

***עזריאלי***

***www.JCE.ac.il***

המחלקה להנדסת תוכנה

**אלגוריתם להתאמת מחרוזות על תעבורת רשת מכווצת**

**חיבור זה מהווה חלק מהדרישות לקבלת**

**תואר שני בהנדסה**

**מאת**

**עילי זידמן**

**טבת תשע"ז** **ינואר 2017**



***עזריאלי***

***www.JCE.ac.il***

המחלקה להנדסת תוכנה

**אלגוריתם להתאמת מחרוזות על תעבורת רשת מכווצת**

**חיבור זה מהווה חלק מהדרישות לקבלת**

**תואר שני בהנדסה**

**מאת**

**עילי זידמן 300634524 izeidman@cisco.com**

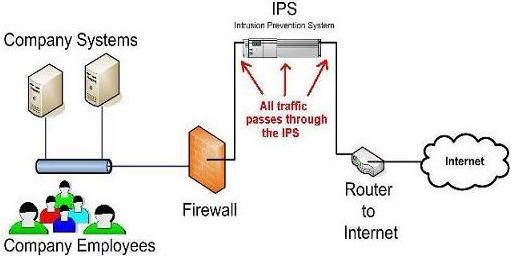
**מנחה**: **ד"ר צור-דוד שמרית**

# 

# תקציר הפרויקט

מערכות איתור חדירות (IDS) פותחו כדי לזהות ולדווח תקיפות ברשת בסוף 1990, כאשר תקיפות האקרים ותולעים ברשת החלו להשפיע על האינטרנט. טכנולוגיות IDS מסורתיות יודעות לזהות תנועה עוינת ולשלוח התראות אבל לא יודעות לעשות דבר כדי לעצור את ההתקפות.

מערכות מניעת חדירות רשת (NIPS) מותקנות אונליין עם קטע הרשת המוגן. כאשר התנועה עוברת דרך המערכת, נבדקת האם יש התקפה. כמו וירוסים, לרוב החדירות יש איזשהו סוג של חתימה. לכן, אלגוריתם-התאמת מחרוזות הוא החלק העיקרי במערכת מניעת חדירות רשת. כאשר התקפה מזוהית, NIPS חוסמת את התקיפה. ישנו trade-off בין הדיוק של הזיהוי ויעילות האלגוריתם. שניהם הם מאוד חשובים כדי להבטיח שתנועה לגיטימית לא מתעכבת או משובשת כאשר היא זורמת ברשת. מסיבה זו, אלגוריתם התאמת המחרוזות חייב להיות מסוגל לפעול במהירות גבוהה, תוך גילוי עיקר החדירות. עם מהירות הרשת המכפילה את עצמה מדי שנה, זה הופך קשה יותר ויותר עבור פתרונות מבוססי תוכנה להיות כאלה מהירים.



כיום כלי אבטחה לא יודעים להתמודד עם תנועה מכווצת, אשר הופכת להיות יותר ויותר נפוצה ב HTTP. פרוטוקול HTTP משתמש בכיווץ GZIP, אשר בהתחלה דורש סוג כלשהו של decompression לפני ביצוע משימת התאמה המחרוזות. לפיכך, זה מאוד לא יעיל להריץ אלגוריתם התאמת מחרוזות על מידע מכווץ.

בפרויקט זה אציג אלגוריתם התאמת מחרוזות על HTTP מכווץ(GZIP). האלגוריתם משתמש ב Ternary Content Addressable Memory (TCAM) והוא מסוגל להתאים מחרוזות רבות בפעולה אחת. האלגוריתם משיג מהירות קו בשיעור של כמה סדרי גודל יותר מהר מאשר עבודות נוכחיות, תוך השגת דיוק דומה של זיהוי. הפרויקט הוא הרחבה של פרויקט שעשתה המרצה שלי בשיתוף עם עוד סטודנטים מהאוניברסיטה העברית שבו האלגוריתם עובד על HTTP לא מכווץ.

האלגוריתם משתמש בעובדה שהחבילות מכווצות ובצורת המימוש של GZIP, ושומר נתונים תוך כדי ריצת האלגוריתם כדי שהוא לא יצטרך לבצע decompression של החבילה.

תוצאות המדידות מראות שהאלגוריתם משפר את הביצועים ביחס לאלגוריתם המקורי גם בכמות הבייטים שהאלגוריתם מדלג וגם בכמות הבדיקות ב TCAM. בנוסף כדי שהאלגוריתם המקורי יעבוד על חבילות מכווצות, הוא צריך לבצע decompression שזו פעולה מאוד יקרה. האלגוריתם שמימשתי לא מבצע decompression של המידע.

# הצהרה

העבודה נעשתה בהנחיית ד"ר צור-דוד שמרית,

במכללה האקדמית להנדסה ירושלים – עזריאלי

המחלקה להנדסת תוכנה.

החיבור מציג את עבודתי האישית

ומהווה חלק מהדרישות לקבלת תואר שני

M.Sc. בהנדסת תוכנה.

# תודות

תודה רבה למנחה הפרויקט ד"ר צור-דוד שמרית על העזרה וההדרכה בפרויקט.

תודה למכללה להנדסה ולכל צוות המכללה.

תודה לאשתי היקרה על התמיכה והעזרה בכל, הכל בזכותך.

תוכן עניינים

[תקציר הפרויקט 3](#_Toc470776466)

[הצהרה 4](#_Toc470776467)

[תודות 5](#_Toc470776468)

[תאור מסגרת הפרויקט: 6](#_Toc470776469)

[הבעיה 6](#_Toc470776470)

[תאור הבעיה: 6](#_Toc470776471)

[הבעיה מבחינת הנדסת תוכנה: 6](#_Toc470776472)

[הפרויקט - מימוש 7](#_Toc470776473)

[דיאגרמת מודולים מרכזיים במערכת: 7](#_Toc470776474)

[מדידות: 7](#_Toc470776475)

[Unit Tests: 7](#_Toc470776476)

[התנהגות: 7](#_Toc470776477)

[מסקנות: 8](#_Toc470776478)

[פתרונות דומים בספרות: 8](#_Toc470776479)

[רשימת ספרות: 8](#_Toc470776480)

[Project summary 8](#_Toc470776481)

# תאור מסגרת הפרויקט:

# הבעיה

## תאור הבעיה:

# תאור הפתרון

הפתרון לבעיה זו, היא לפתח אלגוריתם שידע לזהות ולטפל בחוקי הכיווץ כגון שימוש במצביעים על מנת לא לבצע decompression של המידע ולדלג על כמה שיותר בייטים שהאלגוריתם ידע בוודאות שאין בהם חתימה של וירוסים.

כלומר שה- NIPS יהיה מסוגל לסרוק קובץ מכווץ כמו שהוא כאשר הוא מגיע אליו מבלי להרחיב אותו. מיתרון זה, נוכל להרוויח זמן בשלב הזיהוי של הווירוס ולכן בזמן התגובה של המערכת כאשר היא תחליט לפעול. על מנת לשמור על איכות הזיהוי של חתימות הווירוסים, האלגוריתם יצטרך להיות מסוגל להשתלב עם המערכת SNORT לב פתרון הבעיה הוא שימוש בזיכרון חומרתי הנקרא TCAM . רכיב זיכרון זה הוא מסוג Content-addressable וזמן חיפוש בתוכו הוא O(1) .

## דיאגרמת מודולים מרכזיים במערכת:

## תוצאות שהתקבלו:

מימשתי סימולציה של TCAM בכמה רוחבים שונים ועבור כל אחד הרצתי את האלגוריתם עבור דפי אינטרנט מכווצים ולא מכווצים , ותוך כדי האלגוריתם מדדתי את כמות הדילוגים

## Unit Tests:

# התנהגות:

דוגמא לשימוש במערכת הוא דיאגרמת הרצף (**Sequence Diagram**) הבאה שמתארת משתמש הנכנס לדף שמציג את כל הטסטים(באתר):

# מסקנות:

* מסקנה 1.
* מסקנה 2.

# פתרונות דומים בספרות:

# רשימת ספרות:

# Project summary

My project is about managing and monitoring workspaces in Cisco.

It started from the requirement of automated testers and QA testers to know the state of TV converters that they working with and to do actions on that converter without the need to be at the physical position of the converters.

In my project I implemented web site that shows data on computers and TV converters, for example: how much memory available there is on computer, how much CPU the computer uses and that information is needed for testers to know which workspace is best for them.

Also I have implemented the possibility to do actions from distance on TV converters, for example: to turn on, turn off and restart TV converter.

In this document I defined the problem after it I showed main modules diagram of the system.

I explained broadly the design I made, on the implementation details of the system and on the various technologies that I used in the project. The system implemented in layers: Web site layer, Web services server layer, Data access layer(DAL) and also I implemented computer and STBs monitoring module .



***עזריאלי***

***www.JCE.ac.il***

SOFTWARE ENGINEERING DEPARTMENT

**String Matching Algorithm on compressed HTTP Traffic**

**by**

**Ilay Zeidman**

**Supervisor**: **Dr. Tzur-David Shimrit**



***עזריאלי***

***www.JCE.ac.il***

SOFTWARE ENGINEERING DEPARTMENT

**String Matching Algorithm on compressed HTTP Traffic**

**by**

**Ilay Zeidman**

**TEVET 2017** **January 2017**