

**סמל מוסד - 270538**

**פרויקט גמר**

**הנדסאי תקשוב**

**בהתמחות:** מערכות תקשוב

**הנושא: הקמה, תכנון ופיתוח לחברת “hand controlled”**

**מנחת הפרויקט : סיון שי**

**מגיש/ה:**

שם: אופיר נויבואר ת.ז. 327511713

תאריך ההגשה: מאי תשפ"ה

Contents

[מבוא 5](#_Toc192041412)

[אודות החברה 7](#_Toc192041413)

[תרשים ארגוני 8](#_Toc192041414)

[מחלקות 9](#_Toc192041415)

[סניף הוד השרון 10](#_Toc192041416)

[טבלת כתובות IP בסניף הוד השרון 11](#_Toc192041417)

[מוסכמות השמות 11](#_Toc192041418)

[תמונת טופולוגיית סניף 12](#_Toc192041419)

[HOSTNAME 13](#_Toc192041420)

[BANNER MOTD$ 13](#_Toc192041421)

[Vlan- virtual lan 14](#_Toc192041422)

[נתינת שם ל – vlan 14](#_Toc192041423)

[פרוטוקול VTP – vlan trunking protocol 18](#_Toc192041424)

[STP - spanning tree protocol 22](#_Toc192041425)

[Dot1Q – router on a stick 29](#_Toc192041426)

[Hsrp 34](#_Toc192041427)

[DHCP - Dynamic Host Configuration Protocol 38](#_Toc192041428)

[DNS – (Domain Naming Server) 41](#_Toc192041429)

[mail 42](#_Toc192041430)

[http/https(Hyper Text Transfer Protocol/Secure) server 44](#_Toc192041431)

[FTP(File Transfer Protocol) server 44](#_Toc192041432)

[TFTP(Trivial File Transfer Protocol) server 45](#_Toc192041433)

[Syslog server 46](#_Toc192041434)

[NTP server 47](#_Toc192041435)

[סניף תל אביב 50](#_Toc192041436)

[מוסכמות השמות 50](#_Toc192041437)

[טבלת השוואה מתג שיכבה מתג שיכבה 3 שיכבה 2 50](#_Toc192041438)

[תמונה של טופולוגיה בתולית 51](#_Toc192041439)

[הגדרות שכבר הופיעו 52](#_Toc192041440)

[SVI – Switch Virtual Interface 60](#_Toc192041441)

[פקודה no switchport 60](#_Toc192041442)

[פקודה ip helper-address 61](#_Toc192041443)

[הגדרות הסרבר dhcp 61](#_Toc192041444)

[שרת dns 62](#_Toc192041445)

[שרת http 63](#_Toc192041446)

[שרת ftp 64](#_Toc192041447)

[שרת mail 65](#_Toc192041448)

[שרת tftp 69](#_Toc192041449)

[שרת syslog 71](#_Toc192041450)

[שרת ntp 71](#_Toc192041451)

[טבלת השוואת פרוטוקולי ניתוב 73](#_Toc192041452)

[סיכום eigrp 74](#_Toc192041453)

[סניף קפריסין לימאסול 79](#_Toc192041454)

[טופולוגיה בתולית 79](#_Toc192041455)

[מוסכמות השמות 79](#_Toc192041456)

[הגדרות בסיסיות 80](#_Toc192041457)

[הגדרות שרת ה - dhcp 88](#_Toc192041458)

[שרת dns 88](#_Toc192041459)

[שרת http 89](#_Toc192041460)

[שרת ftp 90](#_Toc192041461)

[שרת mail 91](#_Toc192041462)

[שרת TFTP 95](#_Toc192041463)

[שרת syslog 95](#_Toc192041464)

[שרת NTP 96](#_Toc192041465)

[פרוטוקול ניתוב OSPF- Open Shortest Path First 97](#_Toc192041466)

[Static Route 105](#_Toc192041467)

[ACL – Access Control List 106](#_Toc192041468)

[BGP – Border Gateway Protocol 108](#_Toc192041469)

[NAT – Network Address Translation 114](#_Toc192041470)

[ 115](#_Toc192041471)

[GRE – Genereic Routing Encapsulation 116](#_Toc192041472)

[VPN – Virtual Private Network 117](#_Toc192041473)

[Etherchannel 123](#_Toc192041474)

[הגנות והתקפות סייבר 127](#_Toc192041475)

[ASA 131](#_Toc192041476)

[SSH 135](#_Toc192041477)

[AAA 138](#_Toc192041478)

[DHCP snooping 141](#_Toc192041479)

[native vlan 142](#_Toc192041480)

# מבוא

**רקע עצמי**

תמיד הייתי בן אדם שהתעניין בטכנולוגיה וקידומת אבל המסע שלי בעולם הטכנולגיה לא התחיל עד כיתה י' שם למדתי בפעם הראשונה שפת תיכנות. למדתי שם c# ועם כל מה שאפשר לעשות עם השפה הזאת הרגשתי שעולם מפואר חדש נפתח מולי. מאז אני רק לומד ומתנסה יותר ויותר בכל מה שאני יכול בין אם זה שפות תכנות נוספות לרפרטואר שלי או פרוטוקולי אינטרנט שיעזרו לי להבין אם רק טיפה יותר טוב את הדרך בה האינטרנט מסביבנו עובד.

אני במיוחד התאהבתי בעולם האינטרנט לאחר שעשיתי את הפרוייקט סיום שלי בכיתה י"ב. הפרוייקט היה אתר למידה פשוט עם מערכת משתמשים ובסיס מידע. למרות שהוא היה כל כך פשוט אני רק רציתי לדעת יותר. דבר זה הוביל אותי לרצות ללמוד במכללת כינרת בגלל המוניטין המרהיב שלה בלימוד חומר מעניין ומוסף וגם יצירת תלמידים חכמים במיוחד. ברגע שגיליתי שמכללה גם מציע מסלול של מנהל רשת ישר ניסיתי להיתקבל אליו מכיוון שהוא ענה על כל מה שרציתי.

למכללה באתי עם ידע קודם:

**ידע קודם**

שפות תיכנות:

* C#: ידע בינוני
* Python: ידע בסיסי
* Html: ידע בסיסי
* Javascript: ידע בסיסי
* Css : ידע בסיסי
* SQL : ידע בסיסי

לאחר שלמדתי במכללה אגרתי ידע נוסף שעזר לי להרחיב את הרפרטואר שלי ועזר לי להיות יכול לעשות דברים נוספים מעבר למה שיכולתי לפני:

**ידע נלמד י"ג י"ד**

שפות תכנות:

* C#: ידע בינוני
* Python: ידע בינוני
* Html: ידע מתקדם
* Javascript: ידע בינוני
* Css : ידע מתקדם
* SQL : ידע מתקדם
* React : ידע מתקדם

שרתים ופרוטוקולים במערכת הפעלה linux:

* שרת web
* שרת Ftp
* שרת gmail
* שרת dns
* יישום מערכת גיבוי
* יישום פרוטוקול snmp
* יישום firewall בסיסי
* יישום מערכת מניעת מתקפת brute force

נושאים נוספים שלמדתי במהלך שנים אלה הם:

* Firewall
  + למדתי להשתמש ב – firewall של Fortinet
* בינה מלאכותית
  + אלגוריתמי בינה מלאכותית כגון: מציאת דרך אופטימלית או מהלך אופטימלי הבא במשחק
  + בניית מודלים של בינה מלאכותית
* מנהל רשת
  + פרוטוקולי רשת
  + הגדרת רכיבי רשת
* אבטחת מידע
  + התקפות שונות ודרכי ההתמודדות איתן
  + כלים להבנת מתקפות ולמה הן קרו

**מה קיבלתי מהלימודים**

בלימודים נתנו לי הרבה מידע ששמחתי לקבל, מידע שבא עם מוטיבציה ללמוד עוד ולנסות עוד, המידע גם הביא לי את הביטחון לעתיד מכיוון שגם אם אנסה לימצוא עבודה כרגע אז יהיו לי הרבה אפשרויות לבחור מהן ולא רק מקצוע אחד.

**שאיפות לעתיד**

בעתיד אני רוצה להתנסות בעוד תחומי טכנולוגיה, נוסף על כך ארצה לנסות את ידי בכמה פרוייקטים שונים שיאתגרו אותי, כמובן שאשמח אם פרוייקטים אלו יוכלו גם לעזור לאנשים אם אוכל לעזות זאת באותה העת

# אודות החברה

**פירוט כללי של החברה**

החברה שלנו hand controlled מחפשת לעזור ולהפוך את החיים היום יומיים של כל אחד ואחד לקלים יותר, ללא שימוש בשלטים מסובכים או פאנל שצריך לנווט וללמוד בשביל לתפעל דברים פשוטים בחיי היום יום כמו טלוויזיה או מזגן, החברה שלנו מייצרת מוצרים שנשלטים בפשטות על ידי תנועה או הבעה קלה של היד, מה שמוריד לכם את הצורך בשלט ואת הצורך בלמידה המפרכת.

החברה hand controlled כוללת 3 סניפים מרכזיים אשר אחראיים על תפקוד החברה:

הסניף הראשון נמצא בישראל בעיר הוד השרון. הוא אחראי על מכירת המוצרים שלנו.

הסניף השני גם הוא נמצא בישראל בעיר תל אביב הוא אחראי על אחסון מוצרינו.

הסניף השלישי הנמצא בקפריסין הוא אחראי על מחקר ופיתוח המוצרים שלנו.

החוברה מונה עד כה כ – 300 עובדים בשני הסניפים אשר נמצאים בישראל בהוד השרון ותל אביב הראשונים וכ – 400 עובדים בסניף השלישי אשר נמצא בקפריסין בכל סניף.

עד כה החברה שלנו משרתת כ – 11000 לקוחות שונים שמנויים אצלנו נוסף על כך נמכרו כ 33000 מוצרים שונים ללקוחות מרוצים וחוזרים בשמחה.

**פירוט המוצר**

המוצר החדש שלנו: handy robo הוא רובוט קטן היכול לשמש כצעצוע או עוזר קטן בבית. זה הוא רובוט שבעזרת תנועות פשוטות של היד אשר אתם קובעים.

Handy robo מפותח על ידי מיטב המוחות שלנו עם הטכנולוגיה הכי חדישה לזיהוי תנועות והבעות ידיים בשביל השמחה המיטבית שלכם.

ל – handy robo יש כמה מטרות:

1. להכיר אתכם בקלות למוצרים שלנו עם חוויה משעשעת מלאת צחוקים והנאה.
2. להראות לכם את הפוטנציאל שלנו כחברה מתחילה.
3. לתת לכם להנות מהרובוט הקטן והחמוד הזה.

# תרשים ארגוני

מנכ"ל

מנהל מכירות

סניפים

מנהל פיתוח

סניף קפריסין

סניף הוד השרון

סניף תל אביב

סניף מחקר ופיתוח

סניף אחסון

סניף מחירות

Human resources

Human resources

Human resources

managers

managers

managers

storage

storage

storage

Quality assurance

Warehouse workers

workers

research

drivers

sales

development

inventory

reception

# מחלקות

|  |  |  |
| --- | --- | --- |
| **שם מחלקה מלא** | **שם מקוצר** | **מה המחלקה עושה** |
| human resources | HR | שומרת על זכויות העובדים |
| managers | MAN | אחראית על ניהול העובדים |
| storage | STO | אחראית על ניהול המלאי המקומי של מוצרים בסיסיים |
| Workers | WOR | אחראית על שירות לקוחות |
| Sales | SAL | אחראית על מחירות הניהול וניטור שלהן |
| Reception | REC | אחראית על קבלת הלקוחות לפני שיש עובד שיכול לעזור להם |
| Warehouse workers | WW | אחראית על כל התעבורה של המוצרים במחסן והסידור שלהם |
| Drivers | DRI | אחראית על הבאת ושליחת המוצרים של המחסן |
| Inventory | INV | אחראית על ניהול המלאי של המוצרים במחסן |
| Quality assurance | QA | אחראית על הבטחת האיכות של המוצרים שיוצאים לשוק |
| Research | RES | אחראית על מחקר טכנולוגיה חדשה שתשומש למוצרים |
| development | DEV | אחראית על פיתוח המוצרים שלנו |

# סניף הוד השרון

# טבלת כתובות IP בסניף הוד השרון

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vlan ID | Pc count | Network ID | Subnet mask | Default gateway |
| 11 | 6 | 10.1.11.0 | 255.255.255.0 | 10.1.11.254 |
| 37 | 6 | 10.1.37.0 | 255.255.255.0 | 10.1.37.254 |
| 39 | 6 | 10.1.39.0 | 255.255.255.0 | 10.1.39.254 |
| 53 | 6 | 10.1.53.0 | 255.255.255.0 | 10.1.53.254 |
| 114 | 6 | 10.1.114.0 | 255.255.255.0 | 10.1.114.254 |
| 123 | 6 | 10.1.123.0 | 255.255.255.0 | 10.1.123.254 |

# מוסכמות השמות

|  |  |  |  |
| --- | --- | --- | --- |
| שם הרכיב | שיכבה בטופולוגיה | הסבר השם המקוצר | הסבר על השם בקיצור |
| PC |  | pc<n>-<bn>-<dpt> | N – pc number  Bn - branche name(HH-Hod Hasharon,TL – Tel Aviv,CY – Cyprus)  Dpt – department name |
| Switch 2960 | Access | SW<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Switch pt empty | Distribution | SW<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Router 1941 | Core | R<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |

# תמונת טופולוגיית סניף

A diagram of a network

Description automatically generated

# HOSTNAME

## הסבר הפקודה

פקודה זו נותנת שם לרכיב בו היא נכתבת

## איך כותבים

hostname <host\_name>

## תמונה של הפקודה



## תמונה של האפקט של הפקודה



# BANNER MOTD$

## הסבר הפקודה

הפקודה גורמת לכך שמי שנכנס ל – cli של הרכיב רואה הודעה לפי מה שנכתב בפקודה

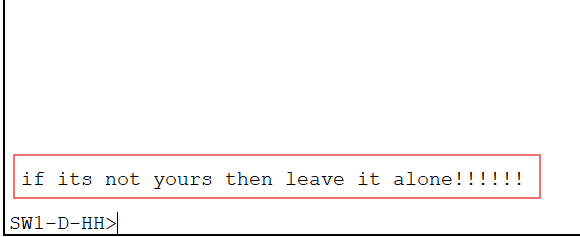
## איך כותבים

banner motd $<message>$

## תמונה של הפקודה



## תמונה של האפקט של הפקודה



# Vlan- virtual lan

זה הוא פרוטוקול המאפשר הפרדה של הרשת לרשתות שונות קטנות יותר

Vlan זו הפרדה לוגית ולא פיזית. ה – vlan מקטין את ה – broadcast domain מה שמקטין את העומס ברשת ומקל על הרכיבים הנמצאים בה.

ההפרדה הלוגית של vlan מאפשרת רק למחשבים מאותו vlan לקבל הודעות ממחשבים באותה רשת גם אם יש מחשבים נוספים המחוברים פיזית אך לא באותה רשת מה שעוזר להבטחת המידע

בהוספת ה vlan למערכת של ה – המתג היווסף מצב port חדש מצב trunk

**מצב trunk**

כאשר פורט הוא במצב trunk אז בניגוד למצב access הוא יכול להעביר frame מ – vlan שונים. לכן בגלל שבעזרת מצב trunk מתגים יכולים להעביר אחד בין השני frame מ – vlan שונה אז המידע שעובר בין המתגים מגיע רק לרשת שהוא צריך להגיע אליו

## יצירת vlan

## הסבר הפקודה

בשביל ליצור vlan נשתמש בפקודה זו

## איך כותבים

vlan <number>

## תמונה של הפקודה



## תוצאות הפקודה

A white background with black lines

Description automatically generated with medium confidence

# נתינת שם ל – vlan

## הסבר הפקודה

נשתמש בפקודה זו לתת שם ל – vlan כאשר אנו נימצאים במצב הגדרות ל – vlan ספציפי

## איך כותבים

Name <name>

## תמונה של הפקודה



## תוצאות הפקודה

A black and white image of a line

Description automatically generated with medium confidence

## הקצאת פורט ל – vlan

**הסבר**

נשתמש בפקודה זו במצב הגדרות לפורט כאשר נרצה להקצות את בפורט ל - vlan

**איך כותבים**

Switchport access vlan <vlan number>

**תמונה של הפקודה**



**תוצאות הפקודה**

A close up of words

Description automatically generated

## פקודות show של vlan

**show vlan brief:**

A screenshot of a computer

Description automatically generated

**הסבר:**

vlan – מספר ה – vlan

name: שם ה – vlan

status: המצב בו נימצא ה – vlan

ports: אלו הפורטים המשוייכים לאותו ה – vlan

**show vlan id <id number>**

**A screenshot of a computer program

Description automatically generated**

**הסבר**

מראה את אותו הדבר כמו show vlan brief כמו שניתן לראות בסימון האדום.

בנוסף מראה שדות לפרוטוקולים אחרים שיכול להיות שישפיעו על ה – vlan כמו שניתן לראות בסימון הכחול

המידע הוא ספציפי ל – vlan עם אותו id שניבחר בפקודה

**show vlan name <vlan name>**

**A close-up of a white background

Description automatically generated**

**הסבר**

ניתן לראות את אותו המידע כמו בפקודה show vlan id <id number> רק שפה ה – vlan נבחר לפי שמו ולא לפי המספר המזהה שלו

**show vlan**

**A screen shot of a computer

Description automatically generated**

A screenshot of a computer screen

Description automatically generated

**הסבר**

פקודה זו משלבת מידע מכל שאר הפקודות show של ה – vlan אך היא עושה זאת עבור כל ה – vlan שקיימים במתג

# פרוטוקול VTP – vlan trunking protocol

פרוטוקול vtp הוא פרוטוקול המעביר עדכונים של טבלאות vlan ברשת המקומית על גבי פורטים שנמצאים במצב trunk

בפרוטוקול זה יש היררכיה בין 3 סוגים של מתגים:

## מתג server

מתג במצב זה מעביר עדכונים אשר כותבים בו למתגים אחרים במצב client. במצב זה בכל פעם שנעשה שינוי בטבלת vlan של מתג זה , הוא שולח את העדכונים ברשת המקומית לשאר המתגים. כל מתג אוטומטית נימצא במצב זה.

## מתג client

מתג במצב זה מקבל את עדכוני טבלת ה – vlan מהמתג במצב server ומכיל אותם על עצמו, לא ליצור או לשנות את טבלת הוילאנים במתג במצב זה, הוא רק מקבל את העיכונים ומעביר אותם

## מתג transparent

מתג במצב זה מקבל את העדכונים מהמתג במצב server אך הוא לא מכיל אותם על עצמו ורק מעביר את העדכונים הלאה, על מתג במצב transparent בניגוד למתג במצב client ניתן ליצור vlan הניפרדים מה – vlan של המתג במצב server

## התנאים להעברת עדכונים על טבלת ה - vlan

קיימים כמה תנאים להעברת עדכונים בטבלת vlanבין מתגים בפרוטוקול vtp

1. Domain name – שם ה - domain בין שני מתגים חייב להיות שווה בשביל שיוכלו להעביר עדכונים בניהם
2. Password- לכל מתג בפרוטוקול vtp יכול להיות סיסמה, אם היא קיימת הסיסמה בים שני מתגים חייבת להיות שווה.

## יכולת vtp pruning

זו יכולת של הפרוטוקול vtp במתג להעביר את ההודעות שהמתג מקבל רק ל – vlan המתאים להודעות אלו. vtp pruning מוריד את התעבורה ברשת ובכך מוריד עומס.

## יתרונות

מאפשר ניהול של vlan נוח יותר ברשת כי ניתן ליצור את ה – vlan במתג אחד והוא מתעדכן בכל שאר המתגים ברשת המקומית

## החלפת מצב מתג

**הסבר הפקודה**

נשתמש בפקודה זו בשביל להחליף את מצב המתג לאחד מהמצבים האלה: server,client,transparent

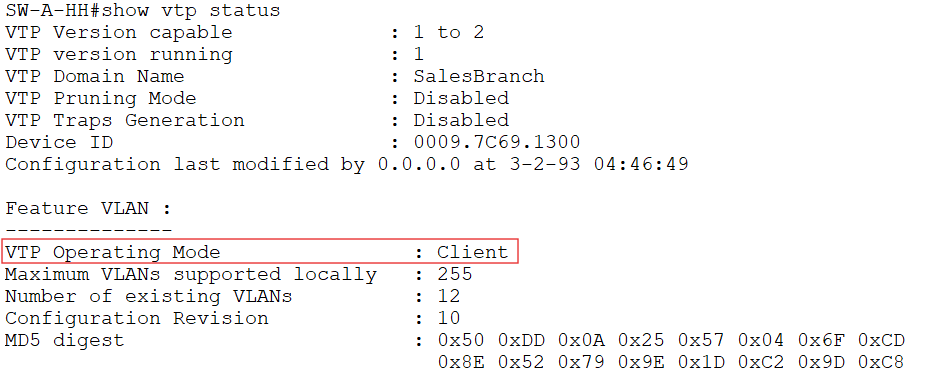
**איך לכתוב**

vtp mode <mode name>

**תמונה של הפקודה**



**תוצאות הפקודה**



## קביעת/שינוי שם הדומיין

**הסבר הפקודה**

נשתמש בקודה זו בשביל לשנות את שם הדומיין של המתג לשם אותו אנו רוצים

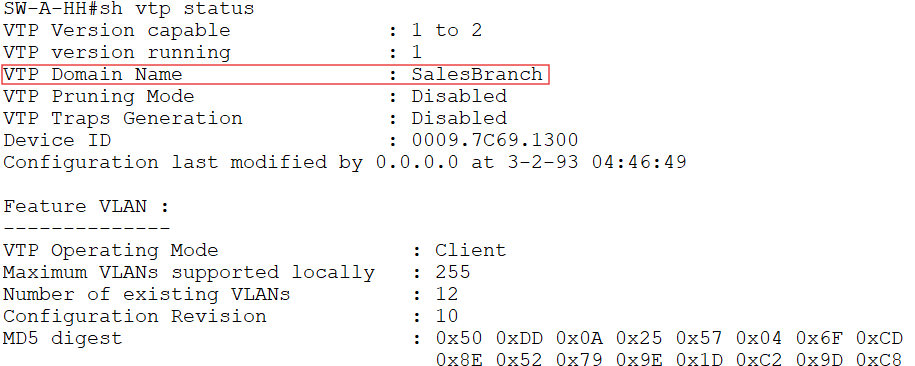
**איך כותבים**

vtp domain <domain name>

**תמונה של הפקודה**



**תוצאות הפקודה**



## שינוי/קביעת הסיסמה של vtp

**הסבר הפקודה**

משתמשים בפקודה זו בשביל לקבוע או לשנות את הסיסמה של ה – vtp במתג

**איך כותבים**

vtp password <password>

**תמונה של הפקודה**



**תוצאות התמונה**



## פקודות show של vtp

**show vtp counters**

A screenshot of a computer program

Description automatically generated

**הסבר**

השדה summary advertisements received – אומר את הכמות ההודעות סיכום של עדכון מידע בפרוטוקול vtp שהמתג קיבל

השדה subset advertisements received – אומר את כמות ההודעות שמפרטות על שינויים בטבלת ה – vlan שהמתג קיבל

השדה request advertisements received – אומר את כמות ההודעות לבקשת קבלת עידכון המידע ב – vtp שהמתג קיבל ממתגים אחרים

השדה summary advertisement transmitted - אומר את הכמות ההודעות סיכום של עדכון מידע בפרוטוקול vtp שהמתג שלח

השדה subset advertisements transmitted - אומר את כמות ההודעות שמפרטות על שינויים בטבלת ה – vlan שהמתג שלח

השדה request advertisements submitted - אומר את כמות ההודעות לבקשת קבלת עידכון המידע ב – vtp שהמתג שלח למתגים אחרים

השדה number of config revision errors – אומר את כמות הפעמים שהייתה בעיה בשינוי הקונפיגורציה של ה- vtp

השדה number of config digestion errors – אומר את כמות הפעמים שהייתה בעיה בהודעת עידכון ה – vtp שהתקבלו

השדה number of v1 summary errors – אומר את כמות הפעמים שקרתה בעיה כי קיבל הודעת עידכון vtp של vtp גירסה 1 כאשר המתג בגרסה יותר מתקדמת ולכן הייתה בעיה לפרש את ההודעה

השדות מתחת לשדה vtp pruning statistics

השדה trunk – מספר הפורטים במצב trunk שעשו vtp pruning

השדה join transmitted – אומר את מספר ההuדעות של נתינת מידע על vlan שנישלח מהמתג

השדה join reveived– אומר את מספר ההuדעות של נתינת מידע על vlan שהמתג קיבל

השדה summary advts received from none pruning capable device – אומר את מספר ההודעות עידכון מידע ב – vtp שהיתקבלו ממתגים שלא תומכים ב – vtp pruning

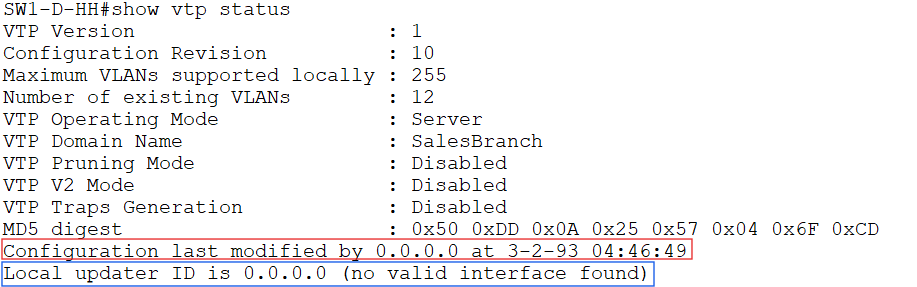
**show vtp password**



**הסבר**

מראה את הסיסמה של ה – vtp

**show vtp status**



**הסבר**

השדה vtp version – אומר באיזה גירסה של vtp המתג משתמש

השדה configuration revision – אומר את מיספר הפעמים שבוצע שינוי למבנה של ה – vtp מהמצב ההתחלתי

השדה maximum vlan supported locally – אומר את כמות ה – vlan המקסימלית שיכולים להיות במתג

השדה vtp operating mode – אומר את המצב בו נימצא המתג: server, client, transparent

השדה vtp domain name – אומר מה השם של ה – domain של ה - vtp במתג

השדה vtp pruning mode – אומר האם vtp pruning מאופשר במתג

השדה vtp v2 mode – אומר האם vtp גירסה 2 מאופשר על המתג

השדה vtp traps generation – אומר האם המתג יודיע לרכיב של snmp על עידכונים ב – vtp

השדה md5 digest – מראה ערך hash שמועבר בהודעת vtp כדי להבטיח את שלמות ההודעה

# STP - spanning tree protocol

**Broadcast storm** - כאשר יש שני מתגים מחוברים במקביל ונשלחת הודעת broadcast ההודעה משתכפלת ללא הכרה ונוצר עומס של הודעות broadcast מה שמעמיס את על הרשת ויכול גם לגרום לה לקרוס



בשביל לפתור את בעיה זו המציאו את פרוטוקול stp

**איך עובד**

פרוטוקול stp קובע מתג אחד root bridge כך שכל הפורטים שלו פתוחים כל שאר המתגים בודקים אם נוצרת להם לופ ואז הם חוסמים את אחד מהפורטים שלהם שנמצאים בלופ

בכדי לבחור את את ה – root bridge כל מתג שולח הודעה שקוראים לה הודעת bpdu שבה נימצא ה – BID(Bridge ID) והוא בנוי מ – [priority][mac address] והמתג עם ה – BID הכי נמוך הוא מי שניבחר כה – root bridge

ברגע שנבחר root bridge ברגע זה כל הפורטים של המתג עוברים למצב forward וכל שאר המתגים נכנסים לתהליך בו יחליטו איזה מצב לתת לאיזה פורט שלהם:

1. לכל כבל יש cost כך שעל כל מתג הפורט שה – cost ממנו ל – root bridge הוא הכי נמוך הופך להיות במצב root port
2. הפורטים האחרים הופכים להיות במצב designated port
3. לאחר מכן כאשר מזהה לולאה, מגיעים למתג בלולאה בו המחיר של ה – designated port הוא הכי גבוהה מבין כל המתגים בלולאה, אם יש כמה פורטים בלולאה בהם מחיר ה – designated port הוא הכי נמוך אז הפורט הנימצא במתג עם ה– BID הכי גבוהה והפורט הזה עובר למצב blocked

תהליך בחירת מצבי הפורטים קורה כך:

forwarding: במצב זה הפורט ישר עובד למצב blocking

Blocking: במצב זה הפורט עובר ישר למצב listenin במצב

Listening: המתגים מקשיבים להודעות BPDU אבל לא שומרים כתובות mac מצב זה נימשך ל – 15 שניות דיפולטיבית

Learning: במצב זה הפורט לומד כתובות mac והוא נימצא במצב זה למשך 15 שניות דיפולטיבית

Forwarding: לאחר מצב learning הפורט עובר למצב forwarding

שים לב!!!

Hello time: דיפולטיבית

Max hello time: 20xhello time

זאת אומרת שזמן ההיתכנסות של stp דיפולטיבית הוא 50 שניות

קיימות כמה גירסאות של stp:

Stp: הגירסה הראשונה שיצאה

Rstp: stp רק שזמן ההיתכנסות שלו הוא מהיר יותר - 2 שניות

Pvst: stp רק עם אפשרות ל – vlan ככה שיכול להיות root bridge שונה ל – vlan שונים ככה שאפשר גם לעשות load balancing

Rpvst: pvst רק שזמן ההתכנסות שלו הוא מהיר יותר – 2 שניות

**PVST – Per Vlan STP**

זוהי גירסה של stp המותאמת ל – vlan כך שיכול להיות מתג שונה שיהיה root bridge לכל vlan שונה. בנוסף ניתן גם ליצור מתג גיבור כך שאם המתג שהוא root bridge נופל אז מתג אחד יכול לקחת את מקומו את שהמתג חוזר.

**סיבות להישתמש:**

* Pvst נותן לנו לעשות Load balancing כך שלא כל ההודעות מכל vlan יעברו דרך מתג אחד
* Pvst נותן לנו לעשות גיבוי כך שגם אם מתג שהוא root bridge נופל, יש מתג אחד שיהפוך ל – root bridge כך שאנו יכולים להעלות את המתג שנפל כשבכל הזמן הזה לא הייתהפגיעה בתקשורת ברשת

**Root guard**

אם במקרה האקר הצליח לקבל גישה למתג הוא יכול לשלוח הודעת bpdu נמוכה מזאת של ה – root bridge ובכך להפוך אטת המתג שיש לו גישה אליו ל – root bridge ולקבל גישה לכל ההודעות שעוברות ברשת

כדי למנוע את מצב זה יש לנוע את ההגדרה של root guard מגדירים את ההגדרה הזאת בפורטים במתגים בין שכבת ה – distribution והשיכבה access. הגדרה זו נותנת להודעות bpdu לעבור בין השיכבה access לשיכבה distribution אך היא חוסמת הודעות bpdu שייגרמו לשינוי ה – root bridge

**Bpdu guard**

במצב נורמאלי מחשב יוכל להתחבר למתג ולשלוח הודעות bpdu למתג כמובן שלא תהיה לו סיבה לעשות זאת, אך אם האקר רוצה הוא יכול לשלוח הודעת bpdu דרך מתג בשכבת access ולהפוך את המחשב שלו ל – root bridge ובכך ללהאזין לכל ההודעות ברשת.

כדי למנוע זאת יש לנו את הגדרת bpdu guard הגדרה זאת מונעת מפורטים שהגדרה זו הוגדרה עליהם מלשלוח הודעות bpdu

**Portfast**

כאשר המתגים עולים ומתחילים לעבוד לוקח ל – stp כ – 50 שניות עד שיהיה אפשר להעביר הודעות ברשת אך המחשבים שנמצאים באותו מתג לא צריכים stp בשביל לשלוח הודעות אחד לשני, לכן יש את הגדרת portfast שנותנת לפורטים שהגדרה זאת מוגדרת עליה ישר להתחיל לעבוד בלי להפעיל stp ובכך מונעים את הצורך של המשתמש במחשב לחכות עד שה – stp יסיים להתכנס

## Spanning-tree mode

**הסבר**

מחליף בין סוגי ה – stp

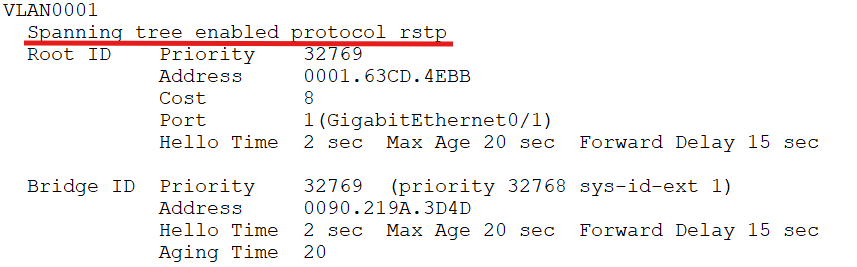
**תחביר**

Spanning-tree mode [mode name]

**תמונה של הפקודה**



**תמונה של אפקט הפקודה**



## Spanning-tree vlan

**הסבר**

הפקודה הזאת נותנת לנו לקבוע מתג שיהיה ה – root bridge בנוסף היא גם נותנת לנו לקבוע מתג גיבוי

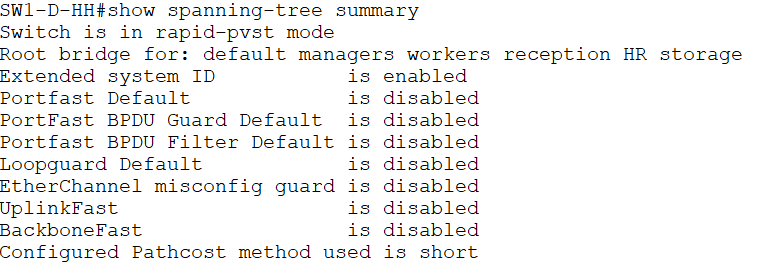
**תחביר**

spanning-tree vlan [vlan number],[vlan number],… root [primary/secondary]

**תמונה של הפקודה**

****

**תמונה של אפקט התמונה**



## Spanning-tree guard root

**הסבר**

משתמשים בפקודה זאת בהגדרה של פורט בשביל להגדיר שהו ישתמש ב – root guard

**תחביר**

Spanning tree guard root

**תמונה של הפקודה**

****

## spanning-tree bpduguard

**הסבר**

משתמשים בפקודה הזאת על ההגדרה של פורט בשביל להפעיל או לכבות bpdu guard על פורט ספציפי

**תחביר**

spanning-tree bpduguard [enable/disable]

**תמונה של הפקודה**



## Spanning-tree portfast

**הסבר**

משתמשים בפקודה זו על הגדרה של פורט כדי להעביר להפעיל עליו את ההגדרה Portfast

**תחביר**

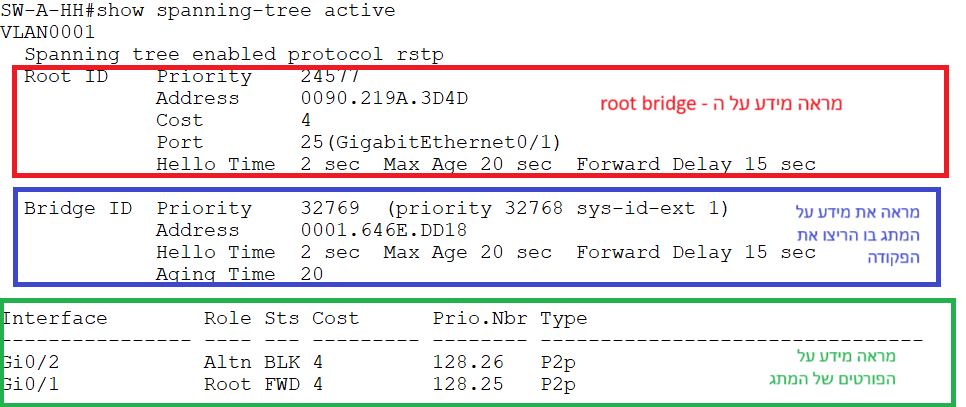
Spanning-tree portfast

**תמונה של הפקודה**



## פקודות show

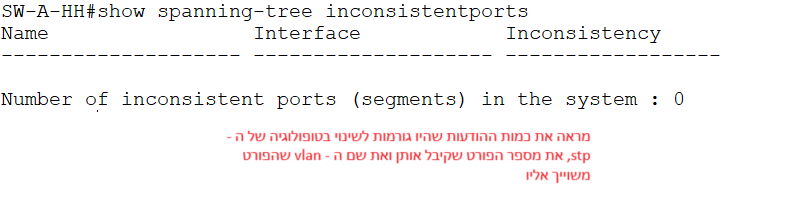
**Show spanning-tree active**

****

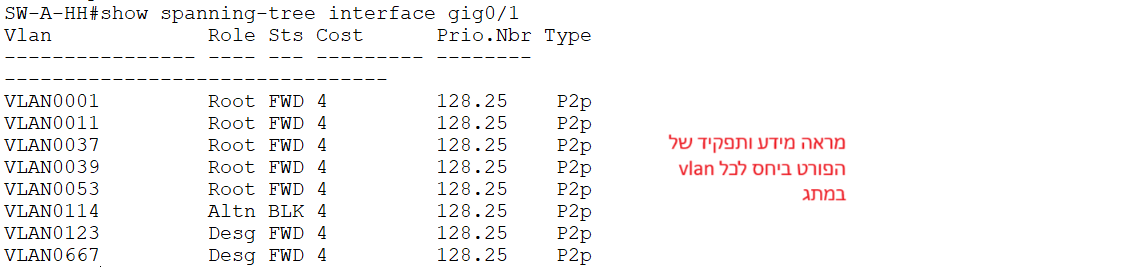
**Show spanning-tree detail**

****

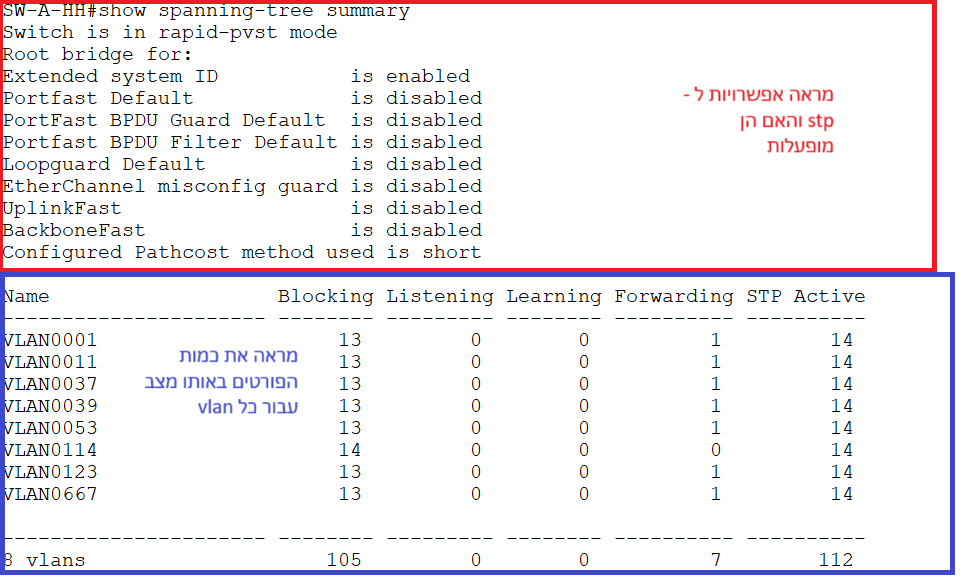
**Show spanning-tree inconsistentports**

****

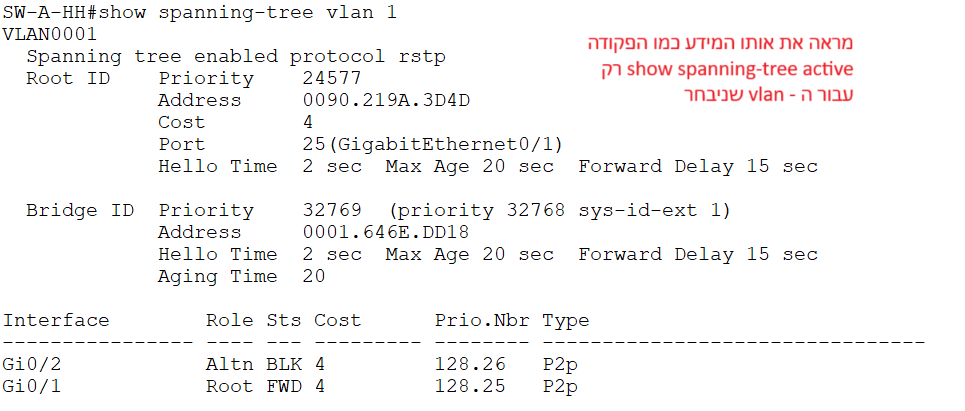
**Show spanning-tree interface [interface number]**

****

**show spanning-tree summary**

****

**show spanning-tree vlan [vlan number]**

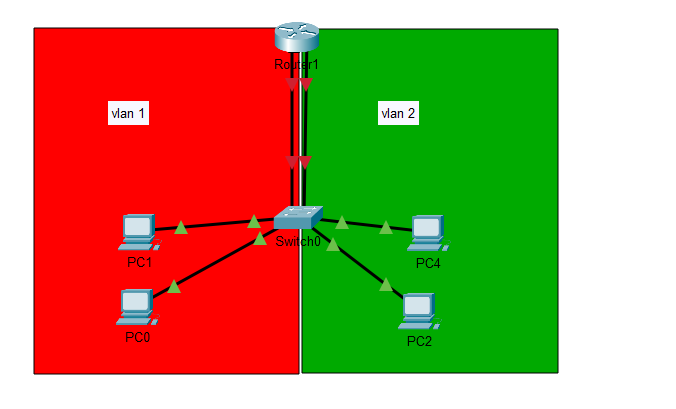
****

# Dot1Q – router on a stick

## הסבר

לאחר שהתווספה האפשרות להוסיף vlan בתוך הרשת הפרטית נוצרה בעיה חדשה, מכיוון שעוד לא היתווסף המצב trunk, כך שכל מתג היה חייב כבל לכל vlan שונה. לכן היו צריכים לחבר את המתג לראוטר בשני כבלים ככה שההודעה נכנסת לראוטר מפורט אחד ויוצאת ממנו בחזרה למתג מפורט אחר השייך לרשת vlan השני.

כך שהיו מחברים רשת בצורה הזו:



בצורה הזו אנחנו משתמשים בשני פורטים של הראוטר כך שנצטרך כמות שווה של פורטים על הראוטר לכמות ה – vlan שיש לנו ברשת בפרטית.

בשביל שלא נצטרך כמות כל כך גדולה של פורטים על הראוטר באה התקינה ה – 802.1q. ה – תקינה זו נותנת לנו את האפשרות להעביר פורט למצב trunk ככה שהודעות מרשתות vlan שונות יכולות לעבור על אותו כבל אך הן עדיין לא יכולות לעבור מרשת vlan אחת לרשת vlan אחרת.

לכן משתמשים בפרוטוקול ה – dot1Q שנותן לנו לעבור בין רשתות vlan על גבי פורט אחד.

A diagram of a computer network

Description automatically generated

ככה אותה טופולוגיה מתמונה קודמת תראה עם dot1q

## איך עובד

הפרוטוקול יוצר sub ports על גבי פורט פיזי עבור כל רשת vlan פרטית כך שכל הודעה ברשת הפרטית מגיעה לפורט הוירטואלי שלה וכאשר היא עוברת לרשת vlan אחרת היא יוצאת מהפורט הוירטואלי של ה – vlan אליו היא ניכנסת.

## Interface

**הסבר**

הפקודה הזאת נותנת לנו להיכנס להגדרת פורט של ראוטר ובעזרתה אפשר גם ליצור פורט וירטואלי

**תחביר**

כניסה לפורט:

Interface [interface number]

יצירת פורט וירטואלי:

Interface [interface number].[vlan number]

**תמונה של הפעולה**

כניסה לפורט:



יצירת פורט וירטואלי:



## Encapsulation dot1Q

**הסבר**

כותבים את פעולה זו בקונפיגורציה בפורט וירטואלי בשביל להפעיל עליו את פרוטוקול dot1q

**תחביר**

Encapsulation dot1Q [vlan number]

**תמונה של הפקודה**



## Ip address

**הסבר**

משתמשים בפעולה זו בשביל להגדיר את הכתובת default gateway לאותו פורט או פורט וירטואלי בו נימצאים בקונפיגורציה שלו בראוטר

**תחביר**

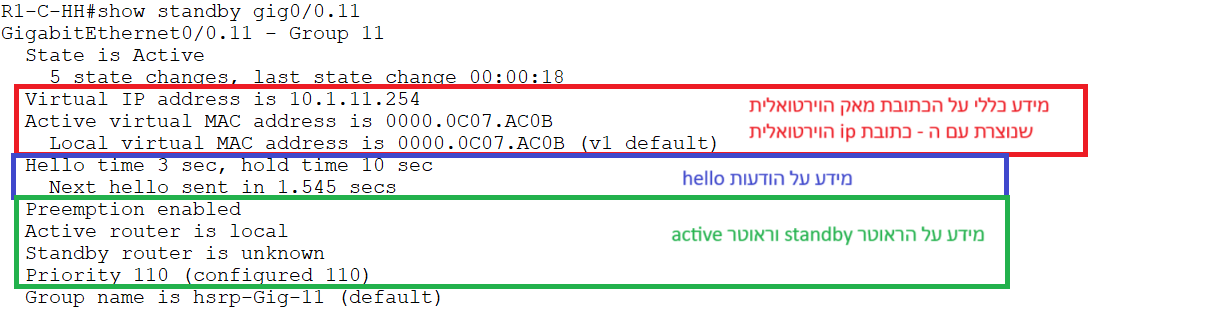
Ip address [default gateway] [subnet mask]

**תמונה של הפעולה**

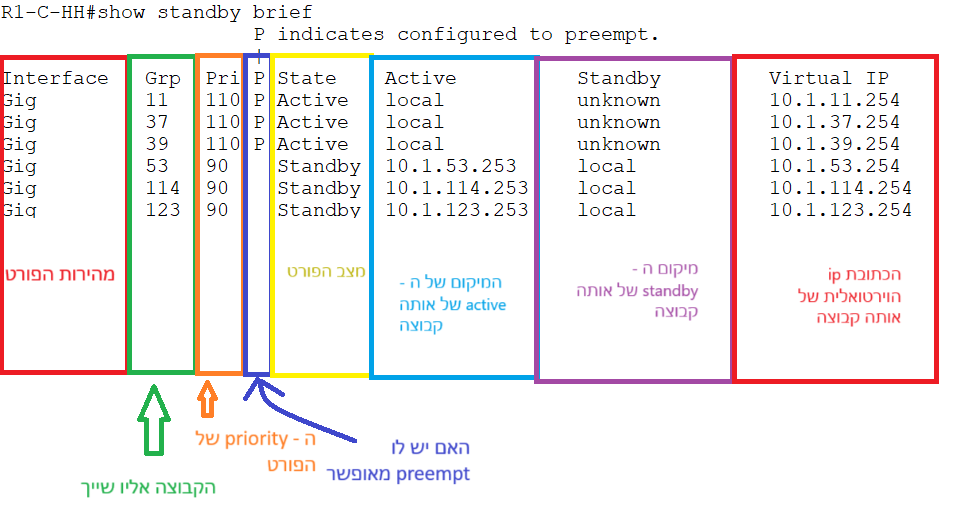
****

## פקודות show

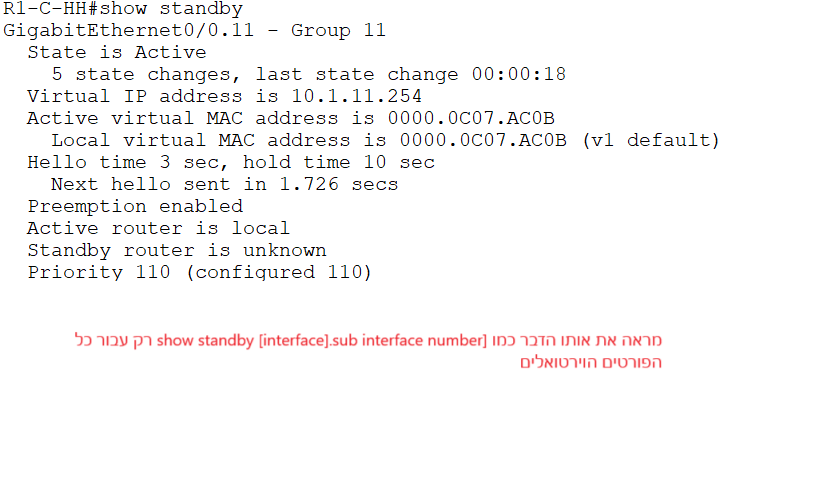
**Show standby [interface].[subinterface number]**

****

**Show standby brief**

****

**Show standby**

****

# Hsrp

## הסבר

פרוטוקול hsrp נותן ליצור נתב גיבוי למקרה שהנתב המרכזי נופל

נתבים מקבלים priority וקביעת הנתב המרכזי ניקבעת לפי הנתב עם ה – priority ההכי גבוה

בגלל שהגיבוי מחולק לפי רשתות vlan אז ניתן גם לעשות load balancing בכך שנתב אחד יהיה המרכזי לחצי מרשתות ה – vlan ויהיה משני לחצי השני של רשתות ה – vlan והפוך לגבי הנתב השני

הדרך בה רכיב ברשת הפרטית יודע לפנות לנתב הנכון היא שבפרוטוקול hsrp בנוסף לכתובת default gateway של כל נתב, ניתנת גם כתובת default gateway וירטואלית והרכיבים לא רואים את שני הנתבים רק כאשר הם שולחים הודעה הנתב המרכזי לאותו vlan ממנו באה ההודעה הוא זה שעונה לה

## Standby [group number] ip

**הסבר**

נשתמש בפעולה זו בשביל לקבוע את הכתובת ip הוירטואלית של אותו vlan

**תחביר**

Standby [group number] ip [ip address]

**תמונה של הפקודה**



## Standby [group number] priority

**הסבר**

נשתמש בפעולה זו בקונפיגורציה של פורט וירטואלי בשביל לקבוע את ה – priority של הראוטר עבור אותו ה – vlan

**תחביר**

Standby [groupe number] priority [priority number]

**תמונה של הפקודה**



## Standby [group number] preempt

**הסבר**

נשתמש בפקודה זו בקונפיגורציה של פורט וירטואלי על vlan שהנתב הוא מרכזי בשבילו בשביל להגדיר שאם הנתב המרכזי נופל ואז חוזר אז הוא אוטומטית יהפוך להיות במצב active עוד פעם

**תחביר**

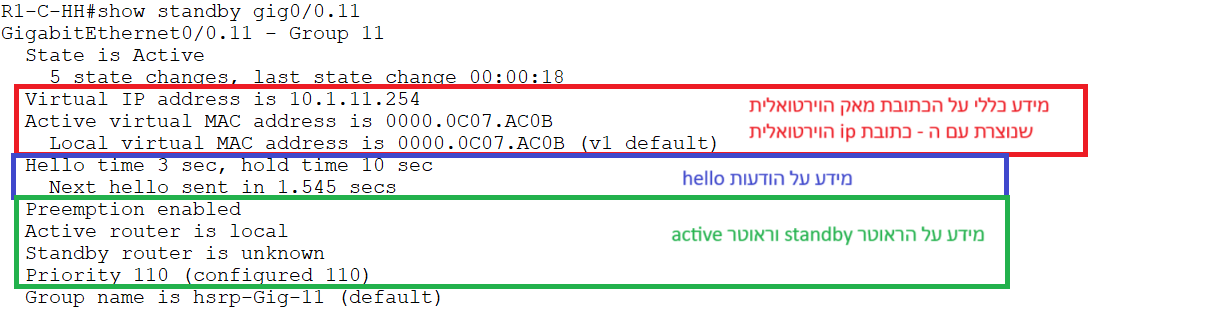
Standby [group number] preempt

**תמונה של הפקודה**

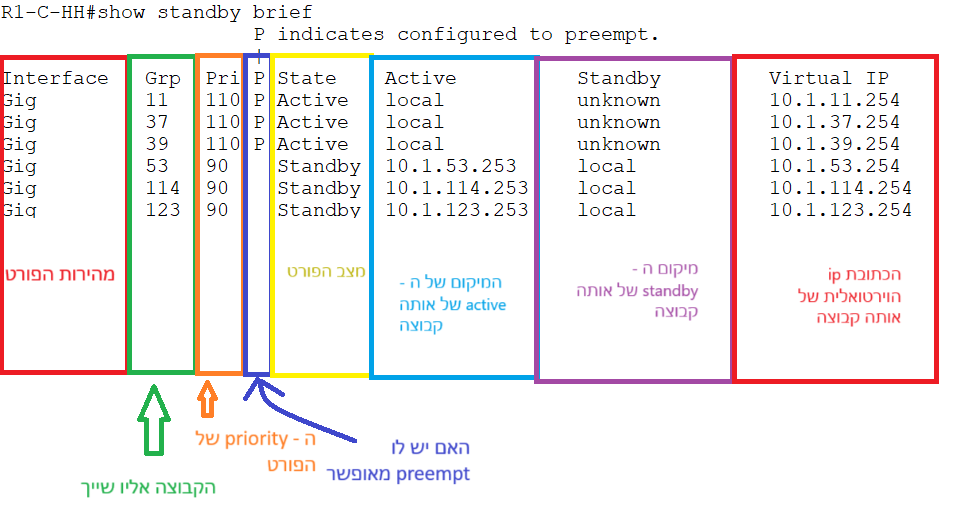


## פעולות show

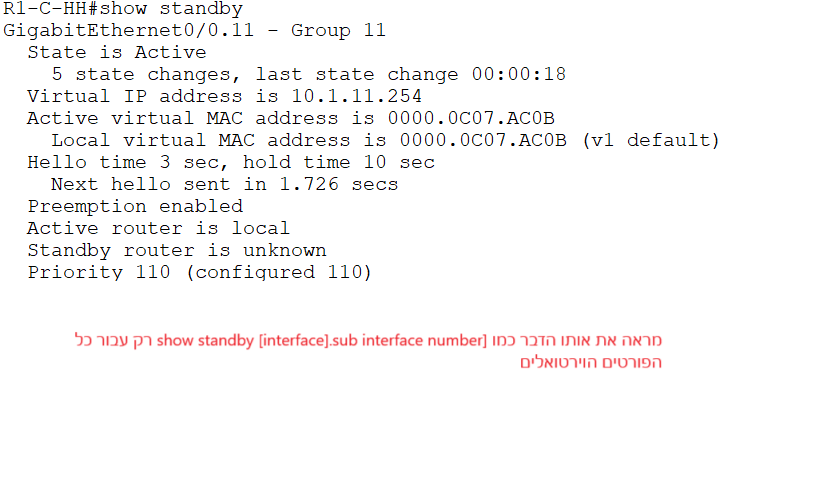
**Show standby [interface].[sub interface number]**

****

**Show standby brief**

****

**Show standby**

****

# DHCP - Dynamic Host Configuration Protocol

## הסבר

כיום ברשתות באינטרנט יש הרבה מחשבים בכל רשת וזה קשה ועבודה מיותרת לתת לכל מחשב ip משלו ידנית במיוחד למכשירים שזזים ולא נימצאים באותה רשת להרבה זמן כמו למשל טלפונים.

לכן יש את הפרוטוקול DHCP שאוטומטית נותן ip לכל מכשיר שמנסה להתחבר לרשת אינטרנט

הפרוטוקול DHCP עובד ב – 4 שלבים שראשי התיבות שלהם בוא DORA:

1. D – Discover: המחשב שואל האם יש שרת dhcp ברשת
2. O – Offer: השרת DHCP שנימצא ברשת מציע למחשב כתובת ip
3. R – Request: המחשב מאשר את הכתובת ip שהשרת DHCP הציע לו
4. A – Acknowledge: השרת DHCP מאשר שקיבל שהמחשב קיבל ולקח את הכתובת שהוא הציע לו

דבר חשוב לזכור מכיוון שלשרת DHCP יש כמות מוגבלת של כתובות ip ולכן הוא רק מלווה כתובות ip לזמן מסויים ולאחר זמן זה הכתובת ip חוזרות למאגר כתובות ip של השרת DHCP

## Ip dhcp pool

**הסבר**

נשתמש בפעולה זו בשביל להתחיל את הרשימה של כתובות שהשרת DHCP ייתן ומה הכתבות שהוא ייתן לשירותים אחרים כמו כתובת default gateway

**תחביר**

Ip dhcp pool [pool name]

**תמונה של התמונה**

****

## network

**הסבר**

נשתמש בפעולה זו בשביל להוסיף רשת ממנה הרשימה תחלק כתובות

**תחביר**

Network [network id] [sibnet mask]

**תמונה של הפעולה**



## Default router

**הסבר**

נשתמש בעולה זו בשביל להגדיר את הכתובת של ה – default gateway שהרכיב מקבל מרשימה זו

**תחביר**

Default-router [ip address]

**תמונה של הפקודה**

****

## ip dhcp excluded-address

**הסבר**

משתמשים בפעולה זו בשביל להגדיר כתובות שהשרת DHCP לא יחלק דוגמה לכתובות כאלו היא הכתובות default gateway הפיזיות והוירטואליות

**תחביר**

ip dhcp excluded-address [start ip of excluded addresses] [end ip of excluded addresses]

**תמונה של הפקודה**

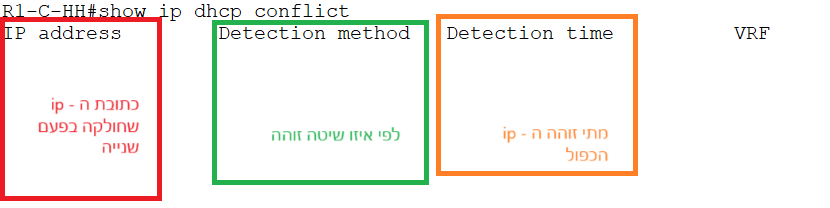
****

## פעולות show

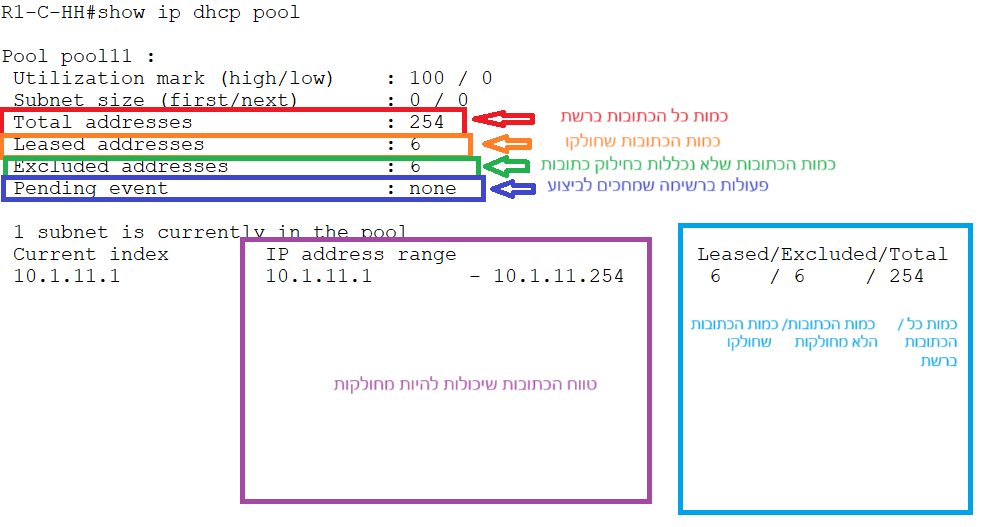
**Show ip dhcp binding**



**Show ip dhcp conflict**

****

**show ip dhcp pool**

****

# DNS – (Domain Naming Server)

## מה הסרבר עושה

המטרה של סרבר dns היא להתאים שמות לכתובות ip השייכים לאתרים שונים

## למה צריך

בנאדם רגיל שמשתמש באינטרנט יכול לנסות לגשת למספר נירחב של אתרים וכבנאדם זה יהיה מאוד קשה לזכור יותר מכמות קטנה של כתובות ip בלי עוד לחשוב על כך שהוא צריך לזכור למה משוייכת כל כתובת ip.

פה בא התפקיד של השרת dns, השרת dns משייך כתובות ip לשמות של אתרים כך שיותר קל למשתמש לזכור שמות של אתרים ומה האתרים

## איך עובד

קיימים 3 סוגים של שרתי dns:

**שרת root dns:** תפקידו להחזיק את כל השמות המרכזיים מהם פונים לשמות אחרים הניגזרים מהם (.com, .net, .app) ולהפנות את הבקשה לשרת המתאים של אותו שם, את הכתובות של השרתים האלה כל המחשבים יודעים ואלו שרתים מאוד חזקים כי הם מנהלים הרבה תעבורה.

**שרת top level dns:** זהו השרת dns שבו נמצאים כל השמות הנגזרים מאותו שם (למשל: site.com עבור השרת של .com) ותפקידו להפנות הלאה לשרת dns שבו נימצאות כל השיוכי ip לשמות מלאים של אתרים.

**שרת authoritative dns:** שרת מכיל שיוכי ip לשמות מלטאים ותפקידו בוא להמחזיר בחזרה את הכותבת ip עבור השם שביקש המשתמש

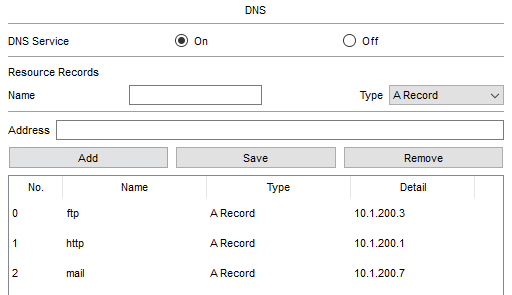
**דוגמה להליך קבלת כתובת ip עבור שם כשאר השם הוא site.com:**

1. מחשב שולך בקשה לקבלת ip עבור השם site.com לשרת root dns
2. השרת root רואה שהשם שייך לקבוצת השמות של .com ולכן הוא שולח את הבקשה לשרת שאחראי על שמות ה - .com
3. השרת top level dns מקבל את הבקשה ורואה שהבקשה היא לשם site ב - .com ולכן הוא שולך את הבקשה לשרת authoritative dns שאחראי על site ב - .com
4. השרת authoritative dns מקבל את הבקשה ורואה שיש לו כתובת ip משויכת לשם site.com והוא מחזיר את הכתובת ip כתשובה למבקש

## על איזה פורט עובד

הפרוטוקול dns עובד על פורט 53 ב – tcp ו - udp

## הגדרת השרת dns



# mail

## איך עובד

לשרת מייל יש שני צדדים שרת שמקבל מייל ושרת ששולח מייל שנימצא בשרת למשתמש, שניהם עובדים על פורט שונים.

**פרוטוקול שליחת מייל:** לו קוראים smtp והוא עובד על פורט ב - 25 ב - tcp.

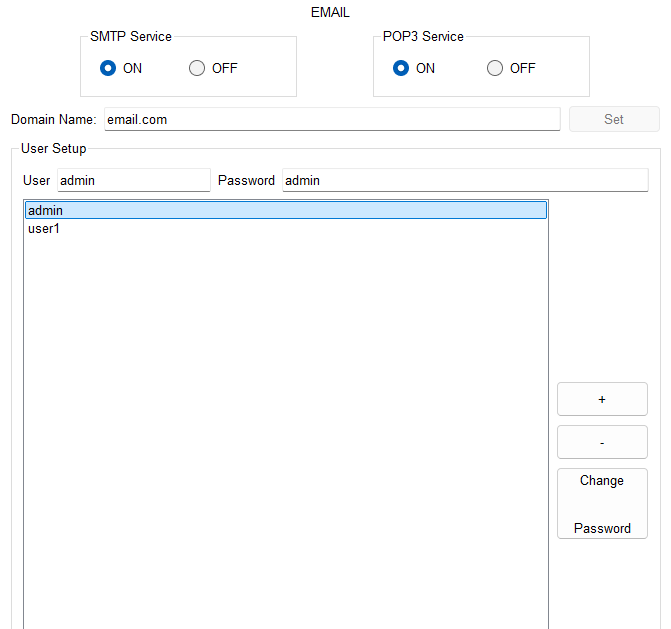
**פרוטוקול משיכת מייל:** לו קוראים pop3 והוא עובד על פורט 110 ב – tcp.

בשרת מייל קיימת תיבת דואר עבור כל משתמש ולכל domain יש רשימה ניפרדת של שמות משתמשים משלה.

ניתן לגשת למשתמש בכך שכותבת [username]@[DomainName]

דוגמה עבור דומיין site.com ומשתמש בשם user1: user1@site.com

## הגדרות



A screenshot of a computer

Description automatically generated

# http/https(Hyper Text Transfer Protocol/Secure) server

## הסבר על שרת web

שרת web הוא שרת שמארך אתרי אינטרנט ובעזרתו משתמשים יכולים להיתחבר לאתרי אינטרנט

**שרת web יכול לעבוד על אחד משני פרוטוקולים:**

1. http: עובד בפורט 80 ומעביר מידע בלי הצפנה
2. https: עובד על פורט 443 ומעביר מידע בהצפנה

## הגדרות

A screenshot of a computer

Description automatically generated

# FTP(File Transfer Protocol) server

## הסבר

שרת ftp הוא שרת הנותן לאחסן ולהוריד קבצים

הוא עובד על שני פורטים ב - tcp

פורט 21: פורט control המטרה שלו היא לנהל את השיחה בין השרת ftp למישתמש

פורט 20: פורט data משומש להעברת מידע

## הגדרות

A screenshot of a computer

Description automatically generated

# TFTP(Trivial File Transfer Protocol) server

## הסבר

עובד על פורט 69 זה פרוטוקול בסיסי יותר מ – ftp ולכן הוא עובד רק עם פורט אחד

הוא עובד ב – udp ולכן הוא משומש רק ברשת המקומית

## הגדרות

**העתקת running config לשרת tftp**

A screen shot of a computer

Description automatically generated

**העתקת הקובץ קונפיגורציה מהשרת tftp ל – running config**

A computer screen shot of a computer code

Description automatically generated

# Syslog server

## הסבר

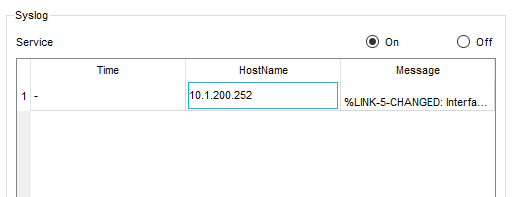
זה סרבר לתיעוד אירועים שקרו ברכיבים ברשת, לא רק שגיעות.

## הגדרות

הגדרת שימוש בסרבר סיסלוג



## תמונה שעובד



# NTP server

## הסבר

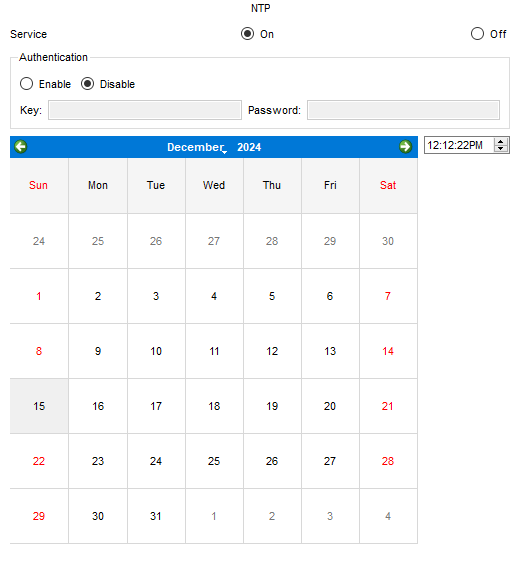
זה הוא שרת המשומש לסינכרון השעה בין כל הרכיבים ברשת

## הגדרות

**הגדרה מה הסרבר ntp בראוטר**

****

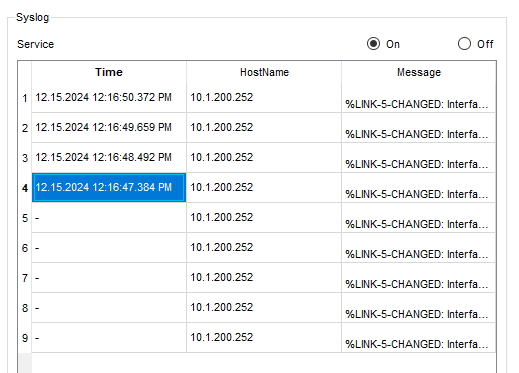
**הגדרת שרת ה – ntp**

****

**הוספת הזמן של השרת ntp להודעות syslog**

****

**תמונה שעובד**

****

# סניף תל אביב

# מוסכמות השמות

|  |  |  |  |
| --- | --- | --- | --- |
| שם הרכיב | שיכבה בטופולוגיה | הסבר השם המקוצר | הסבר על השם בקיצור |
| PC |  | pc<n>-<bn>-<dpt> | N – pc number  Bn - branche name(HH-Hod Hasharon, TL – Tel Aviv, CY – Cyprus)  Dpt – department name |
| Switch 2960 | Access | SW<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Switch pt empty | dmz | SW<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Multi layer 3650 | Distribution | ML<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Router 1941 | Core | R<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Server | dmz | Server<n>-<sn>-<bn> | N – number of the derver  Sn – server name (for example DHCP  Bn – branche name |

# טבלת השוואה מתג שיכבה מתג שיכבה 3 שיכבה 2

|  |  |  |
| --- | --- | --- |
| **קריטריון השוואה** | **Switch layer 2** | **Switch layer 3** |
| שיכבה | Layer 2 (data link) | Layer 3 (network) |
| כתובות בשימוש | Mac address | Ip address |
| מניעת לולאות | על ידי שימוש בפרוטוקול - stp | שימוש בניתוב ip ו – TTL (Time to Live) בשביל לזרוק פאקטות שניכנסו ללולאה |
| מעבר בין רשתות vlan | חייב לעלות עד לראוטר בשביל לעבור לרשת חדשה | יכול להעביר לרשת vlan השנייה מבלי לעלות לראוטר ובכך מוריד עומסים ברשת |
| עלות | יחסית זול יותר כי טכנולוגיה זו קיימת יותר זמן והיא יותר פשוטה כי לא משלבת שת שכבות | יחסית יקר יותר מכיוון שמתעסק בטכנולוגיה מתקדמת יותר המשלבת שני שכבות |
| סיבות לבחירה | קל יותר לתפעול, תקציב קטן יותר, רשת קטנה יחסית | קיים צורך לגמישות העברת תעבורה, צורך בהבטחה מרובה, רשת גדולה ומסובכת יותר |
| אבטחה | אין פילטור של מידע המועבר | קיים פילטור של מידע המועבר דרך רשימות גישה (access list) |

# תמונה של טופולוגיה בתולית

A computer network diagram with multiple colored circles

Description automatically generated

# הגדרות שכבר הופיעו

## Multilayer switch

**show running config**

**A screenshot of a computer program

Description automatically generated**

**Show vlan brief**

**A screenshot of a computer program

Description automatically generated**

**Left access layer switch**

**Show running config**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Show vlan brief**

**A screenshot of a computer

Description automatically generated**

# SVI – Switch Virtual Interface

## הסבר

פרוטוקול זה בדומה לפרוטוקול dot1q נותן לשלוח הודעות בין רשתות vlan שנות

הוא נותן לעשות זאת בכך שהוא יוצר ממשק וירטואלי לכל vlan כך שבניגוד ל – dot1q הממשק הזה לא משויך לפורט ספציפי אלה הוא משויך ל – vlan ספציפי והוא מקבל כתובת default gateway כך שהודעות שמגיעות מפורטים המשוייכים לאותו vlan עם כתובת default gateway המתאים לאותו ממשק מועברות דרכו אל ממשק vlan שונה ובכך יוצאות מפורט המשויך ל – vlan האחר.

## Interface vlan [vlan number]

**הסבר**

פקודה זו נותנת ליצור את הממשק הוירטואלי לאותו vlan

**תחביר**

Interface vlan [vlan number]

**תמונה של הפקודה**



## Ip routing

**הסבר**

פקודה זו נותנת למתג לנתב חבילות לפי הכתובות ip

**תחביר**

Ip routing

**תמונה של הפקודה**



# פקודה no switchport

**הסבר**

הפקודה הזו גורמת לפורט שהיא מופעלת עליו להיתנהג כמו פורט של רואטר כלומר כל תעבור שמגיעה מפורט זה יכולה להיות מנותבת לפי כתובות ip ללא ממשק וירטואלי כמו svi

**תחביר**

no switchport

**תמונה של הפקודה**

****

# פקודה ip helper-address

**הסבר**

הפקודה ip helper אומרת לראוטר במקום לתפקד כשרת dhcp, לשלוח הודעות המיועדות לשרת dhcp אל הכתובת המצוינת בפקודה

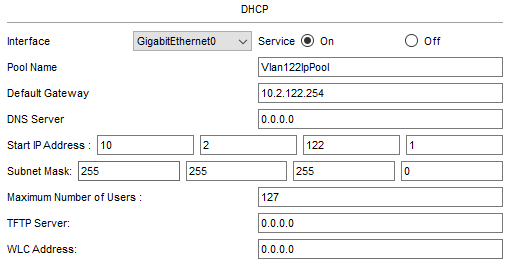
**תחביר**

Ip helper [ip address]

**תמונה של הפקודה**

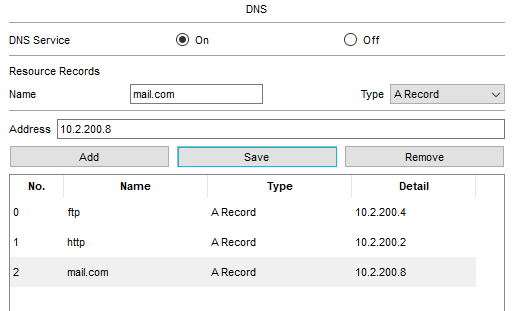


# הגדרות הסרבר dhcp

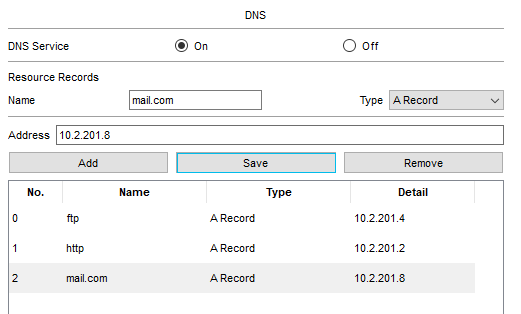


# שרת dns

## Dmz-1

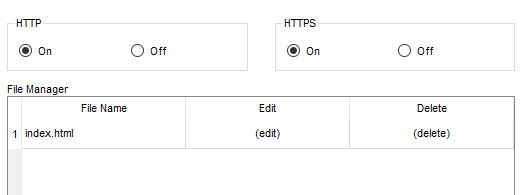


## Dmz-2



# שרת http

## הגדרת השרת http

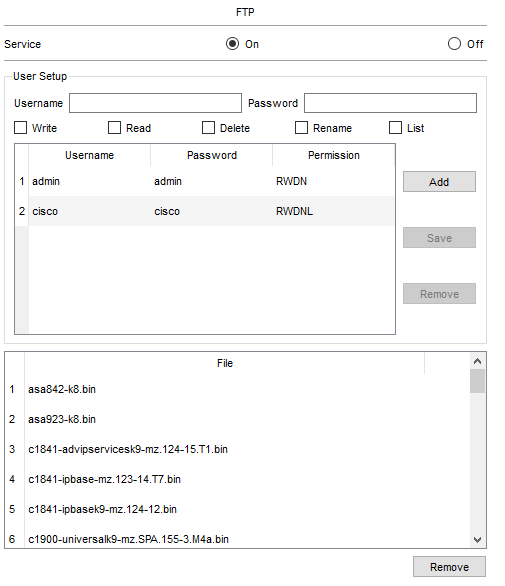


## בדיקה שהשרת עובד

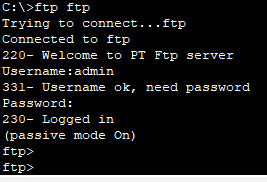


# שרת ftp

## הגדרת השרת

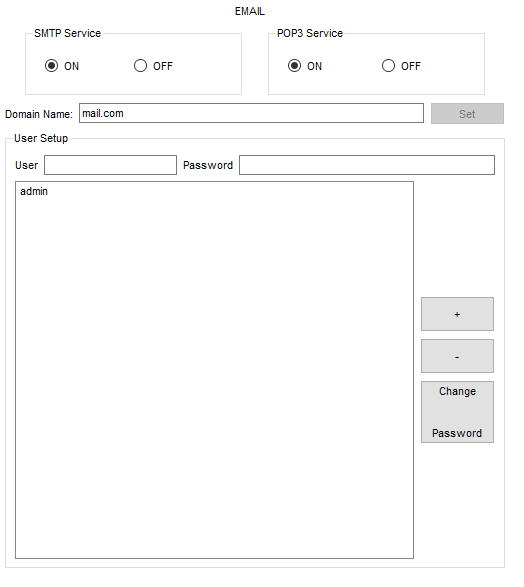


## בדיקה שעובד

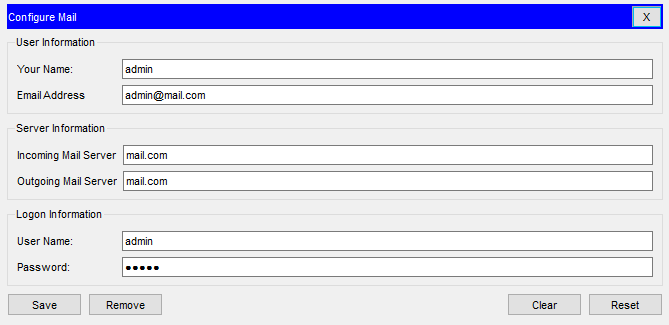


# שרת mail

## הגדרת השרת

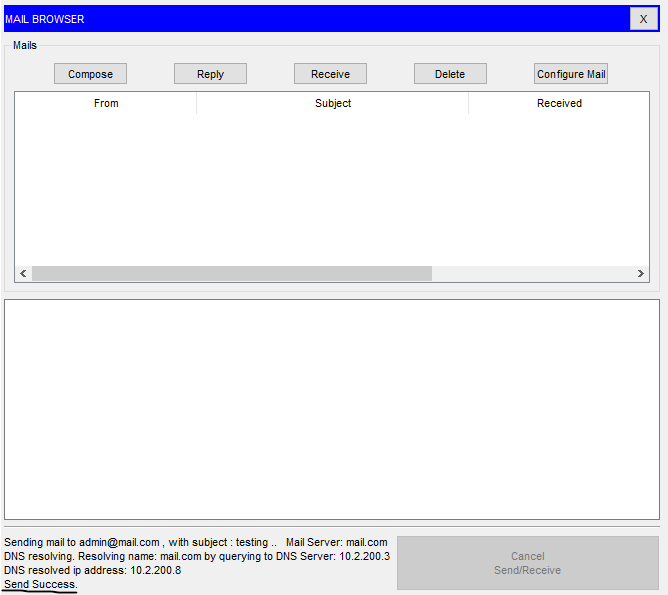


## הגדרת משתמש

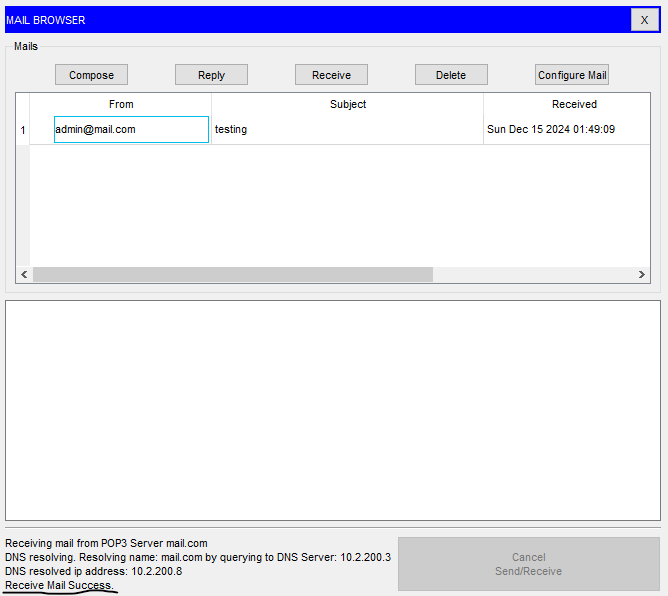


## בדיקה שעובד

**בדיקת שליחת מייל**

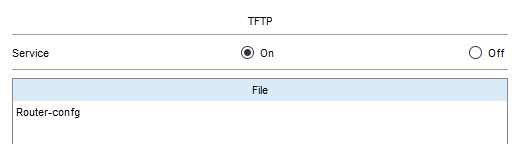


**בדיקת משיכת מייל**



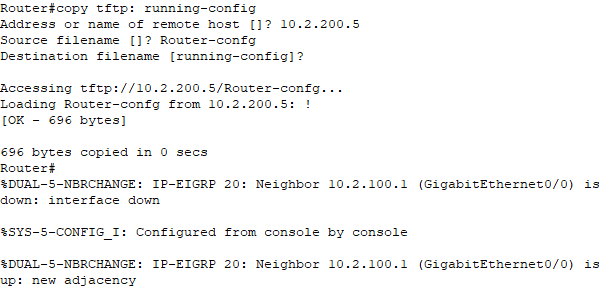
# שרת tftp

## הגדרת שרת

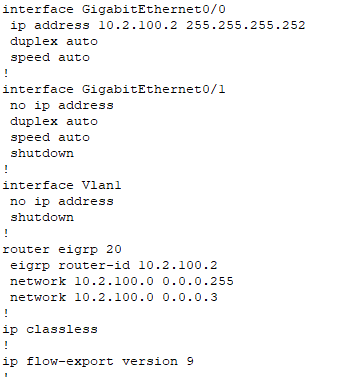


## בדיקה שעובד

**פעולת משיכת running config**

****

**תמונה של show run**

****

# שרת syslog

## הגדרת השרת

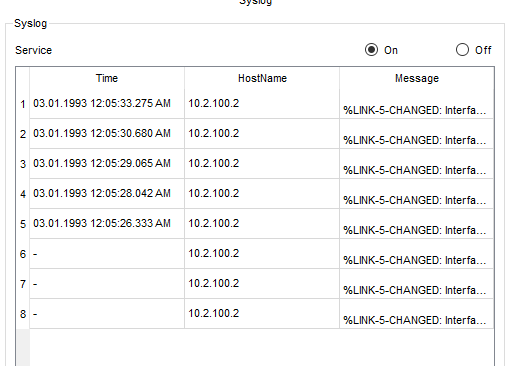


# שרת ntp

## הגדרת השרת



## רשומות syslog עם זמן מהשרת ntp

****

# טבלת השוואת פרוטוקולי ניתוב

|  |  |  |
| --- | --- | --- |
| **מאפיין השוואה** | **Distance vector** | **Link state** |
| פרוטוקולים בסוג זה | Rip, eigrp | ospf |
| ידע על טופולוגיה | מכיא רק שכנים | מכיר את כל הרואטרים בטופולוגיה |
| זמן היתכנסות (זמן עליה מחדש לנתיב חלופי ברגע שראוטר בנתיב המקורי נפל) | נמוך כי מכיר רק את השכנים וצריך להריץ את האלגוריתם מחדש עבור הנתיב על כל הראוטרים | גבוהה כי מכיר את כל הטופולוגיה ויכול לעשות את החישוב לבד |
| אלגוריתמי מציאת נתיב | Rip- bellman ford  Eigrp-dual | Ospf-dijkstra |
| עומס על רכיב | נמוך כי רוב החישוב של האלגוריתם נעשה בין נתבים | גבוה כי רוב החישוב של האלגוריתם נעשה בתוך הנתב |

# סיכום eigrp

## הסבר פרוטוקול eigrp

**מהו פרוטוקול eigrp**

פרוטוקול eigrp הוא פרוטוקול distance vector שמשתמש ב – asים כשהוא יוצר ניתוב עבור כל as

**איך בוחר נתיבים**

ראוטר שעובד בפרוטוקול eigrp מקבל מכל ראוטר המחובר אליו את הניתוב הכי טוב שלו עבור כל רשת. לאותו נתיב יש מחיר RD(Reported Distance) והנתב מוסיף ל - RD את המחיר בינו לבין השכן ממנו קיבל את הנתיב וזה יוצר FD(Feasible Distance) והנתב בוחר את הניתוב הכי טוב לאותה רשת לפי ה – FD הנמוך ביותר

**חישוב נתיבים**

הפרוטוקול eigrp מחשב נתיבים עבור כל רשת כאשר הנתיב עם ה – FD הנמוך ביותר עבור אותה רשת הוא זה שניבחר והנתיב עם ה – FD השני הכי נמוך הוא ניקבע להיות הנתיב המישני במקרא בו נפל הנתיב הראשי.

בדרך זו כאשר נתיב הופך להיות ללא זמין ניתן יהיה לגשת לרשת הרבה יותר במהירות כי יש נתיבי גיבוי.

הנתיב הראשי ניקרא successor והנתיב המשני ניקרא feasible successor.

כדי למנוע לולאות ל - feasible successor יש את התנאי FD > RD.

כאשר נתיב לרשת הוא לא זמין, כלומר הוא נפל או עדיין מחושב הוא במצב active וכאשר הנתיב זמין הוא במצב passive.

**חישוב cost**

חישוב ה – metric נעשה על ידי שימוש ב – 5 ערכים שונים שניקראים חמשת ערכי ה – k. העכריעם האלו הם:

1. K1 = מייצג את האורך קו המינימאלי בחיבור
2. K2 = מייצג את העומס על החיבור
3. K3 = מייצג את הדיליי על אותו חיבור
4. K4 = מייצג את האמינות של החיבור
5. K5 = מייצג MTU כלומר גודל הפאקטה המקסימאלי שיכול לעבור בחיבור

בחישוב ה – cost לא כל ה – k חייבים להיות משומשים ורק k1 ו – k3 משומשים אוטומטית

עבור חישוב ה – cost יש נוסחה שמחושבת לבד בראוטר שקוראים לה dual והנוסחה היא:



**סוגי הודעות**

1. Hello – זו הודעה הנישלחת כדי לדעת שיש ראוטר בצד השני ולשמור על יחסי שכנות, הוא דיפולטיבית 5 שניות וה – hold time (זמן שהראוטר מחכה להודעה עד סיום שכנות) הוא כפול 3 מה – hello time. הודעה זו נישלחת ב – multicast בכתובת 224.0.0.10
2. Update – הודעה זו נישלחת ביצירת שכנות כדי להעביר את הניתובים בין הרואטרים ולאחר מכן הם נישלחים רק בעידכוני ניתוב
3. Query – הודעה שנישלחת כשנתיב נופל למציאת נתיב חלופי
4. Ack(acknowledge) – הודעה שנישלחת לאישור קבלה של הודעה כדי להפוך את ההודעה לאמינה חלק מההודעות שמקבלים עליהם הודעת ack הם: update, querry, reply
5. Reply – זו הודעת תגובה על הודעת query שנותנת מידע על הנתיב שביקשו

**יצירת שכנות**

יש כמה תנאים שחייבים לחול בשביל שיחסי שכנות יוכלו להתקיים:

1. שני הראוטרים עובדים בפרוטוקול eigrp
2. הראוטרים נימצאים באותה as
3. אותם ערכי k משומשים בשני הראוטרים
4. אם יש שימוש ב – authentication אז הוא חייב להיות שווה בשני הראוטרים
5. שני האינטרפייסים על החיבור חייבים לרהיות על אותה רשת

שכנות ב – eigrp נוצרת בקלות יחסית מכיוון שאם תנאים אלו קיימים נוצרת שכנות והשכנים מעבירים בין אחד לשני הודעות update כדי לקבל את הניתובים של הרשת ושכנות נוצרת

## פקודתrouter eigrp

**הסבר**

משתמשים בפקודה זו בשביל להיכנס להגדרות של הפרוטוקול eigrp באותו ראוטר

**תחביר**

Router eigrp [as number]

**תמונה של הפקודה**



## פקודת eigrp router-id

**הסבר**

פקודה זו קובעת מה יהיה ה – router id

**תחביר**

Eigrp router-id [router id]

**תמונה של הפקודה**



## פקודת network

**הסבר**

משומשת בשביל לפרסם רשתות דרך פרוטוקול eigrp

**תחביר**

Network [net id] [wildcard]

**תמונה של הפקודה**



## פקודת passive-interface

**הסבר**

פקודה זו קובעת אינטרפייס שיהיה פסיבי כלומר שהוא לא יקהל או ישלח הודעות hello ויהיה בילתי ניראה מבחינת ראוטרים אחרים

**תחביר**

Passive-interface [vlan/FastEthernet/GigabitEthernet] [interface number]

**תמונה של הפקודה**



## פקודת no auto-summary

**הסבר**

פקודה זו אומרת לראוטר לא להישתמש בסמריזציה אוטומטית של רשתות

**תחביר**

No auto-summary

**תמונה של הפקודה**

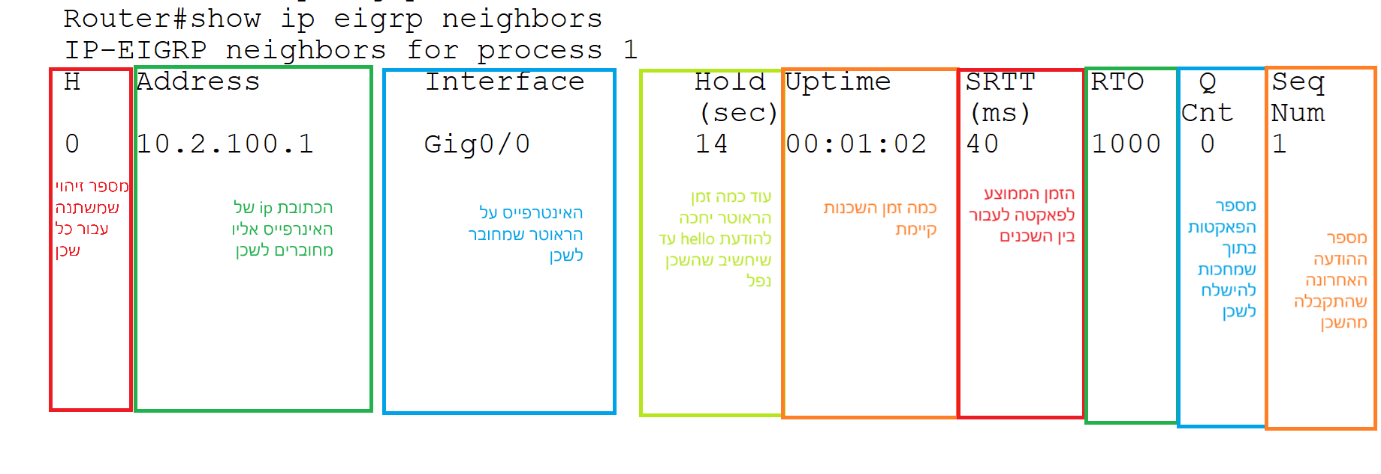


## פקודות show של eigrp

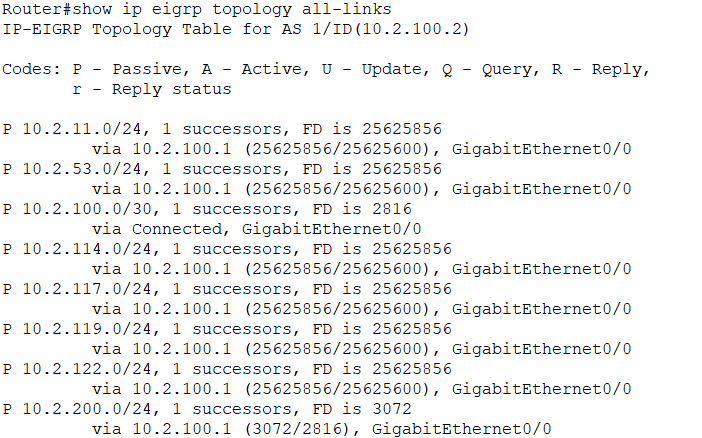
**פקודת show ip eigrp interfaces**

****

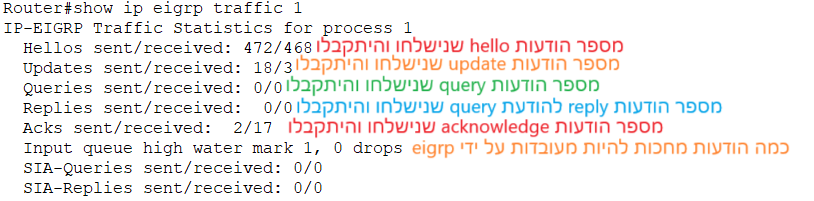
**פקודת show ip eigrp neighbors**

****

**פקודת show ip eigrp topology**

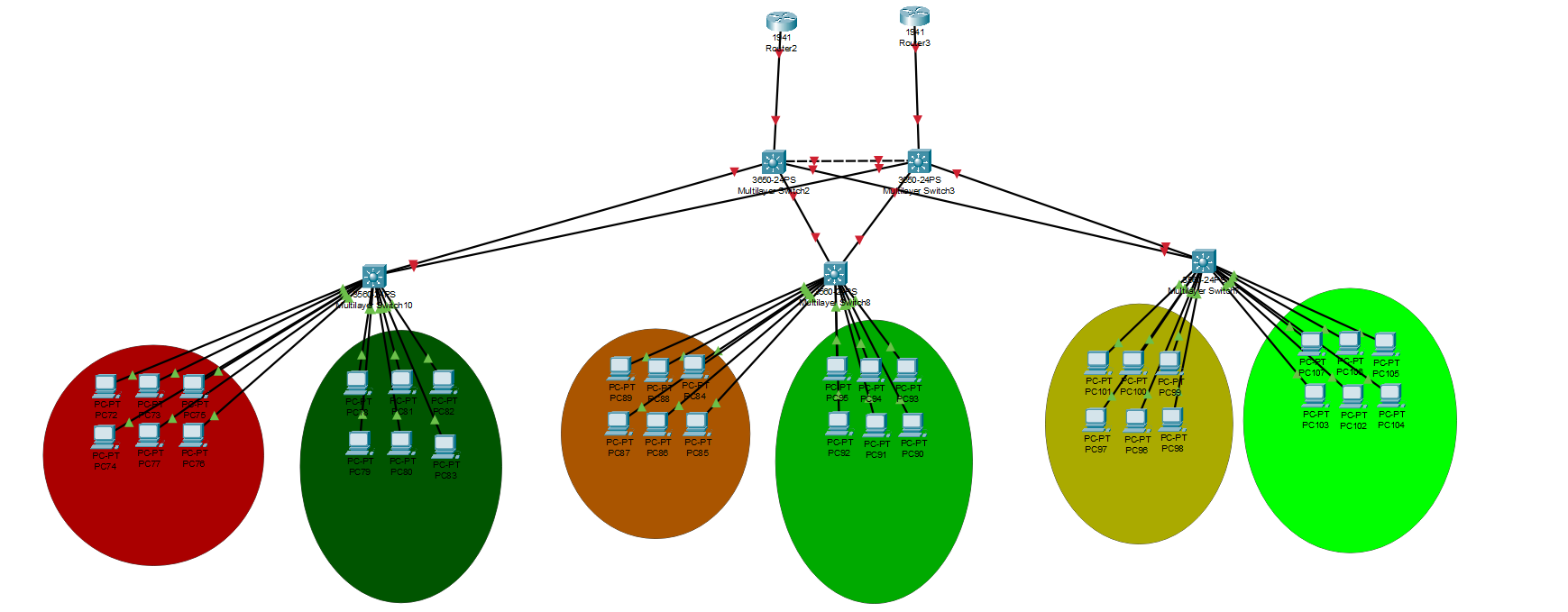
****

**פקודת show ip eigrp traffic**

****

# סניף קפריסין לימאסול

# טופולוגיה בתולית



# מוסכמות השמות

|  |  |  |  |
| --- | --- | --- | --- |
| שם הרכיב | שיכבה בטופולוגיה | הסבר השם המקוצר | הסבר על השם בקיצור |
| PC |  | pc<n>-<bn>-<dpt> | N – pc number  Bn - branche name(HH-Hod Hasharon, TL – Tel Aviv, CY – Cyprus)  Dpt – department name |
| Multi layer 3560 | Distribution | ML<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Switch pt empty | dmz | SW<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Multi layer 3650 | Distribution | ML<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Router 1941 | Core | R<n>-<ln>-<bn> | N – switch number  Ln – layer name  Bn – branche name |
| Server | dmz | Server<n>-<sn>-<bn> | N – number of the derver  Sn – server name (for example DHCP  Bn – branche name |

# הגדרות בסיסיות

## Multi layer switch distribution layer

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

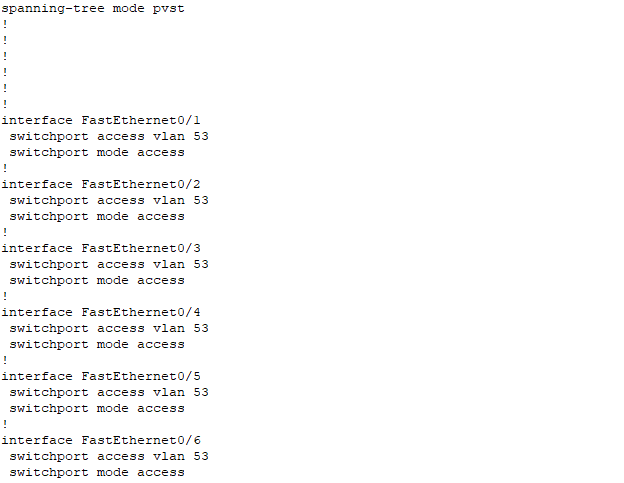
A white background with black dots

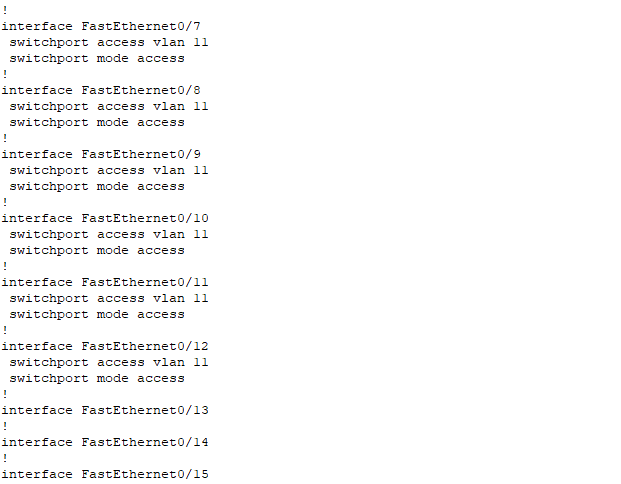
Description automatically generated

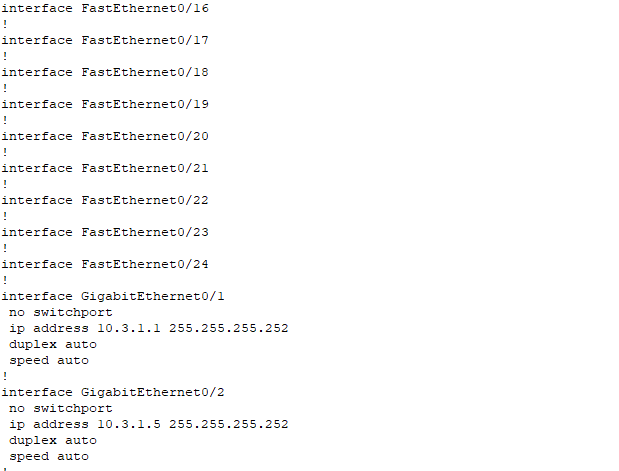
## Multi layer switch access layer

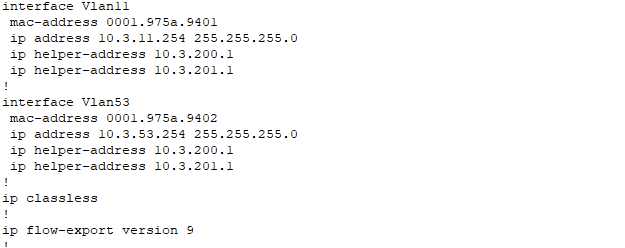
**Show run**





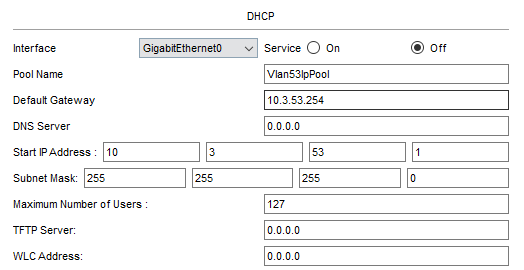






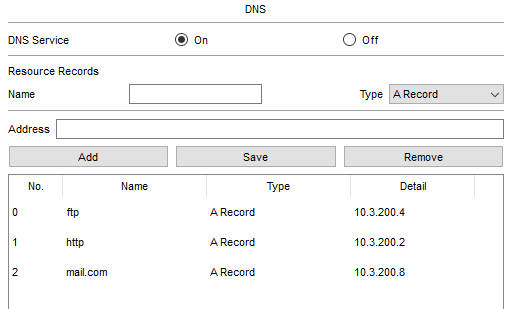


# הגדרות שרת ה - dhcp

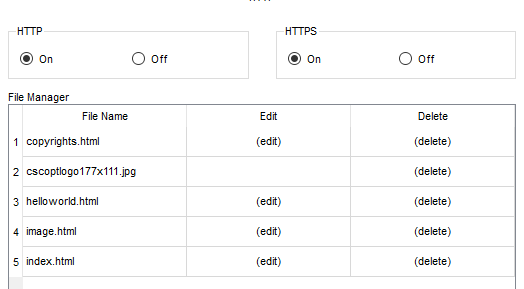


# שרת dns

## Dmz-1

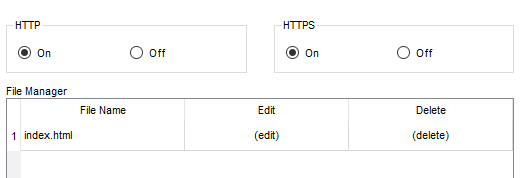


## Dmz-2



# שרת http

## הגדרות השרת

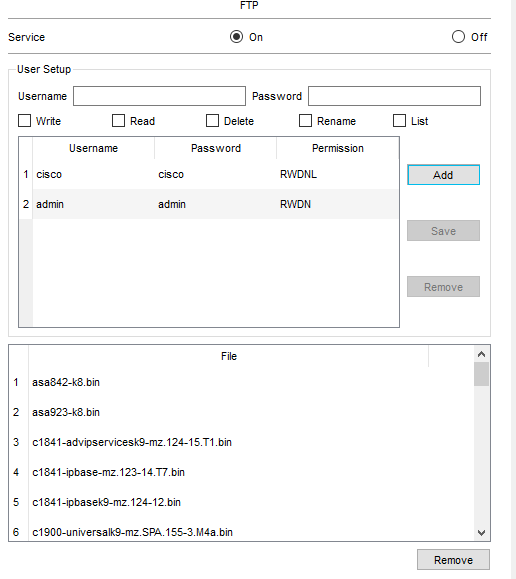


## תמונה שעובד

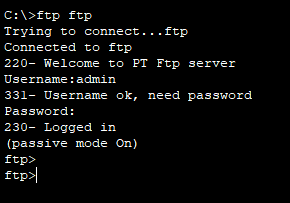


# שרת ftp

## הגדרת שרת

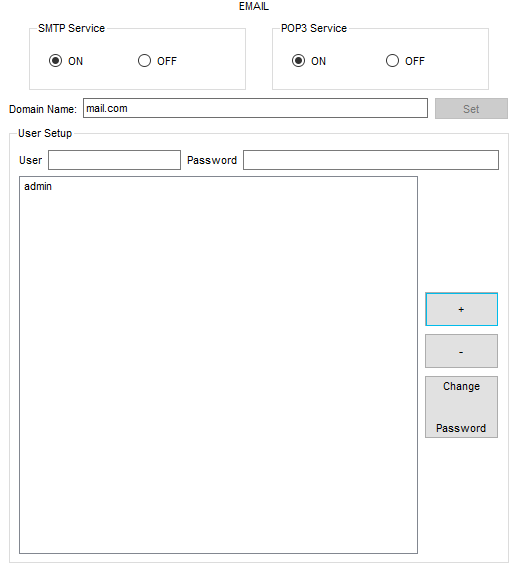


## תמונה שעובד

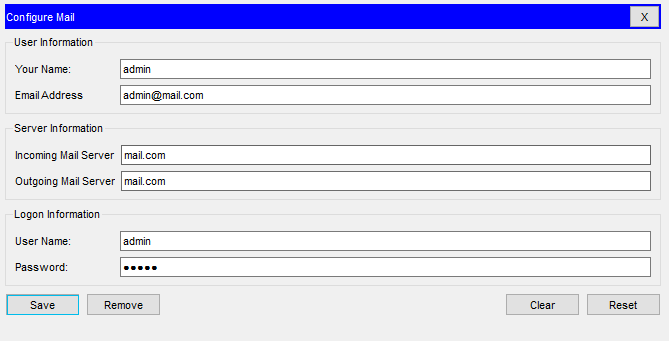


# שרת mail

## הגדרת השרת

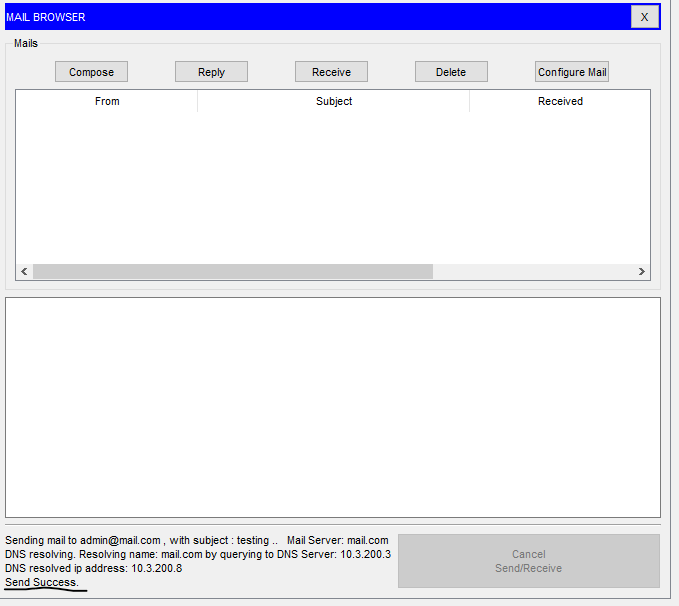


## הגדרת משתמש

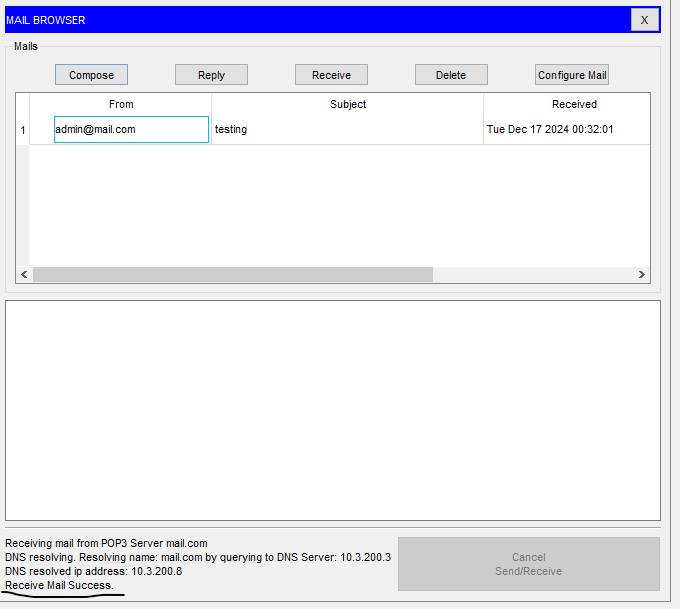


## תמונה שעובד

**תמונה של שליחת מייל**

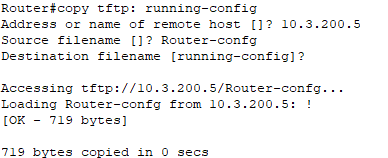


**תמונה של קבלת מייל**



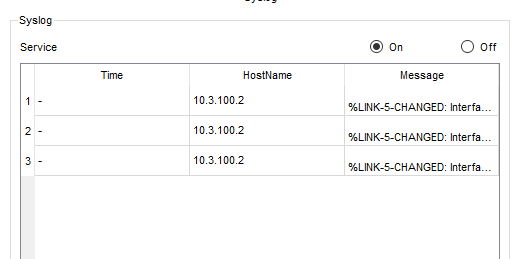
# שרת TFTP

## תמונה שעובד



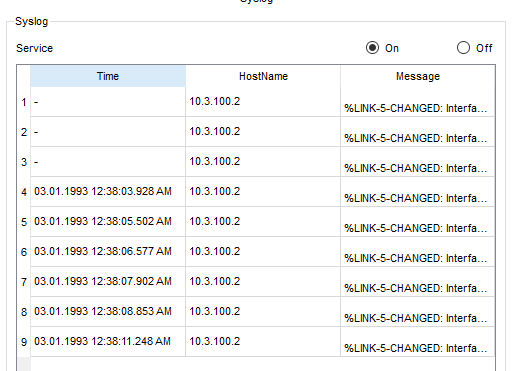
# שרת syslog

## תמונה של לוגים



# שרת NTP

## תמונה של לוגים עם הזמן של השרת ntp



# פרוטוקול ניתוב OSPF- Open Shortest Path First

פרוטוקול ospf הוא פרוטוקול ניתוב דינאמי. פרוטוקול זה מחשב את נתיבי הניתוב שלו לפי אלגוריתם שקוראים לו dijkastra

ה – AD של OSPF הוא 110

פרוטוקול OSPF עובד עם אזורים AREA כך שלכל אזור יש מספר אחר.

הדרך בה הפרוטוקול מונע לולאות ניתוב בין אזורים היא שכל האזורים חייבים להיות מחוברים לאזור 0.

פרוטוקול זה הוא ממשפחת link state, הוא מכיר את כל הטופולוגיה ומחשב את הנתיבים שלו מקומית. כאשר הטופולוגיה משתנה כל הנתבים מעודכנים על כך.

הפרוטוקול מתחזק ויוצר שכנות בעזרת הודעות hello

## Router Id

לכל ראוטר העובד בפרוטוקול OSPF יש ID ספציפי משלו המייחד בפרוטוקול

תהליך בחירת ה – router id היא כך:

* קביעה ידנית
* כתובת ה – loopback עם ה – ip ההכי גבוה
* אם לא מוגדרת כתובת loopback אז כתובת ה – IP ההכי גבוהה בראוטר

## סוגי routers ב – OSPF

**Internal router**

זה הוא ראוטר רגיל עם יחסי שכנות עם ראוטרים מאותו איזור

**Backbone router**

זה ראוטר של ospf שנימצא באיזור 0

**ABR – Area Border Router**

זה ראוטר שנימצא ביותר מאיזור אחד למשל איזור 0 ואיזור 1

**ASBR – Autonomous System Boundary Router**

זה ראוטר שמתפקד עם פרוטוקול ניתוב ospf ופרוטוקול ניתוב נוסף

**DR – Designated Router**

זה הוא ראוטר ברשתות ospf מסוג broadcast בהן ניבחר ראוטר אחד שינהל את העידכונים של ospf באותה רשת הראוטר מקשיב לעידכונים בכתובת 224.0.0.6

**BDR – Backup Designated Router**

זה ראוטר המקשיב לעדכונים ברשת כמו DR רק שהוא מתחיל לעדכן ולהיות אחראי על עידכונים ברגע שהנתב DR נפל

## בחירת DR ו – BDR

לכל ראוטר ב – ospf יש ערך priority הוא דיפולטיבית 1 ואם הוא 0 אז הנתב לא יכול להיבחר כ – DR או BDR. הנתב עם הערך priority הכי גבוה ייבחר להיות ה – DR ואחריו ה – BDR אם ערכי ה – priority שווים אז ייבחרו לפי ה – router ID

## רשתות שונות ב – OSPF

**Broadcast**

זו רשת בה כמה ראוטרים מחוברים אחד לשני בעזרת למשל switch כך שכל הנתבים המחוברים למתג הם באותה רשת, ברשת זו גם ייבחרו נתבי DR ו – BDR. ברשת זו עידכונים על שינויים ברשת יישלחו בכתובת 224.0.0.6 לראוטרים DR ו – BDR והראוטרים DR ו BDR יפרסמו את כל השינויים בכתובת 224.0.0.5

**P2P – Point To Point**

ברשת זו מחוברים שני נתבים ישירות. אין בחירת נתבי DR ו – BDR ברשת הזאת ועידכוני הרשת יישלחו בכתובת 224.0.0.5

## פרמטרים להקמת יחסי שכנות

**פרמטרים שחייבים להיות שווים**

* הראוטרים באיזור משותף
* חייבים להיות באותו סוג איזור
* ה – prefix של הרשת שהם נימצאים בה חייב להיות שווה
* הראוטרים חייבים להיות באותה רשת
* Hello time interval חייב להיות שווה
* Dead interval שווה
* אימות – אם משתמשי באימות הוא חייב להיות שווה

**פרמטרים שחייבים להיות שונים**

* ה – router id חייב להיות שונה
* הכתובת ip חייבת להיות שונה

## סוגי הודעות ב - ospf

**HELLO message**

זו הודעה שנישלחת דיפולטיבית כל 10 שניות. מטרתה היא לתחזק ולהקים שכנות. דיפולטיבית אם לא היתקבל משכן הודעת hello במשך 40 שניות (dead interval) אז השכן נחשב שנפל והשכנות מפסיקה

**DBD – DataBase Desciption**

הודעה המכילה את הסיכום של הבסיס נתונים של ospf בראוטר בשביל לראות האם הבסיסי נתונים של שני הראוטרים מעודכנים

**LSR – LinkState Request**

הודעה זו נשלחת אחרי הודעת DBD במקרה בו הנתן לא מסונכרן בשביל לבקש לקבל את המידע שחסר לו

**LSU – Link State Update**

הודעה זו נשלחת בתגובה להודעת LSU בשביל לשלוח את המידע המבוקש בהודעת LSR

**LSAck – Link State Acknowledgment**

זו הודעה הנשלחת בתגובה להודעת LSU בשביל לאשר את קבלת ההודעת LSU

**LSA – Link State Advertisement**

זו הודעה המכילה רק מידע על ניתובים

## סוגי הודעות LSA

**LSA 1 – Router LSA**

זו הודעה הנשלחת באותו איזור ניתובים על אותו ראוטר ששלח את ההודעה

**LSA 2 – Network LSA**

זו הודעה הנשלחת על ידי DR והיא מתארת את הרשת broadcast הודעה זו נישארת בתוך האיזור

**LSA 3 – Network Summary LSA**

זו הודעה הנשלחת על ידי ABR והיא מפרסמת רשתות מאיזורים אחרים

**LSA 4 – Summary ASBR LSA**

זו הודעה הנשלחת מ – ASBR והיא מפרסמת איך להגיע לואותו ראוטר ASBR

**LSA 5 – External LSA**

זו הודעה הנישלחת על ידי ראוטר ASBR והיא מפרסמת ניתובים לשתות מפרוטוקולי ניתוב אחרים מ- OSPF

**LSA 6 – Multicast LSA**

לא משומש כיום

**LSA 7 – NSSA LSA**

זו הודעה הנשלחת מראוטר ASBR באיזור NSSA, מכיוון שאיזור NSSA לא מקבל הודעות LSA 5 אז בשביל לפרסם באיזור NSSA ניתובים מפרוטוקולים אחרים מסווים את ההודעת ניתוב LSA 5 והופכים אותה להודעת פירסום ניתוב LSA 7

## סוגי אזורים ב – OSPF

**backbone area**

זה אזור מספר 0 ואליו חייבים להיות מחוברים כל האזורים האחרים התומך בהודעות LSA 1-5

**standard area**

זה הוא אזור רגיל של פרוטוקול OSPF התומך בהודעות LSA 1-5

**Stub area**

זה אזור שלא מקבל ניתובים מפרוטוקולי ניתוב שונים מ – OSPF הוא תומך בהודעות LSA 1-3

**Tottaly stubby area**

זה איזור שלא מקבל ניתובים מפרוטוקולי ניתוב שהם לא OSPF כמו איזור stub רק שהוא גם לא מקבל ניתובים מאיזורים אחרים. הראוטר ABR שלו יפרסם גם DSR שיכוון לראוטר ABR איזור זה תומך בהודעות LSA 1 – 2

**NSSA – Not So Stubby Area**

זה אזור שלא מקבל ניתובים מפרוטוקולים שונים מ – OSPF כמו איזור stub רק שהוא מקבל ניתובים מפרוטוקולים שונים המגיעים מראוטר ASBR הנימצאים באותו איזור הוא תומך בהודעות LSA 1-3,7

**TNSSA – Totally Not So Stuby Area**

זה איזור המקבל ניתובים של פרוטוקולים אחרים רק מנתבי ASBR הנימצאים באותו איזור כמו באיזור NSSA רק שבניגוד לאיזור מסוג זה הוא גם לא מקבל ניתובים מאזורים אחרים והנתב ABR של האזור מפרסם גם DSR שמכוון אליו בשביל רשתות שאין לאזור ניתוב אליהם אזור זה תומך בהודעות LSA 1-2,7

## Virtual link

כמו שהסברנו בהסבר על אזור backbone area כל האזורים חייבים להיות מחוברים לאזור זה. זה צורך המון משאבים כי לא בהכרח כל האזורים יהיו קרובים לאזור 0 מה שיידרוש הרבה כסף מכיוון שיהיה צורך בכבלים ארוכים.

לכן נשתמש ב – virtual link בשביל לעבור דרך אזורים אחרים כדי להגיע לאזור 0. נגדיר virtual link על נתבי ABR המחברים בין שני אזורים בשביל ליצור חיבור וירטואלי שיאפשר לשני האזורים המחוברים על ידי שני הנתבי ABR להיות מחוברים אחד לשני ישירות מבחינה לוגית וכך נחבר אזורים מרוחקים ישירות לאזור 0 מבחינה לוגית

## Router ospf

**הסבר**

נשתמש בפקודה זו בשביל להיכנס להגדרות של OSPF

**תחביר**

router ospf [process number]

**דוגמה**



## router-id

**הסבר**

נשתמש בפקודה זו בשביל להגדיר את ה – RID של הראוטר ב – OSPF

**תחביר**

Router-id [router id]

**דוגמה**



## Network

**הסבר**

נשתמש בפקודה זו בשביל לפרסם רשת ב – OSPF שמחוברת ישירות לראוטר

**תחביר**

Network [net id] [wildcard] area [area number]

**דוגמה**



## Virtual link

**הסבר**

נשתמש בפקודה זו בשביל ליצור virtual link

**תחביר**

Area [area virtual link goes over] virtual link [RID of router to make virtual link with]

**דוגמה**



## Area type

**הסבר**

נשתמש בפקודה זו בשביל להגדיר את סוג האזור

**תחביר**

Area [area number] [stub/nssa]

**דוגמה**



## Totally area

**הסבר**

נשתמש בפקודה זו כדי להגדיר שהאיזור יהיה גם מסוג totally

**תחביר**

Area [area number] [area type] no-summary

**דוגמה של הפקודה**



## Redistribute

**הסבר**

נשתמש בפקודה זו בשביל לפרסם ניתובים מפרוטוקול OSPF לפרוטוקול אחר

**תחביר**

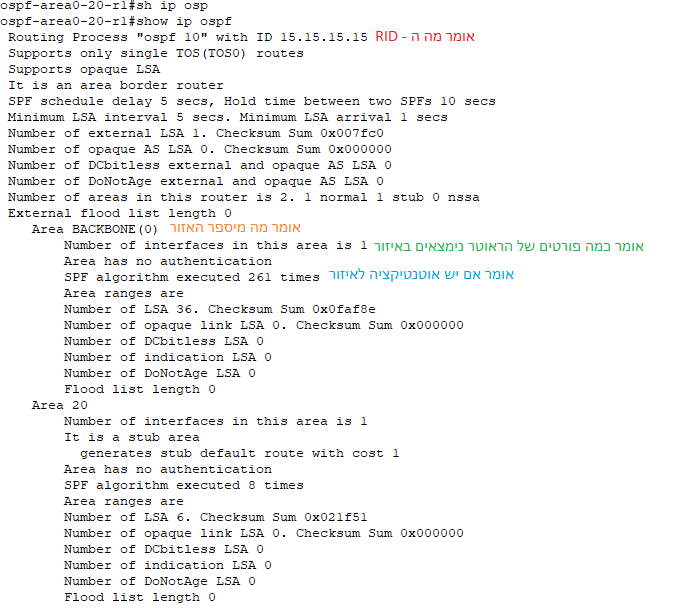
Redistribute ospf [process number]

**דוגמה של הפקודה**

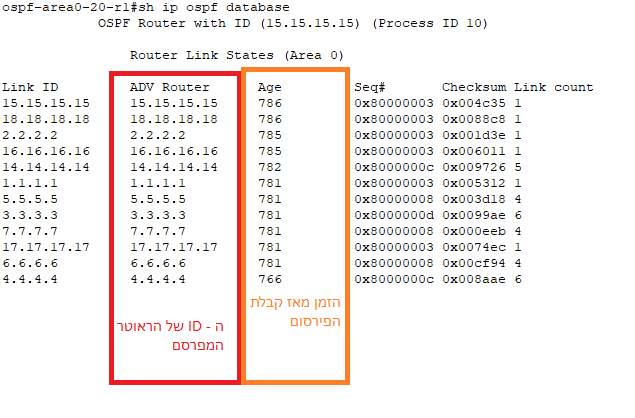


## פקודות show ב – OSPF

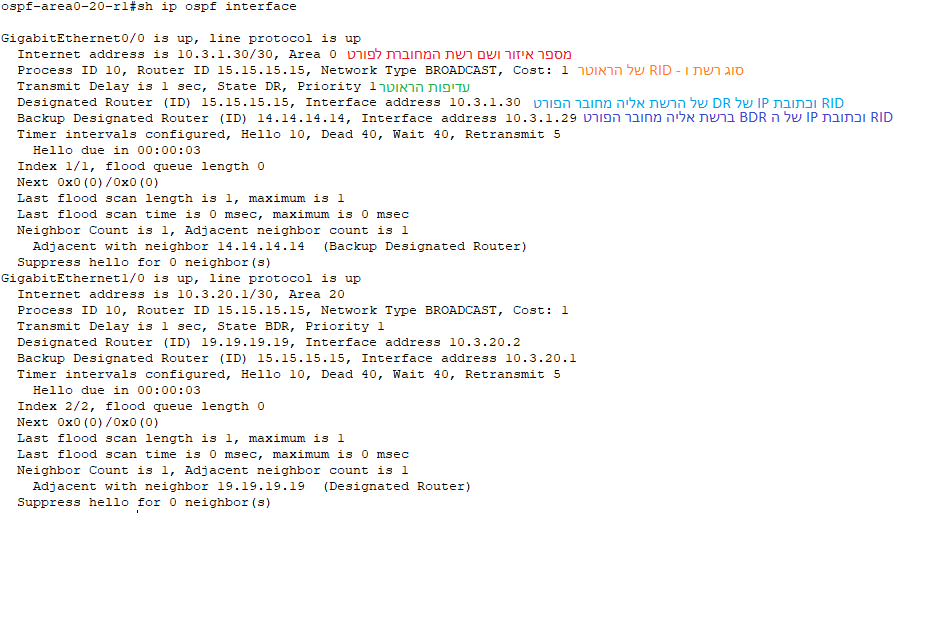
**Show ip ospf**

****

**Show ip ospf database**

****

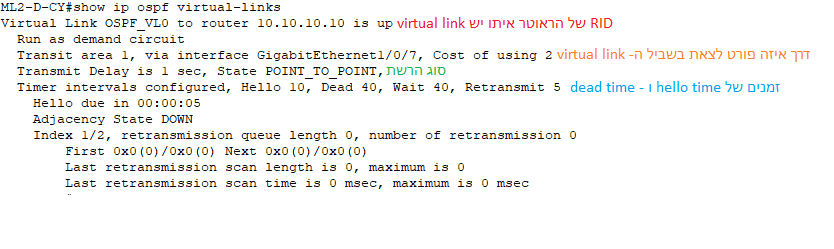
**Show ip ospf interface**

****

**Show ip ospf neighbor**

****

**Show ip ospf virtual-links**

****

# Static Route

## הסבר

ה – static route הוא נתיב אחד שמוגדר סטטית (כלומר הוא לא משתנה בהתאם לשינויים ברשת) על גבי הראוטר

ל – static route יש AD של 1

ה – static route בנוי כך שיש רשת שרצוי להגיע אליה וה – static route אומר מה הקפיצה הבאה שצריך לקחת בשביל להגיע לרשת הזאת, אפשר לעשות זאת בעזרת נתינת הכתובת ip של הקפיצה הבאה או לתת את הפורט שמוביל לקפיצה הבאה.

## Ip route

**הסבר**

נשתמש בפקודה זו בשביל להגדיר static route

**תחביר**

ip route [destination network] [destination SM] [next hop ip address/interface]

**תמונה של הגדרה**



## DSR – Default Static Route

**הסבר**

ה – DSR זה מקרה פרטי של static route בו מגדירים נתיב שלפיו המידע יישלח במקרה בו נתיב לרשת הרצויה לא קיים בטבלת ניתוב.

**הגדרה**

Ip route 0.0.0.0 0.0.0.0 [next hop ip address/interface]

**תמונה של הגדרה**



## שילוב עם פרוטוקולי ניתוב נוספים

**הסבר**

כדי ששאר הראוטרים ברשת יוכלו לדעת למי ללכת במקרה בו אין להם ניתוב בטבלת ניתוב לרשת הרצויה אז נפרסם את ה – DSR ל – ospf כדי ש – ospf יוכל לפרסם שאפשר להגיע לראוטר הזה למיקרים בהם אין להם ניתוב לרשת הרוציה בטבלת ניתוב

**הגדרה**

Default-information originate

**תמונה של הגדרה**



# ACL – Access Control List

## הסבר

בשביל לשלוט בתעבורה העוברת בראוטר נשתמש ב – ACL, ה – ACL נותן לנו לאפשר תעבורה או למנוע תעבורה לפי פרמטרים שונים וקיימים שני סוגי רשימות גישה.

1. standard ACL – זו רשימת גישה שמפלטרת את התעבורה לפי היעד בלבד (טווח מספר של standard ACL הוא 1 – 99)
2. Extended/named ACL – זו רשימת גישה הנותנת לפלטר תעבורה לפי מקור, יעד והפורט המשומש (טווח מספר של extended ACL הוא 100 – 199)

הצורה בה רשימת גישה עובדת היא כך: יש לה חוקים אחד אחרי השני וברה שתעבורה מתאימה לאחד מהחוקים אז התעבורה עוברת או נחסמת לפי מה שהחוק אמר.

## Ip access list

**הסבר**

נשתמש בפקודה זו בשביל ליצור רשימת גישה ולהיכנס אליה

**תחביר**

ip access-list [standard/extended] [number/word]

**תמונה של הגדרה**



## permit

**הסבר**

נשתמש בפקודה זו בשביל לאשר תעבורה

**תחביר**

Permit [protocol] [source ip] [source sm] [destination ip] [destination wildcard]

**דוגמה של הגדרה**



## Deny

**הסבר**

נשתמש בפקודה זו בשביל לחסום תעבורה

**תחביר**

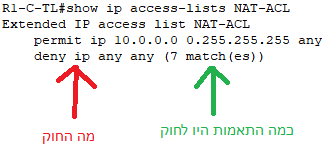
deny [protocol] [source ip] [source sm] [destination ip] [destination wildcard]

**דוגמה של הגדרה**



## פקודות show

**Show ip access-lists**



# BGP – Border Gateway Protocol

## הסבר

הסבר

פרוטוקול הוא פרוטוקול מסוג path vector המשמש לניתוב ברשת

יש שני סוגים של BGP:

1. IBGP(Interal BGP) – המשומש לניתוב בתוך ב – LAN כאשר ה – AD שלו הוא 200 מסיבה זו לא משתמשים ב – IBGP ברשת ה – LAN
2. EBGP(External BGP) – המשומש לניתוב ב – WAN כאשר ה – AD שלו הוא 20 לכן הוא משומש לרוב ב – WAN

**איך עובד**

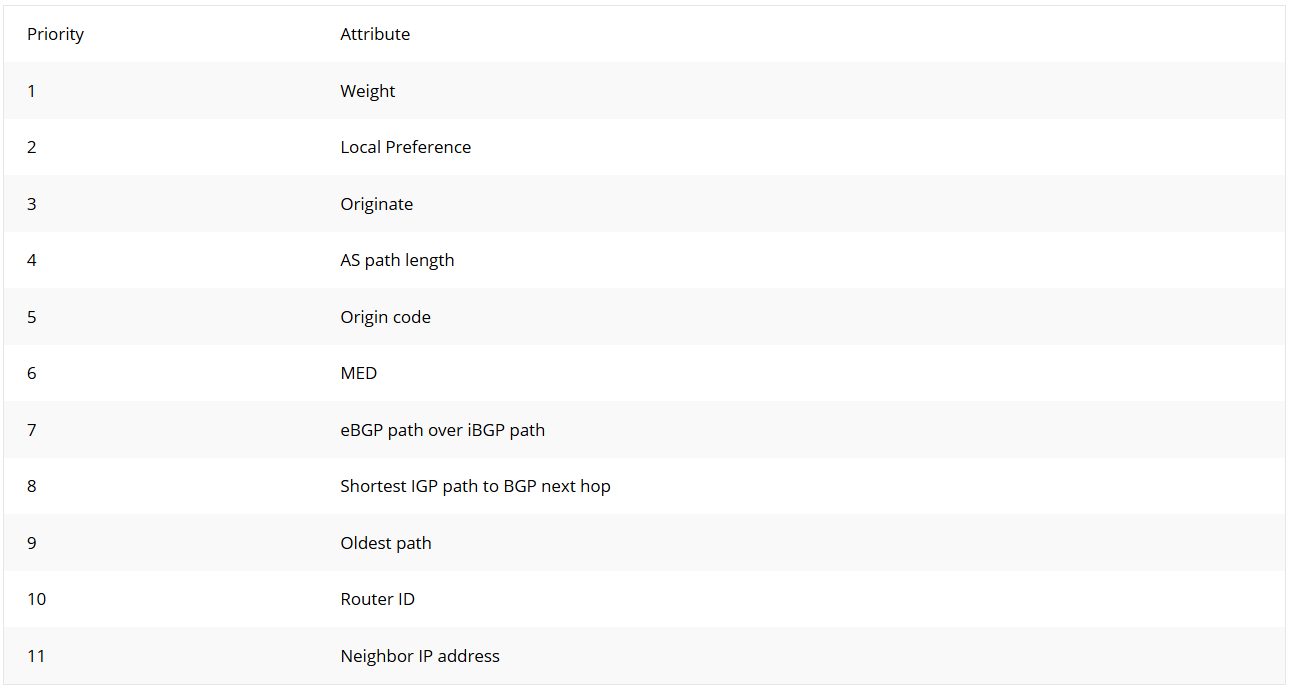
פרוטוקול BGP עובד בכך שהוא מחלק את הרשת לאיזורי – AS(Autonomous System) שונים והניתוב הוא בין אזורי AS שונים ולא בין ראוטרים.

כל AS מקבל מספר יחודי משלו בין 0 ל - 65535

כל נתב מקבל מהשכנים שלו את הניתובים שיש לשכנים שלו ולפי ניתובים אלו הוא מחשב את הנתיב הכי טוב עבור כל רשת ומעביר אותו לשכנים שלו.

**דרכי קביעת נתיב**

לפרוטוקול BGP יש כמה פרמטרים שלפיהם הוא קובע את הנתיבים המועדפים. הוא הולך לפי הפרמטר הראשון ואם הוא שווה בין חלק מהנתיבים אז הוא לפי הפרמטר הבא עד שהוא מגיע לנתיב אחד. הפרמטרים הם:



**Weight** – זו תכונה של נתבי סיסקו ויש בו ערך ספציפי לראוטר המיוחס לשכן והוא לא מפורסם לראוטרים אחרים. הוא דיפולטיבית 0, אפשר לשנות אותו. הערך המועדף הוא הערך הגדול ביותר.

**Local preference** – ערך המועבר בנתבים באותו AS, המטרה שלו היא לקבוע מה הדרך המועדפת לצאת ממנה באותו AS. ערך דיפולטיבי 100, מסלול ניבחר לפי הערך local preference הכי גבוהה שלו.

**Originate** – תכונה זו אומרת שהראוטר יעדיף נתיבים לרשתות שפורסמו מהראוטר במקום נתיבים אחרים לאותה רשת שהגיעו מראוטרים אחרים. למשל פירסום ב - network

**AS path length** – תכונה זו מעידה על כמה אזורי AS הנתיב עובר בשביל להגיע ליעד. יועדף הנתיב עם הערך AS path הכי נמוך.

הדרך בה bgp מבטיח שלא יהיו לולאות היא בכך שהוא זורק את כל הנתיבים בהם יש את ה – AS של אותו ראוטר.

**Origin code** - תכונה זו אומרת מאיפה הגיע נתיב זה אם מתוך ה – AS אז ibgp(i) אם מראוטר מ – AS שונה אז מ – ebgp(e) אם מפרוטוקל אחר למשל ospf ל – bgp אז incomplete(?)

התיעדוף הולך כך: i>e>?

כלומר תמיד יועדף נתיב שבא מתוך ה – AS מאשר מחוץ לה.

**MED** – זה הוא ערך המועבר בין אזורי AS והוא מעיד על מה הכניסה המועדפת לאזור, הערך הקטן יותר הוא המועדף.

**eBGP path over iBGP PATH** – תכונה זו אומרת להעדיף ניתובים שעוברים דרך eBGP על גבי ניתובים שעוברים ב – iBGP כלומר בשביל להגיע ליעד הסופי יועדף לא להיכנס ל – AS אלה לעבור על גביה

**Shortest IGP path to BGP next hop** – תכונה זו אומרת שאם הנתיב עובר בתור AS אז יועדף הנתיב עם עם המחיר הכי נמוך לקפיצה הבאה של BGP שחושב על ידי iBGP .

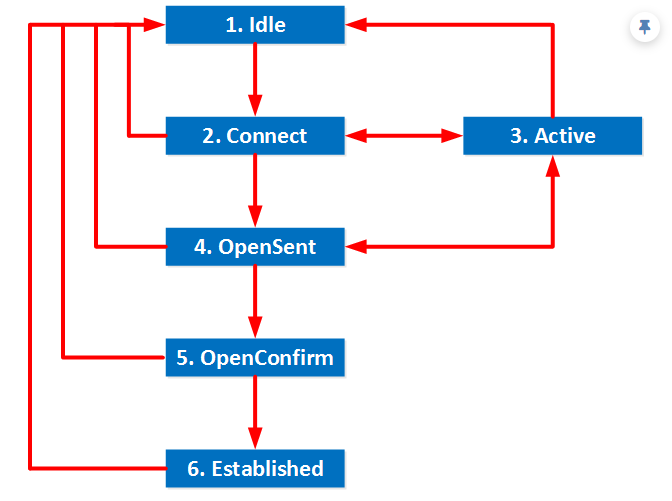
**Oldest path** – תכונה זו אומרת שאם עד עכשיו כל הנתיבים היו שווים בפרמטרים שלהם אז יועדף הנתיב שהגיע ראשון.

**Router ID** – תכונה זו אומרת שיועדף הנתיב של השכן בעל router ID ההכי קטן.

**Neighbor IP address** – תכונה זו אומרת שיועדף הנתיב של השכן עם הכתובת IP הכי נמוכה

## שכנות ב – BGP

**טבלת הסבר תהליך שכנות ב – BGP**

****

הודעות שכנות ב – BGP נישלחות בפורט 179.

ב – BGP שכנות חייבת להיות יזומה משני הצדדים והיא לא יכולה לקראות לבד.

**הסבר תהליך שכנות ב – BGP**

בתהליך השכנות של BGP יש 6 שלבים:

Idle – זה המצב הראשון של שכנות ב – BGP בשלב זה מאותחל ה – connectRetry timer שאומר כל כמה זמן תישלח הודעה לנסות ליצור קשר עם השכן.

בשלב זה הראוטר מנסה להתחיל חיבור TCP עם השכן וגם מאזין לשכן במקרה והוא מנסה ליצור קשר ועובר למצב.

שלב זה יתחיל ברגע שיוגדר ליצור קשר עם שכן חדש ב – BGP בראוטר.

Connect – במצב זה הראוטר יחכה three way handshake לסיים. אם החיבור סוים בהצלחה אז המצב מועבר ל – OpenSent ונישלחת הודעת Open ליעד להיות שכן. במקרה שהחיבור TCP לא הושלם לפני שה – connectRetry timer היסתיים אז הראוטר מועבר למצב Active.

אם משהוא אחר קורה למשל מתנתק הכבל או נשלחה הודעת איפוס אז הראוטר חוזר למצב Idle

Active – במצב זה הראוטר ינסה לעשות TCP three way handshake ויחכה עד משלושה דברים ייקרו:

1. החיבור יצליח ואז תישלח הודעה OPEN לשכן יעד והראוטר יעבור למצב OpenSent
2. ייקרה משהוא שייגרום לחיבור, להתאפס: למשל הודעת reset ואז יחזור למצב Idle
3. ה – connectRetry timer ייגמר ויעבור למצב Connect.

OpenSent – במצב זה נישלחת הודעת OPEN לשכן והוא מחכה להודעת OPEN וכאשר היא מתקבלת אז היא ניבדקת לטעויות, אם יש טעיות כמו מספר AS לא נכון או גרסה לא נכונה אז נישלחת הודעת Notification והראוטר חוזר למצב Idle, במיקרה ואין בעיות אז נישלחת הודעת keepalive והראוטר עובר למצב openConfirm. במיקרה והקשר TCP כשל אז הראוטר חוזר למצב active

OpenConfirm – הראוטר מחכה להודעת keepalive מהשכן, כשמקבל הודעת keepalive מהשכן אז עובר למצב established, אם הראוטר קיבל הודעת notification מהשכן אז עובר בחזרה למצב Idle

Established – במצב זה נוצר קשר בין שני השכנים של BGP, כך פעם שנישלחת הודעת update או keepalive אז ה – hold timer מתאפס. אם היתקבלה הודעת notification או ניגמר ה – hold timer אז הראוטר עובר למצב idle

## סוגי הודעות ב - BGP

**open** –

1. תוכן – version, my AS, hold time, bgp identifier(router ID), optional parameters.
2. מטרה – יצירת קשר שכנות עם שכן.

**Update** –

1. תוכן –
   1. Withdrawn route length – מראה את אורך השדה של הניתובים שצריך להוריד
   2. Withdrawn routes – מראה את הניתובים שצריך להוריד
   3. Total path attribute length – מראה את האורך של השדה של ה – path attributes של הניתובים שצריך להוסיף.
   4. Path attributes – מראה את הערכי path attributes של הניתובים שצריך להוסיף
   5. NLRI – מראה את הניתובים שצריך להוסיף.
2. מטרה – עידכון ופירסום רשתות שקיימות אצל הנתב.

**Keepalive** –

1. תוכן – יש בהודעת keepalive רק את המידע המינימלי
2. מטרה – לשמור על קשר שכנות

**Notification** –

1. תוכן –
   1. Error code – אומר מה הבעיה שקרתה
   2. Error subcode – מספר שמפרט יותר על הבעיה שקרתה
   3. Data – מראה את המידע של הבעיה
2. מטרה – להגיד שקרתה שגיאה ולסגור את הקשר בין השכנים

**Route refresh** –

1. תוכן – אין בהודעה תוכן נוסף חוץ מהמידע שחייבים בכל הודעה
2. מטרה – לעדכן את הניתובים שהתקבלו מהשכן מחדש ולקבל מחדש את כל הניתובים שלו.

## Router bgp

**הסבר**

נישתמש בפקודה זו בשביל להגדיר את ה – as של הראוטר ולהיכנס להגדרות bgp שלו

**תחביר**

Route bgp [as number]

**דוגמה של הגדרה**



## Network

**הסבר**

נישתמש בפקודה זו בשביל לפרסם רשת ב – bgp

**תחביר**

Network [net id] mask [subnetmask]

**דוגמה של הגדרה**



## Neighbor

**הסבר**

נשתמש בפקודה זו בשביל לנסות ליצור שכנות עם ראוטר אחר

**תחביר**

Neughbor [neighbor ip address] remote-as [neighbor as]

**דוגמה של הגדרה**



## Redistribute static

**הסבר**

נשתמש בפקודה זו בשביל לפרסם את הניתובים הסטטי על הראוטר לפרוטוקול אחר

**תחביר**

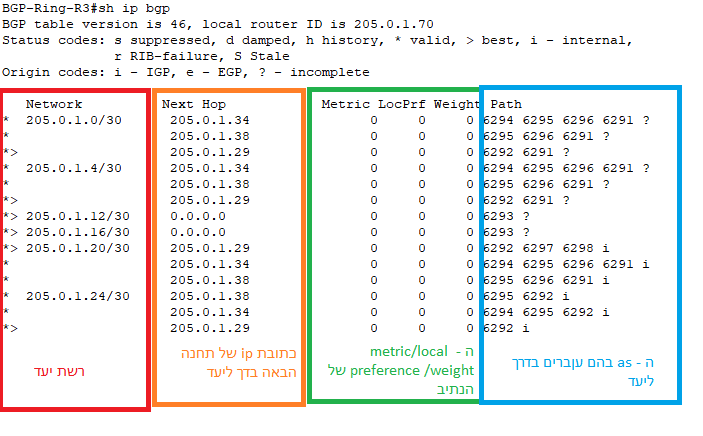
Redistribute static

**דוגמה של הגדרה**

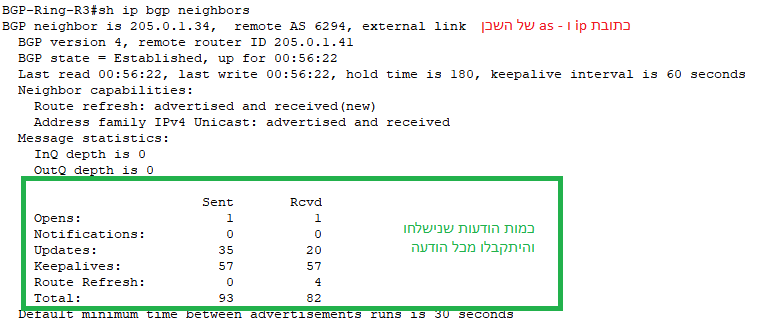


## פקודות show

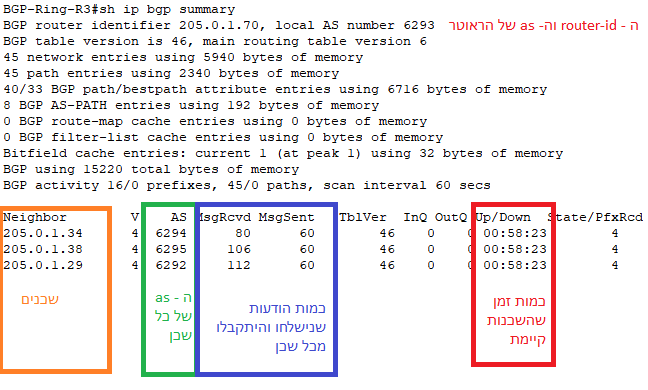
**Show ip bgp**



**Show ip bgp neighbors**

****

**Show ip bgp summary**

****

# NAT – Network Address Translation

## הסבר

הפרוטוקול nat הוא פרוטוקול הנותן לנו לתרגם כתובות ip פרטיות לכתובות ip ציבוריות

יש 3 סוגי nat:

Static NAT – מתרגם כתובת ספציפית פנימית אחת לכתובת ציבורית אחת

Dynamic NAT – מתרגם כמה כתובות פנימיות לכתובות ציבוריות רק שבמקרה זה אין כתובת פנימית שקשורה לכתובת חיצונית כלומר כאשר מחשב רוצה לצאת מהרשת אז כאשר מחשב מנסה לצאת מהרשת הציבורית דרך הראוטר אז הוא מקבל כתוב פנימית מהראוטר עד שהתעבורה שלו תסתיים ואז הכתובת תחזור לראוטר. מכיוון שעדיין כתובת מותאמת למחשב אז כמות הכתובות הציבוריות זה כמות המחשבים שיכולים לצאת לאינטרנט באותה נקודה

NAT overload/pat – מתרגם כמה כתובות פנימיות לכתובת ציבורית אחת. זה נעשה בכך שכל כתובת פנימית מתורגמת לכתובת החיצונית רק שהפורט מקור משתנה וכך כשחוזרת תשובה לראוטר הראוטר יודע למי שייכת ההודעה בכך שהוא מסתכל על פורט היעד ומתרגם את הכתובת והפורט לכתובת ip ופורט של המחשב המתאים

## Ip nat

**הסבר**

נשתמש בפקודה זו בשביל לקבוע אלו פורטים ישמשו כפורט פנימי ואלו פורטים ישמשו כפורט חיצוני

**תחביר**

Ip nat [inside/outside]

**דוגמה של הגדרה**

# 

## Ip nat inside source

**הסבר**

נשתמש בפקודה זו בשביל להגדיר את ה – nat

**תחביר**

Ip nat unside source list [ACL name/number] interface [the interface with the public ip] overload

**דוגמה של הגדרה**

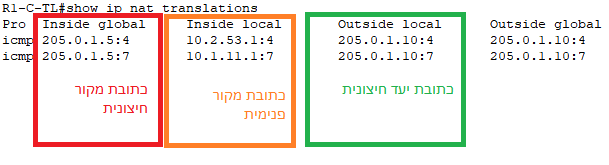


## פקודות show

**Show ip nat statisctics**

****

**Show ip nat translations**

****

# GRE – Genereic Routing Encapsulation

## הסבר

ה – GRE הוא פרוטוקול הנותן לנו לחבר שני רשתות פנימיות מבחינה לוגית בעזרת רשת מקשרת.

**איך**

הצורה בה ה – GRE עובד היא ככה שהוא לוקח את ה – header של ip ואת ה – payload ומוסיף להם header של gre שאומר באיזה פרוטוקול הייתה ההודעה למשל ipv4 ומוסיף לבסוף header של ip של הכתובות החיצוניות כלומר פאקטת ip רגילה:

**| Ip header | payload |**

אחרי ה – gre הפאקטה תראה כך:

**| added ip header | gre(ipv4) | ip header(original) | payload |**

לרשת המקשרת והקישור שאנחנו יוצרים ניקרה מינהרה.

בגלל שהמנהרה מבחינה לוגית היא רשת אחת אז כשנעשה traceroute נראה רק את הרשת של המינהרה בלי הרשתות החיצוניות שהודעה עצמה עברה בה

**מתי נישתמש**

נישתמש ב – gre בשביל ליצור מנהרה ולחבר בין שני רשתות. למשל כאשר יש לנו שני סניפים מרוחקים ואנחנו רוצים שהמחשבים בתוכם יוכלו לדבר אחד עם השני.

נקודה חשובה היא שהמינהרה של ה – gre רק מחברת בין הסניפים ולא מצפינה את העברת המידע בשביל זה נישתמש ב – ipsec vpn

## Interface tunnel

**הסבר**

פקודה זו נותנת לנו ליצור את הממשק הוירטואלי ל - gre

**תחביר**

interface tunnel [number]

**דוגמה של הפקודה**



## Tunnel destination

**הסבר**

נשתמש בפקודה זו כדי לקבוע את הנקודה אליה תתחבר המנהרה, הכתובת ip המצויינת תשומש בשביל כתובת יעד ב – ip header שמתווסף על ידי ה – gre.

**תחביר**

Tunnel destination [up address]

**דוגמה של הפקודה**



## Tunnel source

**הסבר**

נשתמש בפקודה זו כדי לקבוע את הנקודה אליה ממנה תצא המנהרה, הכתובת ip המצויינת תשומש בשביל כתובת מקור ב – ip header שמתווסף על ידי ה – gre.

**תחביר**

tunnel source [ip address]

**דוגמה של הפקודה**



# VPN – Virtual Private Network

## הסבר

ה – vpn נותן לנו לחבר בין שני איזורים בצורה מאובטחת כאשר התעבורה בניהם לא בהכרח מאובטחת.

למשל על גבי ה – wan כאשר אנו לא יודעים דרך אילו ראוטרים התקשורת עוברת ומי מקשיב לתקשורת.

קיימים שני סוגי VPN:

## Ipsec vpn/site to stie Vpn

חיבור זה נותן לנו לחבר בין שני נקודות קצה כאשר כל תעבור שתצא מנקודת קצה אחת ותעבור לשניה תהיה מוצפנת.

נשתמש ב – ipsec vpn למשל בשביל לחבר בין שני קצוות של סניפים.

יצירת חיבור ipsec vpn נעשת על ידי:

## IKE – Internet Key Exchange

זה פרוטוקול לעשיית משא ומתן וביצוע אימות בין שני הרכיבים.

ל – ike יש שני שלבים phase1 ו – phase2:

**IKE phase 1/ISAKMP**

**הסבר**

שלב משומש לאמת בין שני הרכיבים, ויצירת מנהרה מאובטחת לביצוע phase2. הרכיבים לא יכולים לעבור ל – phase 2 לפני ש – phase1 היסתיים.

עבור IKE phase1 יש כמה פרמטרים:

|  |  |
| --- | --- |
| **פרמטר** | **ערכים** |
| Hash – גיבוב | Md5,sha-256,sha512 |
| Authentication – אוטנטיקציה | PSK/RSA |
| Encryption – הצפנה | Aes,3des,des |
| Deffie hellman group – קבוצת דפי הלמן | 1,2,5,14,19,… |
| Lifetime – זמן מחיה | ערך דיפולטיבי – 86400 שניות  הערכים של הזמן מחיה יכולים להשתנות מעבר לכך |

**הסבר deffie hellman group**

קבוצות ה – deffie heleman נותנת לשני רכיבים ליצור מפתח סימטרי משלהם מבלי שהם אי פעם החליפו בניהם את המפתחות הסימטריים.

זה עובד בכך שכל רכיב יוצר זוג מפתחות לא סימטריים ושולך אחד מהם לצד השני. בעזרת המפתח הציבורי של הרכיב השני והמפתח הפרטי של הרכיב, הרכיב יוצר מפתח סימטרי שיהיה מתאים למפתח הסימטרי שהרכיב השני יצר.

נקודה חשובה היא שאם אחד מהפרמטרים האלה שונה אצל אחת מהרכיבים אז המנהרה של phase1 לא תקום.

## IKE phase2/ ipsec

מטרת שלב זה הוא לעשות משא ומתן על מנת להקים את המנהרה בה יעבור המידע. כלומר ומשא ומתן על איך המידע יוצפן, איך הוא יגובב וכן הלאה.

הפרמטרים של phase2:

|  |  |
| --- | --- |
| **פרמטרים** | **ערכים** |
| Hash – גיבוב | Sha – 256, sha 512, md5 |
| Encryption – הצפנה | Aes,des,3des |
| Deffie helman group – קבוצת דפי הלמן | 1,2,5,14,19,… |
| Lifetime – זמן מחיה | ערך דיפולטיבי – 3600 שניות  הערכים של הזמן מחיה יכולים להישתנות מעבר לכך |

## הגדרת crypto acl

בשביל להגדיר מתי מידע יחשב interesting traffic כלומר איזו תעבורה מותאמת ל – vpn ניצור רשימת גישה וכל תעבורה שתתאים לכללים של הרשימת גישה תוצפן על ידי ה – vpn

SSL(Secure Socket Layer) VPN

**הסבר**

ה – SSL VPN הוא vpn המשמש ליצירת קישור מאובטח בין רכיב לרשת מאובטחת אחרת בניגוד ל – site to site המחבר בין שתי קצוות הרשת.

יש שני סוגי SSL VPN:

**SSL Portal VPN**

זה VPN שבעזרתו מתחברים לאתר אינטרנט שיאפשר גישה לכל מיני שירותים של הרשת הפרטית. זה ניקרא מכיוון שזה אתר אינטרנט מ=שממנו יש גישה לכל השירותים האחרים. הגישה לשירותים ניתנת על ידי שם משתמש וסיסמה שיוצר ה – VPN יצר.

**SSL Tunnel VPN**

ה – VPN הזה נותן לרכיב מרחוק להיתחבר בחיבור מאובטח לרשת מקומית, כאילו הוא ממקום בתוך הרשת המקומית. החיבור נעשה בעזרת שם מישתמש וסיסמה שניקבע לפי נותן.

## הגדרת site to site vpn

בשביל להגדיר את ה – VPN צריך להגדיר את ike phase1, ike phase2, הנקודות קצה איתן ניצור vpn, הגדרת crypto map כדי לאגד את הכל, לשייך את ה – crypto map לפורט.

## Crypto isakmp policy

**הסבר**

משתמשים בפקודה זו בשביל ליצור policy חדש להגדרת ike phase1

**תחביר**

Crypto isakmp [number]

**דוגמה לפקודה**

## 

## Encryption

**הסבר**

נשתמש בפקודה זו בשביל להגדיר באיזור הצפנה נשתמש

**תחביר**

Encryption [type] [length]

**דוגמה לפקודה**

## 

## Hash

**הסבר**

## נשתמש בפקודה זו להגדיר באיזה גיבוב נשתמש

**תחביר**

Hash [type]

**דוגמה לפקודה**

## 

## Authentication

**הסבר**

נישתמש בפקודה זו כדי להגדיר באיזו צורה של אוטנטיקציה נישתמש

**תחביר**

Authentication [type]

**דוגמה לפקודה**



## Group

**הסבר**

נישתמש בפקודה זו להגדיר באיזו קבוצת deffie hellman נשתמש

**תחביר**

Group [group number]

**דוגמה לפקודה**



## Crypto isakmp key

**הסבר**

נשתמש בפקודה זו כדי להגדיר מפתח משותף לקצה אחד ספציפי

**תחביר**

Crypto isakmp key [pre shared key] address [ip address]

**דוגמה לפקודה**



## Crypto ipsec transform-set

**הסבר**

נישתמש בפקודה זו בשביל להגדיר איך יוצפן המידע והגדרת ike phase2

**תחביר**

Crypto ipsec transform-set [transform-set name] [encryption type] [hash type]

**דוגמה לפקודה**



## Crypto map

**הסבר**

נישתמש בפקודה זו בשביל להגדיר ולהיכנס להגדרות של crypto map שמאגדת את כל מה שהגדרנו עד עכשיו

**תחביר**

Crypto map [crypto map name] [isakmp policy number] ipsec isakmp

**דוגמה לפקודה**



## Set peer

**הסבר**

נישתמש בפקודה זו בשביל להגדיר עבור איזו נקודת קצה ה – crypto map הזו

**תחביר**

Set peer [ip address]

**דוגמה לפקודה**



## Set transform-set

**הסבר**

נישתמש בפקודה זו בשביל להגדיר באיזה transform-set להישתמש להצפנה

**תחביר**

Set transform-set [transform-set name]

**דוגמה לפקודה**



## Match address

**הסבר**

נישתמש בפקודה זו בשביל לבחור את ה – acl שתחליט מה היא interesting traffic

**תחביר**

Match address [acl name/number]

**דוגמה לפקודה**



## Crypto map

**הסבר**

נישתמש בפקודה זו בשביל להכיל את ה – crypto map על פורט ספציפי

**תחביר**

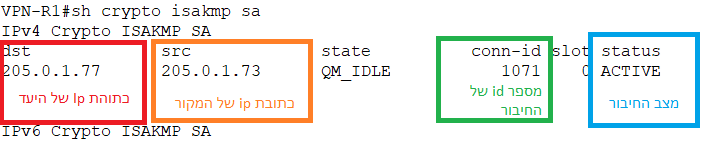
Crypto map [crypto map name]

**דוגמה לפקודה**

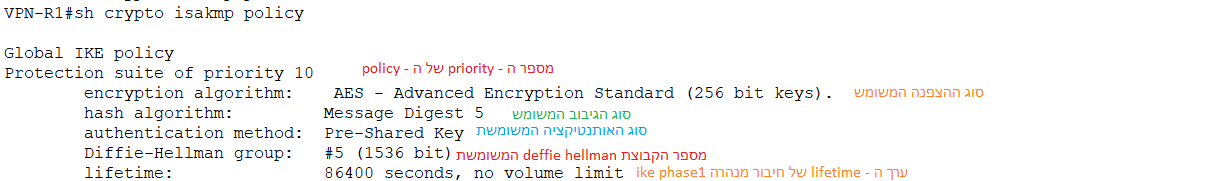


## פקודות show

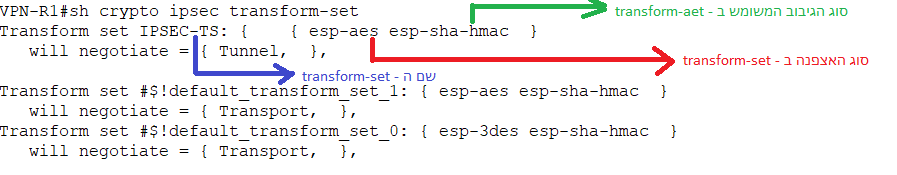
**Show crypto isakmp sa**



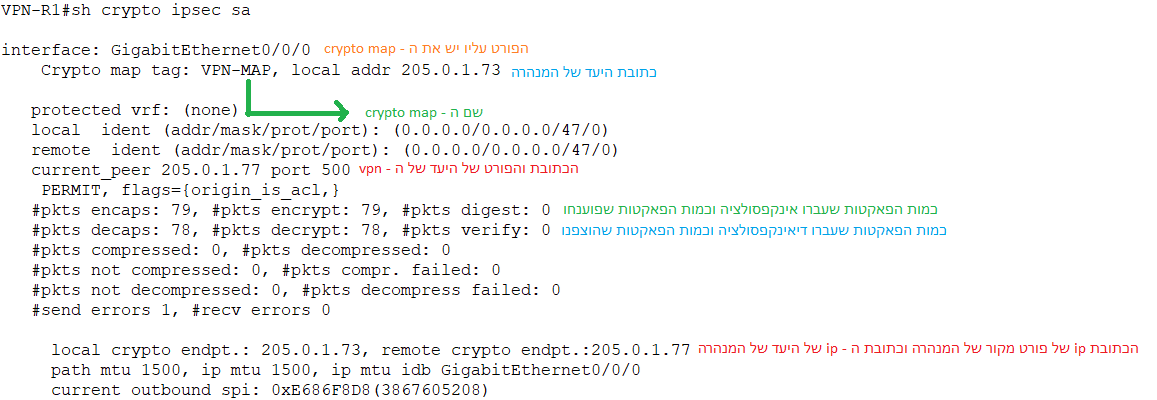
**Ship crypto isakmp policy**



**Show crypto ipsec transform-set**



**Show crypto ipsec sa**



# Etherchannel

## הסבר

ה – etherchannel הוא טכנולוגיה הנותנת לחבר כמה פורטים כפורט לוגי אחד, כך שכאשר שעובר עליו מידע הוא בעצם מעביר את המידע על כל הפורטים שהוא מאגד. כך שזה נותן להגדיל את המהירות העברה מידע ויוצר חלוקת עומסים.

## תנאים ליצירת etherchannel

* צורת duplex שווה בכל הממשקים
* מהירות חיבור זהה בכל הממשקים
* הגדרות vlan זהות
* מצבי ממשקים זהים (trunk או access)

## הפרוטוקולי etherchannel

יש שני פרוטוקולי etherchannel הנותנים יצירה של etherchannel:

1. PAgp (Port Agregation Protocol):

זה פרוטוקול השייך ל – cisco והשימוש בו מורשה על ידי cisco.

פרוטוקול זה נותן יצירה אוטומטית של etherchannel.

בין כל פורט נישלחות הודעות כל 30 שניות כדי לתחזק את הקשר.

הפרוטוקול נותן חיבור של עד 8 פורטים לפורט לוגי אחד.

מצבי הפרוטוקול:

1. Desirable – מצב בו הפורט ינסה לצור משא ומתן עם הצד השני
2. Auto – מצב בו הפורט יחכה לצד השני ליזום משא ומתן ורק אז הוא יעשה משא ומתן
3. LACP(Link Aggregation Control Protocol):

זה פרוטוקול פתוח לכולם שזהה בצורה רבה לפרוטוקול PAgP.

החלק השונה בפרוטוקול זה הוא שיש לו שמות שונים למצבים, בגלל שהוא פתוח לכולם אז הוא יכול לעבוד על כל המתגים.

נותן עד 8 פורטים שמחוברים ו – 8 שמחכים בגיבוי למקרה שהפורטים שעובדי נופלים

מצבי הפורט:

1. Desirable – יוזם משא ומתן עם הפורט ממולו
2. Passive – מתחיל משא ומתן רק אם הפורט ממולו יוזם משא ומתן

יש עוד מצב לשני הפרטוקולים שניקרא on:

1. On – במצב זה לא נעשה שימוש באחד מהפרוטוקולים. הוא מוגדר ידנית בשביל לחבר בלי ניהול משא ומתן בין הפורטים. הוא חייב להיות מוגדר על שני הפורטים בשביל ליצור etherchannel

## למה להישתמש

* מהירות: נותן מהירות גבוהה יותר
* יתירות: נותן לשים כמה פורטים לגיבוי במקרה ופורט אחד נופל

## עוקף stp

בדרך כלל כאשר שני פורטים או יותר מחוברים מאותו מתג לאותו מתג בדיוק אז נוצרת לולאה וכולם חוץ מאחד נחסמים.

אבל ב – etherchannel נוצר פורט לוגי אחד שמאגד את כל הפורטים הפיזים ככה stp מתייחס לפורט הלוגי כפורט אחד ולא חוסם אותו.

## Interface port-channel

**הסבר**

נשתמש בפקודה זו בשביל להגדיר etherchannel ולהיכנס להגדרות שלו

**תחביר**

Interface port-channel [number]

**דוגמה לפקודה**



## Channel-group

**הסבר**

נשתמש בפקודה זו בשביל להגדיר לאיזה etherchannel הפורט שייך, באיזה פרוטוקול לעבוד ובאיזה מצב

**תחביר**

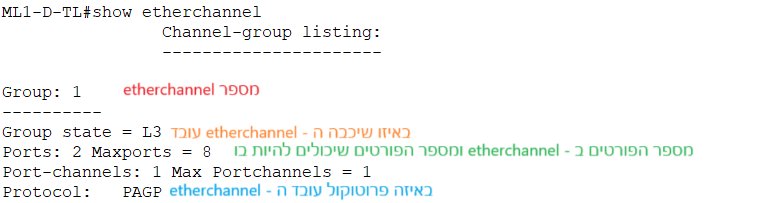
Channel-group [number] mode [desiarable/auto/active/passive/on]

**דוגמה לפקודה**

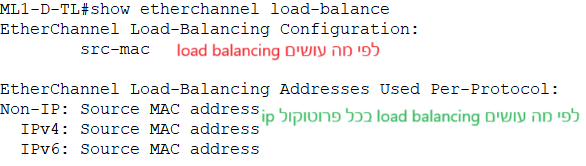


## פקודות show של etherchannel

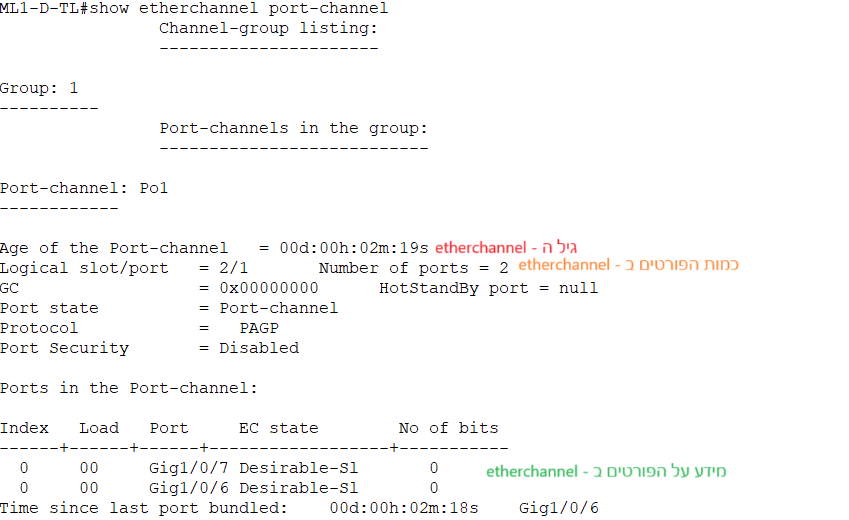
**Show etherchannel**

****

**Show etherchannel load-balance**

****

**Show etherchannel port-channel**

****

**show etherchannel summary**

**A screen shot of a computer program

AI-generated content may be incorrect.**

# הגנות והתקפות סייבר

## Port security

**מה זה port security**

ההגדרה של port security היא הגדרה שנותנת להגדיר איזה רכיבים יוגלו להיתחבר לאותו פורט לפי כתובות המאק של הרכיבים המתחברים

בתוך port security יש כמה הגדרות:

1. Sticky – לא חוסם רכיבים ושומר את הכתובת מאק של הרכיבים המחוברים לאותו פורט
2. max – הכמות המקסימלית של כתובות המאק שיכולות להישמר על אותו פורט
3. Violation type – יש כמה סוגי תגובה למחשב שהיתחבר ולא שמורה כתובת המאק שלו והגדרה זו קובעת מה התגובה

סוגי ההפרות

1. shutdown – חוסם את הרכיב וחוסם את הפורט, בשביל להפעיל אותו עוד פעם צריך לכבות ולהפעיל ידנית את הפורט ושולח log לשרת log
2. Restrict – חוסם את הרכיב ושלוח הודעת log לשרת log
3. Protect – רק חוסם את הרכיב

## פקודת port security

**מה עושה**

מפעילה את ההגדרה של port security

**תחביר**

Switchport port security

**תמונה של הפקודה**



## פקודת mac-address sticky

**מה זה**

הגדרה שנותנת לכך שהכתובת מאק של הרכיב המוחבר לפורט תישמר מבלי להזין אותה ידנית

**תחביר**

switchport port-security mac-address sticky

**תמונה של הפקודה**



## פקודת violation

**מה זה**

משמשת להגדרת מה לעשות ברגע שהייתה הפרת אבטחה

**תחביר**

switchport port-security violation [shutdown, restrict, protect]

**תמונה של הפקודה**



## פקודות show

**Show port-security**



**Show port-security interface [interface name]**



## טבלת ייחוס vlan לסוג violation

|  |  |
| --- | --- |
| **vlan** | **Violation type** |
| HR – human recources(53) | Shutdown |
| MAN – managers(11) | Shutdown |
| STO – storage(114) | Shutdown |
| WOR – workers(37) | Restrict |
| SAL – sales(123) | Protect |
| REC – reception(39) | Protect |
| WW – warehouse workers(117) | Restrict |
| DRI – drivers(119) | Restrict |
| INV – inventory(122) | Protect |
| QA – quality assurance(150) | Protect |
| RES – research(160) | Restrict |
| DEV – development(170) | restrict |

# ASA

ה – cisco asa הוא רכיב של cisco שמתפקד כ – firewall עם כמה תכונות מיוחדות לו שנותנות לו לעשות את התפקיד שלו כ – firewall יותר טוב:

1. שם פורט - Asa נותן להגדיר שם לפורט ולאחר מכן אפשר להשתמש בכל ההגדרות בשם הפורט שניתן ולא בשמו הרגיל (למשל gig0/0). אפשרות זו עוזרת בכך שהיא שהרבה יותר קל ליזכור משהוא לפי שם מאשר מספר, בנוסף השם אומר מה המטרה של הפורט כדי למנוע בילבול.
2. Security level - לכל פורט ב – asa יש security level, כלומר רמת אבטחה. טווח הערכים הוא בין 0 ל – 100 כאשר 0 זה רמת הבטחה הכי נמוכה ו – 100 זו רמת ההבטחה הכי גבוהה.

מנגנון ה – security level עובד כך שמידע שעובר מפורט בעל רמת אבטחה גבוהה לנמוכה יותר עובר בלי בעיה, אבל עבור מידע שעובר מפורט בעל אבטחה נמוכה יותר לאבטחה גבוהה יותר אז הוא צריך להיות מאושר על ידי acl ספציפיט. כך רק תעבורה רצויה תוכל לעבור לאיזורים יותר בטוחים עם כמות שגיאות מינימאלית.

דוגמה להגדרת פורטים ב – asa:

1. פורט OUTSIDE: זה הפורט שיהיה מחובר ל – wan. בגלל שהוא מחובר ל – wan אז אנחנו לא יודעים מי בדיוק האחד שניכנס מהפורט והודעות שבאות מפורט זה הן מחוץ לארגון, אז לפורט זה תהיה את רמת האבטחה הכי נמוכה.
2. פורט DMZ: פורט זה יהיה מחובר לאזור השרתים בארגון. בגלל שאזור השרתים הוא בתוך האירגון שלנו אז ניתן לו אבטחה גבוהה יותר מהפורט OUTSIDE. למרות זאת עדיין נירצה לתת גישה לחלק מהשרתים מהבאים מה – wan לכן ניפתח כמה שירותים דרך acl אבל לא את כולם.
3. פורט INSIDE: פורט זה יהיה הפורט שמחבר לתוך האירגון לרשת הפרטית שלו. בגלל שאנו לא נירצה שתהיה גישה לתוך הרשת הפרטית מה – wan או DMZ אז ניתן לו את הרמת אבטחה הכי גבוהה. למרות זאת עדיין נאפשר למידע שמכוון לרשת הפנימית דרך acl כי אחרי שמידע שיצא מפורט INSIDE ל – wan חוזר דרך פורט OUTSIDE ועובר תירגום דרך ה – nat הוא עדיין צריך אישור כי הוא עובר פורט מרמת אבטחה נמוכה יותר לפורט עם רמת אבטחה גבוהה יותר. בכך שנאשר ב – acl רק כניסה לבודעות שמכוונות לרשת הפרטית, נוכל לאפשר רק להודעות שמקורן ברשת הפרטית לעבור כי רק הודעות שמקורן ברשת הפרטית יתורגמו על ידי ה – nat בחזרה.

## Nameif

**הסבר**

נשתמש בפקודה זו בשביל לתת שם לפורט

**תחביר**

Nameif [name]

**דוגמה לפקודה**



## Security-level

**הסבר**

נשתמש בפקודה זו בשביל לקבוע את רמת האבטחה של פורט

**תחביר**

Security-level [number]

**דוגמה לפקודה**



## Object-network

**הסבר**

נשתמש בפקודה זו בשביל להיכנס להגדרות של אובייקט רשת שבעזרתו ניצור nat

**תחביר**

Object network [name]

**דוגמה לפקודה**



## Subnet

**הסבר**

נשתמש בפקודה זו בשביל להגדיר את הרשת עליה יכול האובייקט רשת

**תחביר**

Subnet [net-id] [subnetmask]

**דוגמה לפקודה**



## Nat

**הסבר**

נשתמש בפקודה זו בשביל להגדיר את ה – nat

**תחביר**

Nat ([inside port name],[outside port name]) [dynamic/static] [interface/{outside ip address}]

**דוגמה לפקודה**



## Route

**הסבר**

נשתמש בפקודה זו להגדיר static route

**תחביר**

Route [port name to sed through] [net-id] [subnetmask] [net hop ip]

**דוגמה לפקודה**



## Access-list

**הסבר**

נשתמש בפקודה זו כדי להגדיר חוק ב – acl

**תחביר**

Access-list [acl name] extended] [permit/deny] [protocol name] [source] [destination]

**דוגמה לפקודה**



## access-group

**הסבר**

נשתמש בפקודה זו בשביל לשייך acl ל פורט

**תחביר**

access-group [acl name] [in/out] interface [interface name]

**דוגמה לפקודה**



## Username

**הסבר**

נשתמש בפקודה זו להגדיר משתמש לוקלית

**תחביר**

Username [username] password [password]

**דוגמה לפקודה**



## Ssh

**הסבר**

נשתמש בפקודה זו כדי להגדיר מאיזה פורט ומאיזה רשת אפשר להתחבר ב – ssh ל – asa

**תחביר**

Ssh [net-id] [subnetmask] [port name]

**דוגמה לפקודה**



## Aaa authentication console LOCAL

**הסבר**

נשתמש בפקודה זו בשביל להגדיר מאיפה לקחת המשתמשים לאוטנתיקציה כשאר מתחברים בקישור מרוחק

**תחביר**

Aaa authentication [ssh/telnet] console LOCAL

**דוגמה לפקודה**



## dhcpd address

**הסבר**

נשתמש בפקודה זו בשביל להגדיר dhcp על פורט ספציפי ואילו כתובות יחולקו

**תחביר**

Dhcpd address [start ip address range]-[end ip address range] [port name]

**דוגמה לפקודה**



## dhcpd enable

**הסבר**

נשתמש בפקודה זו בשביל לאפשר dhcp על פורט ספציפי

**תחביר**

Dhcpd enable [port name]

**דוגמה לפקודה**



# SSH

## הסבר

פרוטוקול SSH הוא פרוטוקול הנותן חיבור מרחוק לרכיב בלי להשתמש בכבל פיזי. כלומר הוא נותן להתחבר מרחוק בצורה מאובטחת אל התקן נוסף ברשת כגון ראוטר, מתג.

## איך עובד

**שלב 1: בקשת חיבור**

הרכיב שרוצה להתחבר שולח בקשת התחברות, בה הוא מציין את גרסת הפרוטוקול ורשימה אלגוריתמי ההצפנה ואימות הנתמכים.

**שלב 2: תגובת המחובר**

מי שנישלחה אלליו בקשת החיבור עונה לשולח עם: גרסת הפרוטוקול שלו, רשימת אלגוריתמי הצפנה ואימות שלו. בנוסף הוא יכול גם לשלוח את המפתח הציבורי שלו אם הלקוח לא קיבל את המפתח בעבר.

**שלב 3: בקשת אימות מהשולח**

לאחר אימות המפתח הציבורי של השרת, השולח בקשת אימות לשרת. בה הוא מציין את השם משתמש וסיסמה שלו.

**שלב 4: אימות לקוח על ידי מקבל הבקשה**

השרת מאמת את פרטי המשתמש שקיבל מהשולח. אם האימות הצליח, השרת ולח הודעת הצלחה בחזרה ללקוח.

**שלב 5: יצירת החיבור המאובטח**

אחרי שהאימות הצליח נוצר חיבור מאובטח בין השולח למקבל הבקשה.

**שלב 6: תקשורת**

כעת נוצר קשר מאובטח וכל התקשורת תהיה מוצפנת על ידי האלגוריתם המוסכם, וניתן לעשות פעולות במכשיר אליו היתחברו.

## Ip domain-name

**הסבר**

נשתמש בפקודה זו בשביל להגדיר שם דומיין (חייבים להגדיר את זה בשביל להגדיר ssh)

**תחביר**

ip domain-name [word]

**דוגמה לפקודה**



## Username

**הסבר**

נשתמש בפקודה זו בשביל להגדיר משתמש עם סיסמה

**תחביר**

Username [username] privilege [privilegelevel] password [password]

**דוגמה לפקודה**



## Line vty

**הסבר**

יש כמה חיבורים מרחוק שאפשר לעשות באותו זמן, נשתמש בפקודה זו בשביל להגדיר את כל החיבורים.

**תחביר**

Line vty [start] [end]

**דוגמה לפקודה**



## Transport input

**הסבר**

**תחביר**

Transport input [ssh/telnet]

**דוגמה לפקודה**



## Login local

**הסבר**

נשתמש בפקודה זו בשביל להשתמש בבסיס נתונים המקומי של הרכיב בשביל משתמשים

**תחביר**

login local

**דוגמה לפקודה**



## Crypto key generate

**הסבר**

נשתמש בפקודה זו בשביל להשתמש ב – rsa בשביל מפתחות

**תחביר**

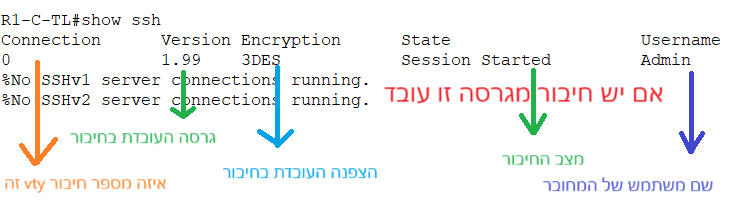
Crypto key generate rsa

**דוגמה לפקודה**



## פקודות show ssh

**Show ssh**



# AAA

עיקרון AAA הוא עיקרון אבטחה שמתקשר למשתמשים הוא עובד כך:

Authentication – אימות המשתמש

Autherization – אישור גישה למשתמש רק למה שהוא צריך

Accounting – שמירת נתונים על הפעולות של המשתמשים

נשתמש ברעיון זה וניצור לו שרת משלו שיישם את עקרון זה וזה שרת AAA.

יש שני פרוטוקולים של אוטנתיקציה: radius, tacacs+

|  |  |  |
| --- | --- | --- |
| **תכונה** | **TACACS+** | **RADIUS** |
| **סוג פרוטוקול** | שייך לסיסקו | פתוח לכולם |
| **הפרדת תהליכים** | הפרדות של פונקציות AAA | שילוב אימות והרשאה ומקשה על הפעלה בודדת |
| **הצפנה** | מצפין כל התקשורת | מצפין רק סיסמה בזמן הקמת קשר |
| **הרשאה** | יכול לשלוט ברמת הרשאה שמקבל משתמש ב – cisco | לא יכול לשלוט ברמת הרשאה שמקבל משתמש ב – cisco |
| **פרוטוקול תקשורת** | משתמש ב – TCP פורט 49 | משתמש ב – UDP פורט 1813/1812 |

## Aaa new-model

**הסבר**

נשתמש בפקודה זו כדי להפעיל את תפקוד ה – aaa

**תחביר**

aaa new-model

**דוגמה לפקודה**



## Aaa authentication

**הסבר**

נשתמש בפקודה זו בשביל להגדיר היתחברות מול סרבר

**תחביר**

Aaa authentication login default group tacacs+/radius local

**דוגמה לפקודה**



## Tacacs-server

**הסבר**

נשתמש בפקודה זו בשביל להגדיר שרת tacacs+

**תחביר**

Tacacs-server host [server ip] key [password]

**דוגמה לפקודה**



## Login authentication

**הסבר**

נשתמש בפקודה זו בשביל להגדיר דיפולטיבית להשתמש לאימות בשרת aaa



## Enable password

**הסבר**

יוצר סיסמה למצב enable

**תחביר**

Enable password [password]

**דוגמה לפקודה**



## תמונה של הגדרות השרת aaa

A screenshot of a computer

AI-generated content may be incorrect.

# DHCP snooping

שימוש ב – dhcp snooping בא לפתוח התקפה שקוראים לה – dhcp spoofing

## Dhcp spoofing

בהתקפה רכיב מעמיד פנים שהוא dhcp וכך הוא יכול להשפיע על התעבורה ברשת.

זו רכיב ברשת מקבל את ההודעת discover של רכיב והוא עונה בתור שרת dhcp כך הרכיב נותן את הכתובת ip והמידע שמתווסף להן לאחד שביקש את הכתובת ip.

## איך dhcp snooping עובד

ב – dhcp snooping נגדיר פורטים שהם trusted ופורטים שהם untrusted. דרך פורט שהוא trusted יכולים לעבור כל הודעות dhcp, דרך פורטים שהם untrusted לא יכולים לעבור הודעות offer והודעות acknowledge

## ip dhcp snooping vlan

**הסבר**

נשתמש בפקודה זו בשביל להגדיר אל אילו vlan נפעיל dhcp snooping

**תחביר**

ip dhcp snooping vlan [vlan number]

**דוגמה לפקודה**



## Ip dhcp snooping trust

**הסבר**

נשתמש בפקודה זו בשביל להגדיר פורט כ – trust

**תחביר**

Ip dhcp snooping trust

**דוגמה לפקודה**



**הסבר עם no**

כשמוסיפים no לפני הפקודה זה הופך פורט ל – untrusted

**תחביר**

no ip dhcp snooping limit rate



## Ip dhcp snooping limit

**הסבר**

נשתמש בהגדרה זו כדי להגדיר כמה הודעות dhcp יכולות לעבור בשנייה בפורט

**תחביר**

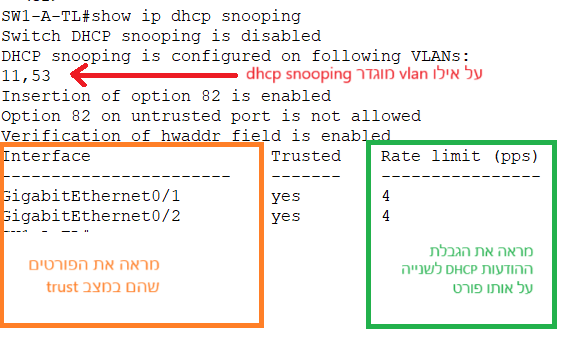
ip dhcp snooping limit rate [number]

**דוגמה לפקודה**



## פקודות show ל – dhcp snooping

**Show ip dhcp snooping**

****

# native vlan

ה – native vlan הוא vlan ספציפית לפורט שמעביר תעבורה שאין לה tag.

אפשר להשתמש ב – native vlan בשביל לעשות התקפה שקוראים לה vlan hopping

## Vlan hopping

ההתקפה vlan hopping זו התקפה שבה נותניפ ל – frame שני תגים אחד פנימי של ה – vlan אליו רוצים להגיע ואחד חיצוני של ה – native vlan. כאשר ה – frame יגיע למתג המתג יוריד את התג של ה – native vlan ויישלח אותו ישירות בלי שינוי לפורט trunk והוא עושה זאת בגלל שהוא חושב שזה frame שלא אמור להיות לו תג. כאשר ה – frame מגיע למתג בצד השני ל – frame עדיין יש את התג של ה – vlan אליו רוצים להגיע ולעשות עליו vlan hopping אז המתג מעביר את ה – frame לאותו vlan כי הוא רואה של – frame יש תג.

בעיה זו יכולה לקרות רק כאשר נישלחת הודעה מפורט על vlan שהוא בשימוש לכן נשנה את ה – native vlan ל – vlan שהוא לא בשימוש.

## Switchport trunk native vlan

**הסבר**

נשתמש בפקודה זו בשביל להגדיר native vlan עבור פורט ספציפי

**תחביר**

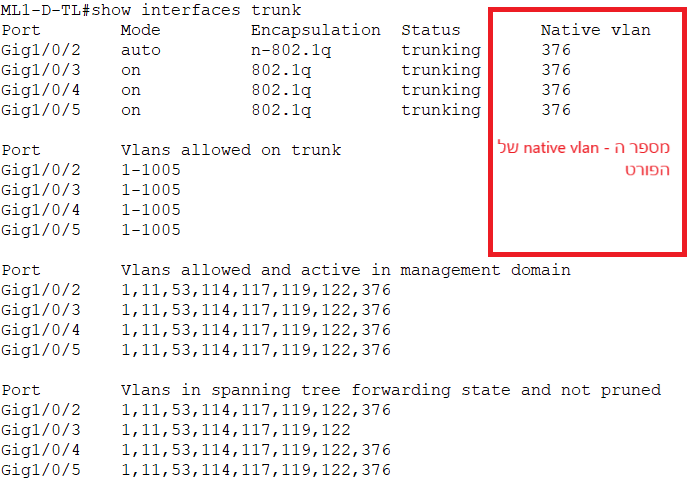
Switchport trunk vlan [number]

**דוגמה לפקודה**



## פקודות show של native vlan

**Show interfaces trunk**

****