

הצעת פרויקט – עבודה גמר י"ד הנדסת תוכנה

פרטי מגיש ההצעה

מספר מודול: 571281

שם מכללה: מכללת אורט סינגלובסקי

שם סטודנט: קוין מנשרוב

ת.ז: 326969805

שם פרויקט: מערכת התראאה לדייהי אונומליות ברשת.

מנחות פרויקט: נילי נווה, אלי גוריאל, אפרת ינברג, אסף אמיר.

תוכן עניינים

4	תיאור הנושא
5	רקע תיאורי בתחום הפרויקט
5	זיהוי אномליות ((Anomaly Detection))
5	למידת מכונה בלתי מופקחת ((Unsupervised Machine Learning))
6	מכונת בולצמן מוגבלת ((RBM - Restricted Boltzmann Machines))
7	תיאור הפרויקט
8	פירוט שלושת המודלים
9	הגדרת הבעיה האלגוריתמית
9	קלט ופלט המערכת
10	תהליך הפתרון
10	1. שלב הקלט - התמודדות עם ריבוי נתונים
10	2. שלב העיבוד - ניתוח ושחזור מידע
11	3. שלב החלטה (מדידת פער)
12	פירוט דата סטים
12	דатаה סט CIC-IDS-2017 לזיהוי פעילות חשודה בראשת
13	דטהה סט CIC-IDS-2017 לזיהוי ניסיונות חדירה
14	דטהה סט CERT Insider Threat (למידה של פרופיל משתמש) לזיהוי הפרת מדיניות ארגונית
15	הליכים העיקריים בפתרון בעיה בטכנולוגיות הנדסה מתקדמות
15	איסוף וניתוח נתונים
15	עבור מודל זיהוי חדירות
15	עבור מודל זיהוי פעילות חשודה
15	עבור מודל לזיהוי הפרת מדיניות
16	נקיי ונורמל נתונים
16	הנדסת מאפיינים וחולנות זמן
17	תרשים תהליכי מערכת
18	הליכים העיקריים בתחום למידת מכונה
18	1. עיבוד מקדים והכנת הנתונים
18	2. תהליך האימון
19	3. מנגנון הזיהוי לאחר שהמודל אומן
19	4. קביעת סף ההחלטה ((Threshold))

20	הליכים עיקריים בתחום רשתות מחשבים/תקשורת נתונים/אבטחת מידע
20	1. ניטור והזנה לתעבורה
20	2. חילוץ ופירסור
20	3. ניהול התראות
21	תיאור פרוטוקולי תקשורת
21	פיתוחים עתידיים
22	תיאור טכנולוגיה הנדסה
22	מנוע זיהוי וניתוח (שפת פיתוח Python)
24	שרת ניהול Backend (שפת פיתוח Java)
25	מוד הנתונים ((MongoDB - Database))
26	צד לקוח וממשק המשתמש (שפת פיתוח React)
27	ארQUITקטורת המערכת והתקשרות
29	פרטים פורמליים
29	לוחות זמינים
30	חתימת הסטודנט
30	חתימת רcz המגמה

תיאור הנושא

התלות הגוברת של ארגונים במערכות ממוחשבות וברשות תקשורת חשפה אותם בפני מגוון רחב של איומים. אחד האתגרים המרכזיים הוא לזהות פעילות חריגה או כוונת תקיפה בתוך אוסף רחב של נתונים, לפני שנגרם נזק.

נושא הפרויקט מתמקד בשימוש תחום ניתוח הנתונים ולמידת מכונה כמענה לאתגרי אבטחה אלו. כלים אלו מאפשרים מערכת הגנה שמסוגלת למודד התנהבות רגילה של הרשת, משתמשים ומערכות בארגון.

השימוש בנייחות נתונים ולמידת מכונה הוא לצורך זיהוי אונומליות (סטיות מהנורמה שנלמדה) בזמן אמת. למידת מכונה עוזרת לעבד כמות גדולה של נתונים (תעבורה ראשית, פעילות משתמשים) ולמצוא דפוסים חשודים שקל לפופו.

היכולת לזהות אונומליות בעזרת למידת מכונה מתחילה לכמה איומים. זיהוי ניסיונות חדירה - כגון סריקת פורטיפם, תנעה רוחבית חשודה ברשת שמטרת להגיע למערכות ארגון קרייטיות, או למידע רגיש.

איתור תקשורת זדונית - גילוי תקשורת חריגה בראשת המUIDה על בקרת תוקף חיצונית או הוצאת נתונים רגיסטים, הננתינה בתוך התעבורה בראשת. אכיפת מדיניות ארגונית - זיהוי פעילות משתמשים או מערכות המפירה את מדיניות האבטחה הארגונית שנקבעה.

רקע תיאורטי בתחום הפרויקט

דיהוי אונומליות (Anomaly Detection)

anomalיה היא דפוס נתונים שאינו תואם את התנהוגות הצפואה ומוגדרת מראש. בהקשר לאבטחת מידע וסיביר דיהוי אונומליות הוא קריטי מכיוון שרוב מתקפות הסיביר מתחילה בפעולות חריגות שאין מזוהות על ידי מערכות הגנה מסורתיות המבוססות על חתימות ידועות.

מערכות דיהוי אונומליות מבוססות על הנחה שפעולות התקפה תהיה שונה לגמרי פעולה לגיטימית רגילה. יתרון המרכזי הוא יכולת לזהות איום חדש ולא מוכרם, שכן אין משתמשות על ידע מוקדם על התקפה אלה על הכרת ה"נורמלי".

למידת מכונה בלתי מפוקחת (Unsupervised Machine Learning)

למידת מכונה היא תחום בביונה מלאכותית המאפשר למערכות למדוד נתונים ולשפר את ביצועיהן ללא תכננות מפורש. בפרויקט זה הגיעה היא למידת בלתי מפוקחת, שבה האלגוריתם מקבל נתונים גולמיים ללא תוצאות (ללא ידע מוקדם מהי התקפה ומה לא).

המטרה בלמידה בלתי מפוקחת היא לגלוות את המבנה הפנימי הנסתור של הנתונים. במקרה של דיהוי אונומליות, המודל לומד את המאפיינים של הנתונים ה"נורמליים" המהווים את הרוב המוחלט של המידע, ובכך מסוגלת לזהות כל נתון חדש שחוורג מבנה זה כח凶手.

מכונת בולצמן מוגבלת (RBM - Restricted Boltzmann Machines)

RBM היא סוג של רשת עצבית מלאכותית המשמשת למינימיזציה למשפחת המודלים הגנרטיביים. היא מורכבת משתי שכבות - שכבה נראית (Visible Layer) המכבלת את הקלט, וscrição נסתרת (Hidden Layer) הלומדת לייצג את המאפיינים הנסתורים של הנתונים. ההגבלה (Restriction) במודל זה היא שאין קשרים בכל שכבה, אלה רק קשרים דו ציווניים בין שתי השכבות. מבנה מוגבל זה מאפשר תהליכי אימון יעילים ומהיר יותר (ביחסו למכונת בולצמן מלאה). מטרתו הכללית של המודל היא ללמידה את התפלגות ההסתברות של הנתונים והיצוגים הנסתורים שלהם, והוא משתמש לעיתים קרובות למשימות של מיידית מאפיינים (Feature Learning), הפקחת מדדים ומערכות המלצה.

תיאור הפרויקט

הפרויקט מתמקד בפיתוח ארכיטקטורה לחיה אונומליות או יומיים ברשות מbossot למידת מכונה (נשתמש במכונית בולצמן) שתתריע בזמן אמיתי על פעילות חריגה ברשות.

בניגוד למודלי למידת מכונה אחרים (רגרסיבי וסיווג) שהזדים תוצאה ישירות, מכונת בולצמן לומדת את הקשרים ואת המבנה הסטטיסטי בתוך הנתונים עצם (אייה תכונות יש בתוך הנתונים ואיך הם קשורים אחד לשני).

בהקשר לפרויקט, תפקיד המכונת הוא לא לסואג התקפות ידועות, אלה למוד את התפלגותם ואת המבנה של נתונים "נורמליים" ברשות הארגונית. המערכת תאומן על תעבורת רשות כדי לבנות קו בסיס של התנהגות תקינה.

או יומיים או אונומליות יזהו כאשר יגיע קלט חדש (כגון חבילת מידע) ואחד המודלים יתנו כפלט ציון גובה או שגיאת שחזור גובהה, מה שמסמן על כך שהנתון אינו תואם הדפוסים הנורמליים שמערכת למדה.

פירוט שלושת המודלים

המערכת تعمل על גבי שלושה מודלים מקבילים, ליבת הפרויקט היא פיתוח שלושה מודלים נפרדים של מכונות בולצמן, הפעילים בו זמנית.

המודל הראשון מזהה פעילות חשודה בראשת

המודל יאומן על מאפיינים כלליים של תעבורת רשת (כגון נפחים, פרוטוקולים, תדיות ומשר התקשרויות). בכר המודל יצליח לזהות anomalיות סטטיסטיות רחבות בתעborת הרשת כגון זיהוי דפוסים המעידים על תקשורת זדונית או סמייה (בקרה תוקף חיצונית - התקשרויות קטנות וקבועות לעדים לא מוכרים, הוצאה נתונים - העברת נתונים חריגה בנפח ועוד).

המודל השני מזהה ניסיונות חדרה

המודל יאומן על נתונים יותר מדויקים, כגון יומני אימות, נתוני זרימה בין רכיבים פנימיים לצורך זיהוי התקפיים המכוניים לנכס הארגון. מודל זה יזהה ניסיונות חדרה אקטיביים, לדוגמא - סריקות פורטימ (ריבוי חיבורים כושלים מייד בודד), התקפות כוח גס (ריבוי ניסיונות אימות כושלים), תנוצה רוחבית חשודה ועוד.

המודל השלישי מזהה הפרות מדיניות ארגון

המודל יאומן על דפוסי גישה לגיטימיים של משתמשים וקבוצות למשאים רגיסטים לצורך אכיפת כללי אבטחה. מודל זה יזהה פעולות המפרות את מדיניות הארגון לדוגמא - זיהוי מצב בארגון שבו חשבון ממחלקה מסוימת מנסה ל搶ת למשאב או מידע שלא ניתן אליו עד כה. פועלה זו תזוהה כ anomalיה ביחס לפרופיל התנהגות הנורטטיבי של אותה קבוצה משתמשים.

הגדרת הבעיה האלגוריתמית

הבעיה המרכזית שהמערכת צריכה לפתור היא: איך לזהות משהו "לא בסדר" \ "לא נורמלי" בראשת, מבלי שיוודעים מראש נראית פעילות חריגה \ סטייה מהנורמה (המודל מסתמך על הנתונים שניתנו לו שזה הנורמה \ פעילות לא חריגה).

במקום לחפש איזומים ספציפיים, המודלים פועלים הפוך, למידת הנורמה:
 כל שלושת המודלים לומדים בrama גובה ודיק גובה איך נראית שagara תקינה, כל מה שלא תואם למודלים האלה (בין אם זה ניסיון פריצה, תקשורת חריגה או הפרת נהלים) יסמן כחריג.

קלט ופלט המערכת

<u>פירוט</u>	<u>הגדרה</u>
מאפיינים רבים ממדים המציגים את תמונה המצב של הרשת בכל רגע נתון. הוקטורים אינם מסומנים (למידה לא מפוקחת)	קלט המערכת
ציוו שגיאה שחזרו: ערך מסופרי המציין את מידת החריגה. מתוך זה ניתן לבצע החלטה בינארית (נורמלי / לא נורמלי)	פלט המערכת

תהליך הפתרון

1. שלב הקלט - התמודדות עם ריבוי נתוניים

המערכת מקבלת בכל רגע נתון "תמונה מצב" של הרשת. האתגר הוא ש"תמונה המצב" הזה מורכבת מהמוני פרטיים קיטינים: כתובת השולח, גודל הקוביץ, שעה, סוג פרוטוקול ועוד. המערכת צריכה לקחת את כל אוסף הפרטיים זהה ולהפוך אותו לשורה אחת של נתונים שהמחשב יוכל לעבד.

2. שלב העבודה - ניתוח ו恢復 מידע

זהו שלב במערכת המבוצע על ידי מודלי למידת מכונה (מכונות בולצמן - RBM).

כדי להבין אם הנתוניים תקינים, המודל מבצע תהליך דו שלבי -

1. המודל לוקח את הנתוניים ומנסה לתרצת אותם (למידת התבנית של הנתוניים).
2. לאחר מכן, המודל מנסה לשחזר את המידע המקורי מתוך אותו תמצות, בהתבסס על ה"שגרה" התקינה שהוא מאומן עליו.

אם הנתוניים תקינים (מוכרים למערכת) - המודל יצליח לשחזר את הנתוניים המקוריים בהצלחה, כי המודל "מכיר" את הדפוסים של הנתוניים האלה.

אם הנתוניים חריגים (פעילות לא מוכרת) - המודל יתקשה לשחזר את הנתוניים המקוריים, וה结論ה לא תהיה מדויקת.

3. שלב ההחלטה (מדידת פער)

בשלב הסופי, המערכת צריכה לקבל החלטה של כן או לא (האם להפעיל התרעעה?). המערכת עשויה זאת על ידי חישוב של ההבדל של מה שנכנו לבין מה שהמודל הצילח לשחזר. הבדל זה נקרא "שגיאת השחזר".

שגיאה נמוכה: המערכת הצילה לשחזר את המידע -> הפעולות תקין.

שגיאה גבוהה: המערכת נכשלت בשחזר (הפער גדול מדי) -> הפעולות חשודה ומוגדרות אנומליות.
המערכת משווה את גודל השגיאה ל"קו האדום" (סף) שנקבע מראש. אם השגיאה עוברת את הקו האדום נשלחת התראה למנהל המערכת.

פירוט>Data סט

דата ט 2017 CIC-IDS לזייה פעילות חשודה בראשת

משתנים בלתי תלויים - מאפיינים סטטיסטיים של זרימה כגון: סך בתים / חבילות שנשלחו, יחס פרוטוקולים (TCP / UDP), מושך חיבור, קצב העברת נתונים.

משתנה תלוי - ציון שגיאת שחזור (Reconstruction Error Score).

- דוגמא

	A	B	C	D	E	F
1	Destination	Flow Durat	Total Fwd F	Total Back	Total Lengt	Total Lengt
2	3268	1.13E+08	32	16	6448	1152
3	389	1.13E+08	32	16	6448	5056
4	0	1.14E+08	545	0	0	0
5	5355	100126	22	0	616	0
6	0	54760	4	0	0	0
7	88	617	7	4	484	414
8	1031	8	1	1	6	6
9	88	881	9	4	656	3064
10	88	1056	9	6	3134	3048

דата ט 2017-CIC-IDS-לזיהוי ניסיונות חדירה

משתנים בלתי תלויים - מאפייני ניסיונות חדירה כגון: מספר ניסיונות אימות כושלים (Brute Force), מספר ניסיונות שנסרקו על ידי IP בודד (Port Scanning).

משתנה תלוי - ההחלטה אם הפעולות היא Normal / Intrusion.

דוגמא -

	AR	AS	AT	AU	AV
1	FIN Flag C	SYN Flag C	RST Flag C	PSH Flag C	ACK Flag C
2	0	1	0	0	1
3	0	1	0	0	1
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	1	0
8	0	0	0	0	1
9	0	0	0	1	0
10	0	0	0	1	0

דатаה טו CERT Insider Threat (למייה של פרופילי משתמש) לזייה הפרת מדיניות ארגונית

משתנים בלתי תלויים - מאפייני התנהגות משתמשים כגון: שעות התחברות חrigerות, גישה לקבצים רגילים, שימוש בהתקנים חיצוניים, כמות מיילים שנשלחו מוחוץ לרשות.

משתנה תלוי - ההחלטה אם היה הפרה של המדיניות או לא היה הפרה של המדיניות.

- דוגמא -

	A	B	C	D	E
1	id	date	user	pc	activity
2	{S7A7-Y8Q####}	DTAA/RES0	PC-3736	Connect	
3	{G7A8-G1O####}	DTAA/BJC0	PC-2588	Connect	
4	{R3L8-N0LV####}	DTAA/EMZ0	PC-1479	Connect	
5	{I2F1-B5FB####}	DTAA/ZKH0	PC-1021	Connect	
6	{P7R6-C5T####}	DTAA/RES0	PC-3736	Disconnect	
7	{K5Q6-F1A####}	DTAA/CVW0	PC-0282	Connect	
8	{M0F6-O2F####}	DTAA/RQH0	PC-4225	Connect	
9	{F5H8-07Q####}	DTAA/AQG0	PC-1127	Connect	
10	{C2M6-A5G####}	DTAA/OJH0	PC-1730	Connect	

:קישור לדטה טו CIC-IDS-2017

<https://www.unb.ca/cic/datasets/ids-2017.html>

:קישור לדטה טו CERT Insider Threat

https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247?file=24855644

הלייצים עיקריים בפתרון בעיה בטכנולוגיות הנדסה מתקדמות

איסוף וניתוח נתונים

מקורות הנתונים והאיסוף בשלב הראשון, علينا להשיג מידע נתונים (dataset) המציג תובורת רשת אמיתי. מכיוון שהמערכת מבוססת על שלושה מודלים שונים, הנתונים צריכים להכיל מספר רבדים של מידע.

עבור מודל דיהוי חדירות

נאסוף יומני אימות (Authentication Logs) משרתים ונקודות קצה (כגון Active Directory Logs או SSH Logs) הcoliils מידע על ניסיונות כניסה מוצלחים וכושלים.

עבור מודל דיהוי פעילות חשודה

נאסוף יומני זרימה (Network Flows) ויומי פרוטוקולים. המספקים מטא-דטה על התעבורה (מי דבר עם מי, מתי, כמה מידע הועבר).

עבור מודל לזיהוי הפרת מדיניות

נשתמש בנתוני גישה למשאים וקבצים (File Access Logs).

המטרה היא להשתמש במאגר נתונים קיים ומוכר או בנתונים מסימולציה מבוקרת, אשר מכילים בעיקר תעבורת תקינה לצורך אימון המודל, אך גם דוגמאות מתוויות של התקפות לצורך בדיקה של המערכת בשלב מאוחר יותר.

ניקוי ונרמול נתונים

נתוני רשות מגיעים לרוב סטוקסט לא מובנה, עם ערכים חסרים או "רעש" מיותר. בשלב זה נבצע ניקוי של רשומות פגומות והסרה של מידע לא רלוונטי ללמידה.

לאחר הניקוי נבצע תהליך של נרמול, מודלים מסווג RBM הם מודלים מבוססי "אנרגיה" משתמשים בפונקציות אקטיבציה יחסית וגישה. אם המודל קיבל שדה אחד עם ערכים קטנים ושדה אחד עם ערכים עצומים המודל יתקשה להתכנס.

לכן יש צורך לנរמל את כל הערכים המספריים לטווח אחד כדי שהמשקלות ברשות ילמדו בצורה מאוזנת.

הנדסת מאפיינים וחלונות זמן

שלב זה מתרץ בהתאם הנתונים למודל ה-RBM. הפרויקט זה לא נzin את המודלים עם חבילות מידע (packets) בודדות זו אחר זו, אלה נבצע תהליך של איחוד נתונים בחלונות זמן.

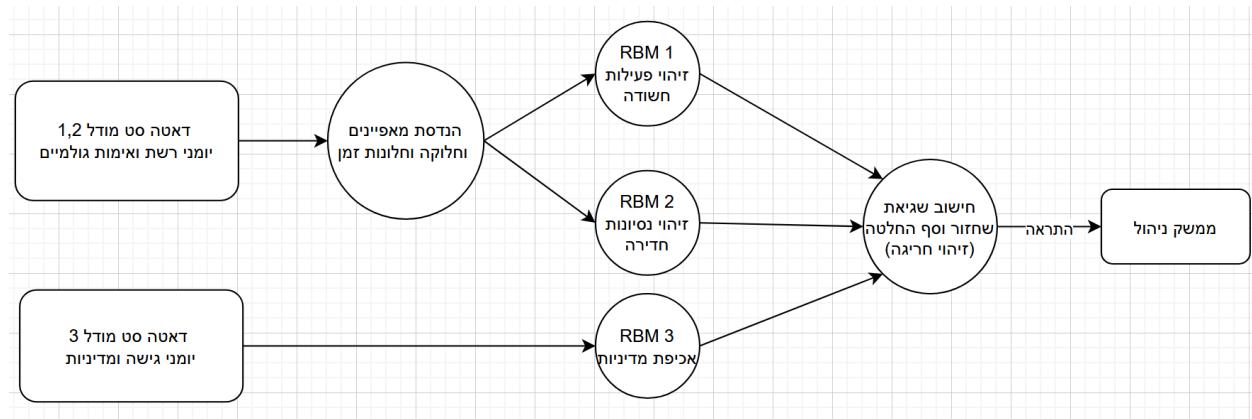
הסיבה לשלב זה נובעת משתי סיבות, אבטחה וביצועים:

- 1. הקשר התנהוגותי (Context)** - התקפות רבות, כגון סריקת פרוטוקום או מניעת שירות (DDOS), מורכבות מאלפי חבילות שכל אחת מהן נראהיה תקינה בפני עצמה. האנומליה מתגלה כאשר בוחנים את הקצב והכਮות לאורך זמן, ניתוח חבילה בודדת תחסיר את הקשר בין החבילות لكن הוא לא שימושי.
- 2. יעילות חישובית** - ברשות מודרניות עוברות מיליאני חבילות בשניה ברשות. ניסיון לעבד כל חבילה בנפרד ייצור עומס חישובי ועיכוביים.

המערכת תחלק את התעבורה לחלונות זמן קבועים (לדוגמא, כל חמיש שניות). עברו כל חלון זמן ניצור וקטור מאפיינים המסכם את הפעולות באותו רגע - כמה נסיוונות אימות נכשלו? מה היה היחס בין חבילות שנשלחו לאלו שהתקבלו?

תהליך זה הופך את זרם הנתונים הגלומי לטבלה מסודרת של "מצבי רשות", המאפשרת למודל ה-RBM ללמוד את ההתפלגות הסטטיסטית של הרשות ולזהות חריגות.

תרשים תהליכי מערכת



הלייצים עיקריים בתחום למידת מכונה

1. עיבוד מקדים והכנות הנתונים

לפני שהנתונים נכנסים למודל, עליהם לעבור התאמה מכיוון שמכונת בולצמן רגישות לטוווחים שונים של מספרים.

נורמל - המירה של כל הערכים המספריים (כגון גודל חבילה, משך שייחה) לטווח באמצעות שיטת Min-Max Scaling של המודל.

קידוד קטגוריאלי - המירה של נתונים טקסטואליים (כגון סוג פרוטוקול) לייצוג מספרי, כך שהרשות תוכל לעבד אותם.

חלוקת DATA - הפרדת הנתונים לשיט אימון המכיל תעבורת תקינה בלבד, וווט בדיקה המכיל ערבות של תעבורת תקינה והתקפות, לצורך אימות הביצועים.

2. תהליכי האימון

האימון מתבצע בשיטה למידה לא מפוקחת על שלושת המודלים במקביל. המודל לומד את התפלגות ההסתברות של הנתונים התקינים. המודל מנסה למצוא את מערכת המשקولات שיגרום לנ נתונים התקינים להיות בעלי אנרגיה מינימלית.

נستخدم באлогרitem Contrastive Divergence. האלוגרitem מבצע קירוב של הגראדיאנט על ידי דגימה Gibbs Sampling) הוא מעביר את הנתונים מהשכבה הנראית לנסתרת ובחזרה, מעדכן את המשקولات כדי לצמצם את ההפרש בין היקלט המקורי לשחזור שלו.

3. מנגנון הדיזהוי לאחר שהמודל אומן

שחזר - כל וקטור נתונים חדש (V) שנכנס למערכת עובר דרך הרשת. הרשת מייצרת ייצוג פנימי של וקטור הנתונים (H) ומנסה לשחזר אותו בחזרה (V').
חישוב השגיאה - המערכת מחשבת את המרחק המתמטי בין הקלט המקורי (V) לבין הפלט שימושזר (V'). לצורך חישוב השגיאה השתמש בפונקציית Mean Squared Error.

רציונל - אם הרשת זיהתה דפוסים דומים באימון (התעבורה תקינה), השחזר יהיה מדויק והשגיאה תהיה קרובת לאפס. במקרה והרשת נתקלה בדפוס חדש (חשור) היא תיכשל בשחזר והשגיאה תהיה גבוהה.

4. קביעת סף החלטה (Threshold)

הסף נקבע על סמך ביצועי המודל על סט הנתונים הנקי (ללא חビルות חדשות). לדוגמה נקבע את הסף כך שיכיל 95% אחוז מהשגיאות הנורמליות.
כל חריגה מעל הסף זה בזעם אמת תסוווג מיד כאנומליה ותפעיל התראה.

הלייצים עיקריים בתחום רשותות מחשבים/תקשורת נתונים/אבטחת מידע

1. ניטור והאזנה לתקשורת

המערכת פועלת כרכיב האזנה ברשת. כדי לקלוט את כל המידע, כרטיס הרשת (NIC) מוגדר לעבוד במצב Promiscuous Mode (מצב האזנה מלאה).

במצב רגיל, כרטיס רשת מתעלם מחבילות שלא מיועדות אליו. במצב האזנה מלאה, הכרטיס קולט ומעביר למעבד את כל החבילות שעוברות. גם אם הן מיועדות למחשבים אחרים. דבר זה מאפשר למערכת לקבל תמונה מצב מלאה של התקשורת בזמן אמת.

2. חילוץ ופירסום

לאחר קיליטת חבילות המידע הגלומות (רצף של אחדים ואפסים) מתבצע תהליך של פירסום - פירוק החביליה לשכבות לפי מודל ISO.

המערכת קוראת את הכותרות של הפרוטוקולים כדי להלץ שודות מידע קרייטיים (כגון כתובות IP מקור ויעד, פורטים, דגלים, גודל Payload).

שלב זה הופך את המידע לנוטונים מובנים שנייתן לבצע עליהם חישובים סטטיסטיים.

3. ניהול התראות

כאשר המודל מחשב שגיאת שחזור גבואה החוצה את הסוף המוגדר, המערכת מייצרת אירוע אבטחה. ההתראה נשלחת למשק ניהול ונשמרת במסד הנתונים. היא תכיל זמן אירוע, מה אירוע שזזהה, היישוות המעורבות (כתובות IP).

תיאור פרוטוקולי תקשורת

1. פרוטוקoli תעבורה (TCP / UDP) המערכת מנטחת לעומק את שכבת התעבורת (רמה 4 במודל OSI).

TCP: המערכת עוקבת אחר תהליך לחיצת היד המשולשת (Three-Way Handshake). ריבוי של חבילות עם דגל SYN ללא מענה ACK תואם מעיד לרוב על סריקת פורטים או התקפת Dos.

UDP: מכיוון שהוא פרוטוקול ללא חיבור (Connectionless), המערכת מנטחת נפחים חריגים שעשוים להיעיד על הצפת מידע או ניסיונות אימות חריגים לשירותים כמו DNS.

2. פרוטוקול בקרה (ICMP) פרוטוקול המשמש לבדיקות ואבחון (כמו Ping).
ה anomalיה: תעבורת ICMP אמורה להיות דיללה מאוד. המערכת תזהה דפוסים של חבילות ICMP גדולות או תדרות גבוהות כחсад להגנת מידע בתוך הפינג (ICMP Tunneling).

פיתוחים עתידיים

1. מערכת מנעה אקטיבית: שדרוג המערכת כך שלא רק תתריע, אלא גם תבצע חסימה. המערכת תתמכשך לחומת אש (Firewall) ותוסיף כל חסימה לכתובות-IP החשודה באופן אוטומטי ברגע שהמודלים מזהים חריגה.

2. למידה היברידית: כיום המערכת מודיעה רק אם יש חריגה או לא. בעtid, נוכל לשלב מודול מסווג, שירוץ אחריו שהחריגה זוהתה, מודול זה ינסה לתת לחריגה שם ספציפי על סמך דוגמאות עבר. המטרה היא לתת לנו החלטה מיידית מדויק יותר על החריגה.

תיאור טכנולוגיה הנדסה

פיתוח המערכת מבוסס על Stack טכנולוגי מודרני המורכב משלושה עמודי תווים: Python לטעות מודולית מידית מכונה ועיבוד התעבורה, Spring Boot לטעות צד השרת וניהול הנתונים, ו-React לבנית משאך המשמש לדשبورד.

מנוע דיזיוי וניתוח (שפת פיתוח Python)

צד זה של המערכת אחראי על האזנה לרשות, עיבוד מתמטי והרצת המודלים.

הוא יפותח בשפת Python בשל סדריית הכלים העשירה שלו בתחום הסיבר וניתוח נתונים.

1. Scapy - ספרייה מתקדמת למניפולציה של חבילות רשות, המאפשרת שליטה מלאה על שדות הכותרת (Headers) בכל שכבות ה-OSI.

Scapy משתמש בחישון של המערכת, היא מאפשרת לכרטיס הרשת (NIC) במצב Promiscuous קולטת את התעבורה בזמן אמת, וביצעת פירוק של החבילות כדי לחלץ נתונים גולמיים.

2. Pandas & Numpy - ספריית Numpy מספקת תשתיית לחישוב מתמטי מהיר על מטריצות, ספריית Pandas מאפשרת ניהול ועיבוד מידע טבלאי וסדרות עתיות.

רכיבים אלו אחראים על הפיכת המידע הגלומי לקלט שנייתן להזין למוכנה:

אגרגציה: איחוד חבילות בודדות לחלונות זמן (לדוגמא, סיכון סטטיסטי של תעבורה כל 5 שניות) נרמול: המרת הערכים בטוחה באמצעות Min Max Scaling, תהליך קריטי עבור מודלים מבוססי אנרגיה.

3. PyTorch. ספרייה למידה عمוקה (Deep Learning) המאפשרת בניית רשתות עצביות וчисוב מבואו Tensors (מבנה נתונים לייצוג מספריים בFormats שונים).
באמצעות ספרייה זו שלושת מודלי ה-RBM יומשו. הספרייה אחראית על: ניהול הארכיטקטורה של הרשת (שכבות נסתבות ונראות), ביצוע תהליך השחזור (Forward and Back Pass), חישוב שגיאת השחזור בזמן אמת ויזיהו חריגה מהוסף שנקבע.

שרת ניהול Backend (שפת פיתוח Java)

צד זה אחראי על הלוגיקה העסקית. הוא אחראי על קבלת החלטות, ניהול תקשורת הא-סינכרונית מול המנווע של המודלים בלמידה מכונה, שבירת המידע והגשתו למשתמש הקצה. השירות יפותח בשפת Java על גבי פרויקט מוירק Spring Boot המוביל לפיתוח מערכות Enterprise יציבות.

MicroServices: נבחרה תשתיית המאפשרת פיתוח מהיר של שירותים. **Spring Boot Framework** התשתיית מספקת:

שרות Web מובנה: המאפשר הריצה עצמאית של האפליקציה ללא צורך בהתקנות חיצונית מורכבות. ניהול תלויות (Dependency Injection): ארכיטקטורה המבוססת קוד נקי, מודולרי וקל לתחזוקה ולבדיקה.

אבטחה וביצועים: מנגנים מובנים לטיפול בעומסם ובטחת הקוד.

תפקיד השירות והארQUITטורה: השירות ממומש בארכיטקטורת שכבות (Controller, Service, Repository) ומבצע מספר תפקידים:

1. ממשק קליטת נתונים - חישפת REST Controllers המאזינים לבקשת POST ממונו הלמידת מכונה. שכבה זו אחראית על בדיקת תקינות הקלט כדי להבטיח שרק התראות במבנה JSON תקין נכנסות למערכת.

2. לוגיקה עסקית - עיבוד התראות גולמיות ש מגיעות מהמודלים.

- הוספה חוות דעת זמן שרת וսטטוס התחלתי ("Open").

- ניתוח ציון האנומליה (Anomaly Score), וקבעת רמת החומרה כדי לעזור למנהל הרשות להתמקד בעיקר.

3. שכבת הנתונים - שימוש בספריית Spring Data MongoDB המאפשרת מיפוי אובייקטים (ORM) לשירות למסמכים JSON. רכיב זה אחראי על שמירה יעילה ושליפה מהירה של היסטורית התראות והלוגים מממד הנתונים.

4. ממשק לדשبورד: חישפה נקודות קצה מסוג GET עבור צד הלקוח. השירות מבצע אגרגציה (ביצוע חישוב על קבוצת נתונים והחזרת ערך אחד) של הנתונים שלוח סטטיסטיות מוכנות כדי להקל על הדף להציג את טעינת הדשبورד.

מסד נתונים (Database - MongoDB)

מסד נתונים מסוג NoSQL המבוסס על מסמכים (Documents-Oriented). מסד נתונים זה נבחר בשל הגמישות שלו בשימוש נתונים בפורמט JSON, התואם בדיקת הפלט שmaguim מודולי למידת המכונה. המערכת תנהל שלושה אוסף נתונים (Collections) עיקריים:

1. אוסף התראות (Alert Collection): זהו אוסף המרכז במערכת, המרכז את כל אירועי האבטחה שזוהו. כל רשומה באוסף תכיל -

- timestamp: חותמת זמן מדויקת של אירוע.
- source_ip / dest_ip: כתובות ה-IP המעורבות.
- detected_by: שם המודול שזיהה את החיראה.
- anomaly_score: הציון המספרי של שגיאת השחזר (אפשר סינון לפי חומרת השגיאה, לצורך تعدוף הטיפול בהתראות).
- status: מצב טיפול באירוע (פתוח/טיפול/סגור) לשימוש מנהל המערכת.

2. אוסף משתמשים והרשאות (User Collection): משמש את מערכת ניהול לצורך בקרת גישה לדשبورד. כל רשומה באוסף תכיל -

- username: שם משתמש של מנהל הרשות.
- password: סיסמה של המשתמש (מוספנת).
- role: רמת הרשות (למשל - צופה בלבד או מנהל מלא).
- last_login: תיעוד כניסה אחרונה למערכת.
- Created: מתי המשתמש נוצר.
- Modified: מתי שינה בפעם האחרונה.

3. אוסף מטריקות וסטטיסטיקה (Metric Collection): אוסף זה שומר נתונים רציפים לצורך הצגת גרפים בדשبورד, גם אם לא נוצרה התראה. נשמר דוגמאות זמן (Time-Series Data) של:

- ממוצע שגיאת שחזור בדקה الأخيرة.
- כמות תעבורת כוללת. נתונים אלו מאפשרים לשרת ה-JAVA לשלוף היסטוריית ביצועים ולהציג למשתמש את בריאות הרשת לאחר זמן.

צד ללקוח וממשק המשתמש (React פיתוח (React.js

השכבה הוויזואלית המאפשרת למנהל הרשות לנטר את המצב.

React.js: ספרייה Javascript פופולרית לפיתוח ממשק משתמש דינמיים וレスpondיביים.

היא תשמש לבניית דשبورד ניהול:

המשק יציג תרשימים דינמיים של שגיאת השחזר על ציר זמן. הגרף מאפשר לראות מתי שגיאת המודול

עולה מעלה הסוף המוטר, דבר המעיד על אי-יעור אבטחה פעיל.

הציגת יומן אי-יעורים (Event Log) אינטראקטיבי. המשק כולל טבלה חכמה המאפשרת למנהל הרשות

לסנן, למין ולחפש התראות לפי פרמטרים קritisטים (כגון: כתובת IP תוקפת, רמת חומרה, סוג המודול

שזיהה), ובכך מקלה על תחקור אי-יעורי אבטחה.

שכבת Frontend אחראית על ביצוע קריאות API REST א-סינכרוניות מול שרת h-Boot.

היא מושכת את הנתונים העדכניים ברקע וمعدכנת את המסר באופן מיידי ללא צורך בטעינה מחדש של

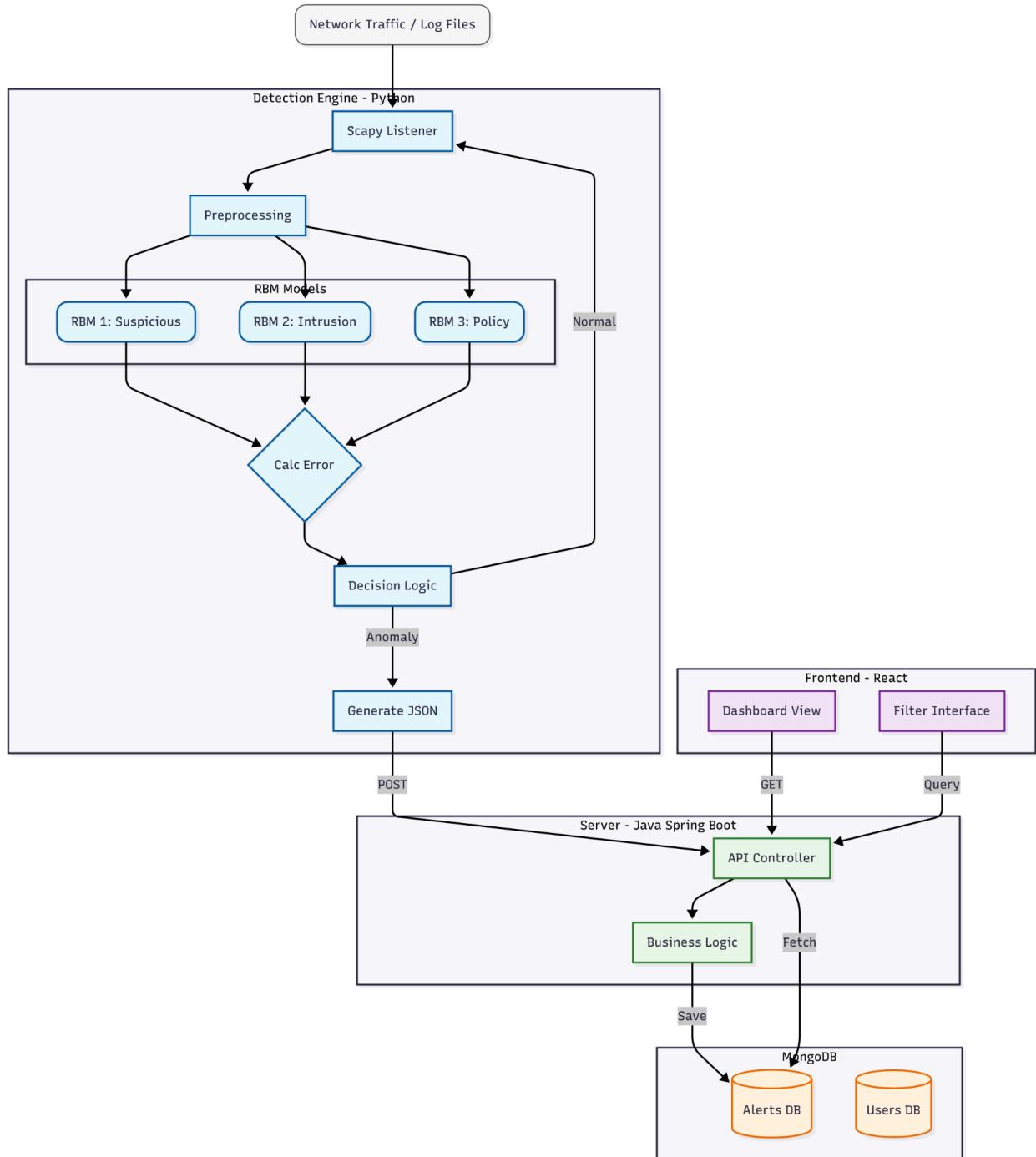
הדף, מה שמבטיח חווית ניטור רציפה וחלקה.

ארquitektورת המערכת והתקשורת

המערכת מבצעת הפרדה בין מנוע הזיהוי לשרת הניהול. זרימת התקשורת בין שירות המזהה (Python) לשירות Backend מתבצעת על גבי פרוטוקול HTTP באמצעות ממשק Restful API.

כasher מודל ה-RBM ב-**Python** מזהה חריגה, הוא מייצר אובייקט **JSON** המכיל את פרטי האירוע (Timestamp, Source IP, Anomaly Score).

מנוע ה-**Python** שולח בקשה **HTTP POST** א-סינכרונית לכתובת ה-**API** של שירות Backend. שירות Backend קולט את הבקשה, שומר את ההתראה ב-**MongoDB** וمعدכן את הדשبورד **React**.



פרטים פורמליים

לוחות זמנים

שלבי עבודה	תאריך סיום	פירוט
חקר מקדים ולמידת טכנולוגיות	16.12.2025	למישהו לעומק של מכונת RBM והבנת המתמטיקה שמאחוריו. חקר הנתונים כדי להבין אילו עמודות מיותרות ואילו לא.
הקמת והנדסת נתונים	24.12.2025	הורדת הדטה סטם. כתיבת קוד לנקיי הדטה, נרמול הדטה.
פיתוח מודל ה-RBM	15.1.2026	שימוש המודלים בעזרת PyTorch אימון שלושת המודלים "ישום פונקציית שגיאת השחזור וקבעת הסף (Threshold)
פיתוח צד השרת	25.1.2026	הקמת פרויקט Spring Boot כתיבת הלוגיקה העסקית חיבור לבסיס הנתונים
ממשק המשתמש	10.1.2026	הקמת אפליקציית React ועיצוב הדשبورד. חיבור לשרת-h-boot
אינטגרציה מלאה	28.1.2026	חיבור הקצוות של השרת והמודלים למדת מכונה. בדיקות המערכת וסימולציות.
תיעוד והגשה	28.1.2026	כתיבת ספר פרויקט והגשה.

חתימת הסטודנט

A handwritten signature in black ink on a light beige background.

חתימת רץ המגמה