ba53.
2. G. Mehrotra , K. K. Mohan, C & Huang, H. (2017). **Aomaly Detection Principles and Algorithems**. Syracuse, NY, USA: Springer.
3. Lahoda, . (09/17/2019). **Getting Started with EF Core**. נדלה ב 26 במאי 2020 from https://docs.microsoft.com/en-us/ef/core/get-started/?tabs=netcore-cli.
4. Scapy, W. (May 25, 2020). **Welcome to Scapy’s documentation!**. נדלה ב 26 במאי 2020 from https://scapy.readthedocs.io/en/latest/.
5. Wikipedia, W. ( 8 April 2020). **Unsupervised learning**. נדלה ב 23 במאי 2020 from https://en.wikipedia.org/wiki/Unsupervised\_learning.
6. Zhou , Z. Liu, F & Ting , . (2008). **Isolation Forest**. National Key Laboratory for Novel Software Technology, Nanjing 210093, 1.
7. סיני, י. (2015 מרץ). **מבוא לשימוש ביכולות Learning Machine בפתרונות אבטחת מידע וסייבר**. נדלה ב 26 במאי 2020 from https://www.digitalwhisper.co.il/files/Zines/0x3B/DW59-1-ML-Security.pdf.