**Network assurance**

* Ping – יכולת לשלוח מידע מכתובת IP אחת לאחרת על מנת לבדוק קישוריות ביניהן
* Debugging - כלי אשר משמש לבדיקת שינוים אשר מתבצעים על ציוד תקשורת בזמן אמת

**SNMP**

כאשר יש לנו ברשת איזשהו NMS נררצה שהוא יצליח לבדוק את הציודים מבוססי IP שלנו וגם לקבל מהם מידע, על מנת לעשות זאת יצרו פרוטוקול אשר משמש לשליחת מידע מלקוח מסוים לשרת שגם יכול לבדוק דברים על הלקוח.

* השרת יהיה SNMP MANAGER, תוכנה אשר רצה על שרת מסוים אשר ינטר את הציודים ברשת
* הלקוח יהיה SNMP AGENT, סוכן אשר ירוץ על הציוד שנרצה לנטר ויאפשר שליחה וקבלת מידע מהשרת
* לכל ציוד שמריץ SNMP יש DBTABASE שנקרא MIB – management information base אשר מכיל בתוכו אובייקטים כאשר כל אובייקט הוא פורט במצב של UP/DOWN או גרף של הCPU או שכן OSPF וכדומה.   
  כל אובייקט נקרא OID - ObjectIdentifier ויש לו מספר זיהוי ייחודי על אותו ציוד.
* השרת יכול לשלוח POLLים אשר יבקשו מהציוד מידע על אחד מהערכים שקיימים לו בMIB וכך ניתן לאחסן בשרת אירועים שקורים בציוד ובמשך תקופה גם לייצר סיכומים על אותו ציוד.
  1. אותם POLLים יבוצעו על ידי הודעת SNMP GET REQUEST
* ניתן גם לשלוח מידע בצורה מידית מהציוד אל השרת שמשמש כNMS, הודעות שנשלחות על ידי הציוד נקראות TRAP
* כדי למצוא ערך מסוים יש לבקש אותו על פי הOID שלו, אבל השאלה איך הMIB יודע מה מסמל כל OID???  
  הMIB בנוי בצורה הירכית כך שכל ערך נמצא תחת עץ שנבנה לפי RFC, VENDOR כמו CISCO שיש להם פרוטוקולים קניינים גם מייצרים עצים בהתאם לפרוטוקול ולמידע הרלוונטי אליו
* ישנם 2 רמות של הרשאות בSNMP:
  1. RO – read only – מאפשר רק לקבל מידע מהAGENT, ניתן לקבל תשובה רק על הודעות SNMP GET
  2. RW – read write – מאפשר לקבל מידע ולהגדיר את הציוד בו נמצא הAGENT, ניתן לקבל תשובה על הודעות SNMP GET והודעות SNMP SET
* לSNMP יש 3 גרסאות:
  1. V1 – ישן וכבר בקושי בשימוש, משתמש בPLAIN-TEXT כדי לבצע AUTH מול הAGENT
  2. V2C – ישן וכבר בקושי בשימוש, משתמש בPLAIN-TEXT כדי לבצע AUTH מול הAGENT
  3. V3 – חדש ובשימוש בתקופה של ימינו, משתמש בAUTH של משתמש מול הAGENT ויש 3 סוגים:
     + noAuthNoPriv – שם משתמש אבל ללא הצפנה
     + authNoPriv – MD5 או SAH אבל ללא הצפנה
     + authPriv – MD5 או SAH ועם הצפנה

**Syslog**

מנגנון אשר מזהה אירועים שונים בציוד ומקפיץ הודעות SYSLOG אשר מסבירות מה קרה ומתי.

* צפייה בלוגים:
  1. לראות היסטוריית לוגים show logging, יש מגבלה של כמה BYTES הציוד ישמור וניתן לשנות אותה בציודי CISCO על ידי הפקודה logging buffer <bytes size>
  2. בזמן שמנהל רשת מחובר לציוד של CISCO לא יקפצו הלוגים בזמן אמת מכיוון שהוא צריך לאפשר זאת עם הפקודה: terminal monitor (כאשר מתחברים בSSH/TELNET)  
     בCONSOLE זה מאופשר דיפולטיבי עם הפקודה logging console
* ניתן להגדיר לציוד שרת לוגים אשר אליו הציוד ישלח את הלוגים, משמש לשמירת היסטוריית לוגים
* מבנה ההודעה:
  1. Timestamp – מתי ההודעה קפצה (על פי הזמן של הציוד)
  2. Facility – כותרת במציינת מה סוג האירוע שקרה, כל כותרת מכילה מספר תתי כותרות שמסבירות בקצרה מקרים מסוימים השייכים לאותה כותרת.
  3. Severity level – רמת חומרה של ההודעה
  4. Mnemonic – תת כותרת לאירוע שקרה, כל תת כותרת מציינת בקצרה מקרה אחר
  5. Description – פירוט מלא של מה שקרה
* Severity level – מכיוון שישנם הודעות חשובות יותר אשר נרצה לקטלג אותם לפי רמת חשיבות, ישנם 8 רמות שונות, ככל שהמספר של הרמה נמוך יותר ככה ההודעה חשובה יותר.

1. Emergency – system unstable
2. Alert –immediate action needed
3. Critical – critical condition
4. Error – error condition
5. Warning – warning condition
6. Notice – normal but significant condition
7. Informational – informational message only
8. Debug – debugging message

* בחלק של syslog נדרש לזכור בעל פה את כל הlog level בעל פה ואת המספר והתיאור שלהן

**DEBUGGING**

לא על כל דבר שקורה בציוד קופץ LOG, על מנת לראות דברים שקורים בכל פרוטוקול נשתמש בDEBUG ונראה את המידע בצורה מפורטת.

* נרשום את הפקודה DEBUG ואז סימני שאלה ומשם לנסות מה שרוצים.
* ניתן להוציא את הלוגים לתוך קובץ.

**SPAN**

עולמות L2

הרבה פעמים נרצה לנתח מידע מהמון סיבות(מה הסוג של המידע עובר/איזה אפליקציות בשימוש) ניתן להגדיר שיכפול מידע מפורט אחד לפורט אחר או ממספר פורטים למספר פורטים אחרים.

הגדרות:

* Monitor session <session number> source interface <interface name> tx/rx/both
* Monitor session <session number> destination interface <interface name>

RSPAN

אם נרצה להעתיק מידע מפורט במתג אחד ופורט במתג אחר נשתמש במשהו שנקרא RSPAN, נגדיר איזהו VLAN שישמש כVLAN רק לRSPAN.

על המתג בו יש את הפורט מקור נגדיר:

* Monitor session <session number> source interface <interface name> tx/rx/both
* Monitor session <session number> destination remote vlan <vlan id>

על המתג בו יש את הפורט יעד נגדיר:

* Monitor session <session number> source remote vlan <vlan id>
* Monitor session <session number> destination interface <interface name> tx/rx/both

ERSPAN

ניתן לשלוח שכפול של מידע על גבי L3 בעזרת משהו שנקרא ERSPAN שמאפשר לשלוח את המידע ליעד IP מסוים (חשוב שהיעד ידע לעשות decapsulation למידע אחרת אנחנו סתם שולחים לו) וכך לשלוח הקלטות לשרת הקלטות שיש לנו ברשת.

* נגדיר session מסוג erspan
* נגדיר פורט מקור
* ביעד נגדיר כתובת IP של היעד
  1. בנוסף נגדיר ID לERSPAN
  2. נגדיר IP מקור ממנו יצא המידע
* נגדיר כמה TTL יהיה לפקטה

**Flexible NetFlow**

שדרוג של NetFlow רגיל, ביצוע האזנה לtraffic ברשת ושליחת הcapture לשרת NetFlow שיש לי ברשת.

לדוגמא: ניתן לבדוק כמה פקטות נשלחות ליעד מסוים ומזה להבין למי נשלח לי הכי הרבה מידע בחלק מסוים ברשת

יכולת להעתיק תעבורה(FLOW) של מידע בL3 ושליחה לאיזה NMS.

ניתן לבצע תפיסה למידע שנכנס/יוצא ולתפוס את המידע על פי ערכים מסוימים:

* IP מקור
* IP יעד
* פורט לוגי
* ישנם RECORD מוכנים דיפולטיבי על הציוד

יצירת Flexable NetFlow מורכב ממספר חלקים

* Record
  1. איזה מידע נרצה לתפוס
  2. מה נרצה לתפוס באותו סוג מידע, עד רמת השדות בפקטה
* Export
  1. לאן נרצה לייצא את המידע ->כתובת IP
  2. באיזה פורט לוגי
  3. באיזה פרוטוקול לשלוח לו את המידע (netfloe-v9/netflow-v5) החדש יותר זה 9
* Monitor
  1. לוקח record וexport ומקשר ביניהם
  2. נייצר monitor על מנת לתפוס מידע (record) ולשלוח אותו ליעד מסוים(export)
* הגדרה על ממשק בו נרצה לתפוס את המידע
  1. נגדיר על הממשק את הmonitor אותו נרצה להכיל ואז המידע שייתפס יהיה מהממשק עליו הגדרנו את הmonitor

נגדיר לכל RECORD איזה פרט במידה נרצה לתפוס, לדוגמא: אם נרצה לתפוס את כל המידע ליעד מסוים (KEY) ונרצה לדעת רק כמה פקטות נשלחו אליו(NON-KEY)

**IP SLA**

מאפשר לי לשנות את הצורה בה הנתב עובד במקרה ובדיקה מסוימת נכשלת