Protocol RFC 913 – Simple File Transfer Protocol

סמסטר א

שנה"ל תשע"ח

|  |
| --- |
| **מגישים** |
| דקל תמם, ת.ז. 305242125, הנדסת תוכנה  דקל משה, ת.ז. 200862142, הנדסת תוכנה |

תוכן העניינים

1 רקע הפרוטוקול 3

1.1 רקע כללי והיסטוריה 3

1.2 סיכום תקן RFC 3

1.3 סיכום נייר אקדמי - Simple File Transfer Protocol - Debashis Gangully, Darshan Shetty 4

1.4 סיכום נייר אקדמי – Characteristics of Network Traffic Flow Anomalies – Paul Barford, David Plonka 5

1.5 סיכום רקע כללי 6

SFTP – Deployment Diagram 7

2 הגדרת הניסוי 8

2.1 בדיקות לקוח/שרת (ברמת Acceptance) 8

3 מדריך משתמש 9

3.1 הסבר על מבנה התיקיות: 9

3.2 צד השרת: 9

3.3 צד הלקוח: 9

3.3.1 הפקודות הממומשות לשימושו של הלקוח הן: 9

3.3.2 טבלת משתמשים 10

4 ביבליוגרפיה 11

# רקע הפרוטוקול

## רקע כללי והיסטוריה

SFTP – Simple File Transfer Protocol הינו פרוטוקול פשוט להעברת קבצים מהשרת ללקוח. הפרוטוקול תומך בשליטה של המשתמש, העברת קבצים, יצירת רשימה של תכולת התיקייה , שינוי הנתיב של הקובץ, שינוי שם הקובץ ומחיקתו.

הפרוטוקול ניתן לשימוש רק על ידי חיבור TCP (Lottor 1984).

## סיכום תקן RFC

RPC זה מציע פרוטוקול עבור קהילת האינטרנט ARPA – Advanced Research Projects Agency ודורש דיונים והצעות לשיפור. פרוטוקול זה בא לתת מענה לאנשים אשר מעוניינים בפרוטוקול יותר עוצמתי מפרוטוקול TFTP (זהו פרוטוקול העברת קבצים אשר בולט עקב פשטותו) ופחות עוצמתי ויותר קל למימוש מ-FTP (זהו פרוטוקול העבר קבצים מתקדם יותר).

פרוטוקול זה ניתן למימוש עם באמצעות כל פרוטוקול אמין המבוסס על זרימת 8 בתים. SFTP ניתן לשימוש רק באמצעות חיבור TCP בעוד שTFTP ממומש באמצעות UDP (Lottor 1984).

הפרוטוקול:

יצירת חיבור TCP בין השרת ללקח. שליחת פקודות SFTP והמתנה לתשובה. פקודת SFTP נשלחת לשרת ב-4 אותיות ASCII מלוות ברווח בודד.

פקודות הלקוח:

* USER – תעודת הזהות של המשתמש לצורך אימות של השרת.
* ACCT – שם המשתמש לצורך אימות של השרת.
* PASS – סיסמה של המשתמש לצורך אימות של השרת.
* LIST – נתיב התיקייה
* CDIR – שינוי התיקייה לתיקייה שהועברה בארגומנט
* KILL – מחיקת הקובץ
* DONE – סיום וסגירת החיבור עם השרת
* RETR – בקשה להעברה של קובץ

פורמט הפקודות:

<command> : = <cmd> [<SPACE> <args>] <NULL>

<cmd> : = USER ! ACCT ! PASS ! LIST ! CDIR

KILL ! NAME ! DONE ! RETR !

<response> : = <response-code> [<message>] <NULL>

<response-code> : = + | - | | !

<message> can contain <CRLF>

## סיכום נייר אקדמי - Simple File Transfer Protocol - Debashis Gangully, Darshan Shetty

המאמר מתאר רכיבים ותתי רכיבים של המערכת הסופית שמומשה על מנת לבדוק את פרוטוקול SFTP, כמו כן המאמר משומש כמדריך ברמה גבוהה למימושו.

במאמר מתוארות ברמה הגבוהה אסטרטגיות בדיקות וגישות סטטיסטיות להוכחת נכונות, מדידות ויעילות במונחים מטרים של המערכת שמומשה.

מטרת המאמר הינה להסביר את התכנון, המימוש הבדיקות והניתוח של פרוטוקול SFTP אשר באמצעותו נבצע העברת קבצים בין שתי נקודות קצה של מערכת מבוזרת.

במאמר זה השתמשו בפרוטוקול SFTP באמצעות חיבור UDP בתור בקרה וערוץ המידע. על זה מימשו פרוטוקול Go-back-N וCRC בשביל בקרת שגיאות וזרימה. שמות השרתים משוכפלים ממומשים בשביל חסינות של המערכת וכל שרת יכול להתמודד עם מספר מרובה של לקוחות בו זמנית. לתוך החבילה של המידע הוכנס במכוון שגיאות בביטים וביצעו סימולציה של הסתברות הצפיפות וביטול המנות על מנת ללמוד את הביצועים של ה-FTP.

המערכת הממומשת כוללת רכיבי: שרת שם, שרת קבצים ולקוח SFTP.

שרת השם הינו רכיב אפליקציה שרץ על מערכות מבוזרות ומספק שירותים לשרת הקבצים, שרת הקבצים הינו אפליקציה אשר רצה על מערכות מבוזרות ואחריות לספק שירותים ללקוח, הלקוח הינו רכיב הצרכן של המערכת המבוזרת אשר נותן אפשרות לשימוש בפקודות בשרת הקבצים ומשומש על ידי משתמשי הקצה.

על מנת לממש את הפרוטוקול לקחו את הרכיבים האלו והשתמשו בהם בשכבות מאוחדות על מנת ליצור מערכת שלמה, השכבה העליונה היא ממשק המשתמש אשר משמש את המשתמש להכניס את הפקודות ללקוח הSFTP, הם מימשו זאת כממשק CLI.

ממשק המשתמש חשוף למשתמשי קצה על מנת להכניס פקודות שדרכן תהייה אינטראקציה עם המערכת אשר ממומשת בפרוטוקול SFTP ויוכלו לנווט בארכיוני הקבצים במערכת ולשלוט בהם.

שכבה נוספת הינה מיועדת לשליטה וניהול הקבצים, היא ממומשת אצל הלקוח והשרת, קודם היא מפרשת את הפקודות של המשתמשים ואז מתרגמת אותם לשכבה התחתית כמו כן שכבה זאת אחראית לניהול הקבצים וטיפול בשגיאות של השכבה התחתית.

על מנת לזהות שגיאות הכותבים מימשו CRC על מנת לזהות את השגיאות בנתונים, על מנת לשלוט בזרימה הכותבים מימשו את פרוטוקול Go-back-N מכיוון שהוא דורש פחות משאבים מבחינת נתונים.

השכבה התחתונה הינה של רכיב העברת ערוצים ושליחת נתונים, רכיב זה בונה את הבסיס של פרוטוקול ה-SFTP וממוש בכל המערכות אשר משתתפות בארכיטקטורה המבוזרת, הוא אחראי על העברת השליטה והנתונים ברשת.

כדי להפוך את המערכת לקלה יותר למימוש השתמשו בקבצי הגדרות שהחליפו את הצורך בקידוד-כבד, כמו כן המערכת אשר הם בנו למען מימוש פרוטוקול SFTP דורשת מערכת הפעלה מבוססת UNIX או LINUX, את הקוד עצמו כתבו ב C/C++.

על מנת לבצע ניתוח סטטיסטי יצרו 10 תיקיות שונות שכל אחת מהן מכילה קבצים בגדלים שונים ובדקו את הזמנים שלוקח להעביר את הקבצים בין הגדלים.

בהתחלה בדקו הורדה בביצועים מול ההסתברות לשגיאות, הם גילו שכאשר הסתברות השגיאה בערוץ עולה זמן התגובה של המערכת יורד. כמו כן הם גילו שקבצים קטנים הם לא עקביים ולא תורמים לבדיקות.

לאחר מכן בדקו הורדה בביצועים מול דחיסות הקבצים, גילו שכאשר הדחיסות עולה זמן התגובה יורד, גילו שכאשר יש עלייה בדחיסות זמן התגובה יורד בצורה חדה.

בסופו של דבר בדקו את מימוש הפרוטוקל שלהם עם פרוטוקול Go back N מול Stop and Wait וגילו שברמות נמוכות של שגיאות הם מאוד דומים מבחינת ביצועים אך כאשר סבירות השגיאה עולה Stop and Wait,

לסיכום הם הצליחו לממש את הפרוטוקול וביצעו בדיקות שהראו שהוא עובד טוב אך כאשר אחוז השגיאות עולה יש הורדה בביצועים של המערכת (Debashis Ganguly 2015).

## סיכום נייר אקדמי – Characteristics of Network Traffic Flow Anomalies – Paul Barford, David Plonka

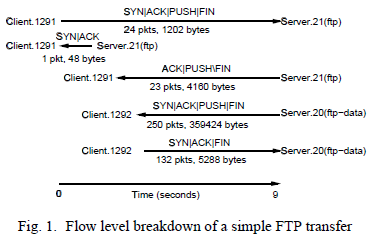
אחת מהמשימות העיקריות של מנהלי רשתות היא לנתר את הנתבים והסוויצ'ים לתנועות חריגות, זיהוי חריגות אלו לרוב נעשה על ידי שיטות Ad-Hoc ושנים של ניסיון, ישנם כלים שעוזרים בתהליך זה אך כדי לעשות זאת דרוש להגדיר הרבה משתנים.

המטרה של החוקרים היא לבחון את החריגות ברשת ולסווג אותם לסוגים שונים כך שיהיה ניתן לקטלג ולהבין אותם על מנת לעשות זאת השתמשו ב-FlowScan כדי לנתח מידע שעובר ברשתות בפרוטוקול.

תכונות של תנועת רשת נחקרה לעומק כבר כמה זמן אך רוב העבודות מתמקדים בדרך בהתנהגות החבילות, במחקרה שנעשה התמקדו בזרימה ואפיון התנהגות חריגה.

FlowScan אוסף נתונים של זרימת רשתות שיוצא על ידי הנתבים של סיסקו וכולל נתונים על מקורות ויעדים, חבילות וספירת בתים, זמני התחלה וסיום של זרימה ומידע של פרוטוקול.

לדוגמא תעבורה יחידה של FTP בפרוטוקול SFTP מיוצגת כמספר זרמי נתונים בין שתי נקודות קצה של שרתים.



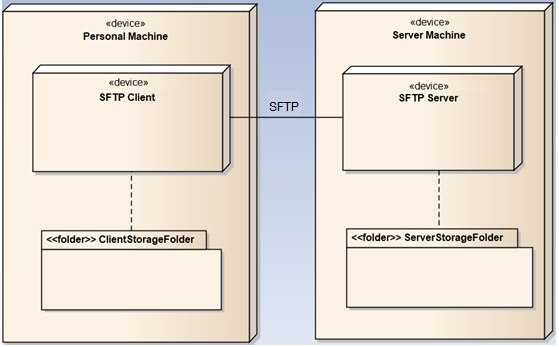
ניתוח ויזואלי של חריגות בזרימה הוביל לסיווג של שלושה קטגוריות כלליות, חריגות של פעולות רשת, חריגות של קהל בזק וחריגות של שימוש לא נאות ברשת כמה שידוע לחוקרים הם הראשונים שקיטלגו סוגים שונים של חריגות זרימה ברשת.

החוקרים הצליחו לקטלג חריגות רשת בזרימה ומקווים בעתיד לזהות בצורה מדויקת יותר קטגוריות נוספות, בזמן הכתיבה של המאמר הם עבדו על ליצור ארכיון מקיף יותר של חריגות בקשת אשר מבוססות על זרימת רשת לפי IP בקמפוס שלהם (Plonka and Barford 2001).

## סיכום רקע כללי

פרוטוקול SFTP הינו פשוט, קל להבנה ולמימוש. מסיבות אלה משתמשים בו לצורכי מחקר של דברים חדשניים, לצורכי בדיקות שונות וביצוע ניתוחים מורכבים. באמצעות הפרוטוקול ניתן לקבל ולהסיק מידע רב לגבי הרשת, לגבי החריגות ולגבי אחוזי המנות שנפלו בדרך. מכיוון שהפרוטוקול ומימושו מובנים מאליו, הפרוטוקול בעצם הופך להיות האמצעי ולא המטרה כלומר משתמשים בו כדי להסיק מידע חדש לגבי הנושא שחוקרים.

# SFTP – Deployment Diagram

****

# הגדרת הניסוי

## בדיקות לקוח/שרת (ברמת Acceptance)

|  |  |  |
| --- | --- | --- |
| הדרישה | המבדק | תוצאה |
| 1. המערכת תקבל כקלט שם קובץ מלא וכתובת שרת SFTP, תתחבר אל השרת ותעביר את הקובץ | 1. התקנת צד השרת במחשב א', צד הלקוח במחשב ב', הרצת שתי האפליקציות, הזנת שם הקובץ וכתובת מחשב ב' במחשב א' ובדיקה שהקובץ אכן הועבר לצד ב'. | הצליח |
| 1. מבדק לעצירה במקרה של כל אחד מהקלטים הבאים באופן לא תקין: 2. User 3. Acct 4. Pass 5. LIST 6. CDIR 7. KILL 8. DONE 9. RETR | 1. המערכת תעצור ותוציא הודעת שגיאה מתאימה לקלט השגוי | הצליח |
| 1. המערכת תקבל כקלט נתיב של תיקייה ותציג למשתמש את כל הקבצים בתיקייה. | 1. המערכת תציג את רשימת הקבצים והתיקיות הנמצאים בנתיב אשר התקבל בהתאם לקלט שהתקבל (list v, list f) . | הצליח |
| 1. המערכת תקבל כקלט נתיב חדש בשימוש בפקודת CDIR ותעבור לנתיב המבוקש בצד של השרת ובדיקת פעולה נאותה של פקודות לאחר המעבר לתיקייה החדשה. | 1. המערכת תציג הודעת אישור על המעבר לתיקייה שהועברה מהלקוח. פקודות שהתקבלו ע"י המערכת לאחר המעבר עובדים על התיקייה החדשה | הצליח |
| 1. המערכת תקבל כקלט קובץ למחיקה. לאחר מכן המשתמש יתנתק מהשרת ויתחבר שוב ויבדוק האם הקובץ קיים במערכת. | 1. המערכת תמחק את הקובץ מהשרת לצמיתות. | הצליח |
| 1. המערכת תקבל כקלט קובץ גדול מעל gigabyte להעברה. ותעביר ללקוח. | 1. התקנת צד השרת במחשב א', צד הלקוח במחשב ב', הרצת שתי האפליקציות, הזנת שם הקובץ ובדיקה שהקובץ אכן הועבר בשלמותו לצד ב'. | הצליח |
| 1. המערכת תקבל כקלט קובץ לשליחה לשני משתמשים בו זמנית, ותבצע העברה לשניהם. | 1. התקנת צד השרת במחשב א', צד הלקוח במחשבים ב', ג'.   הזנת שם הקובץ במחשבים ב', ג' ובדיקה שהקובץ הועבר לשני המחשבים. | הצליח |
| 1. המערכת תקבל כקלט שם קובץ להעברה. שם הקובץ יהיה בעל רווחים ומספרים. | 1. המערכת תבצע העברה של הקובץ אל המשתמש. | הצליח |
| 1. התנתקות/חיבור לקוח מהשרת. | 1. המערכת תסגור/תפתח תהליכון בהתאם לפקודות של המשתמש. | הצליח |

# מדריך משתמש

## הסבר על מבנה התיקיות:

* לשרת וללקוח ישנם תיקיות בשם "ClientFolder" ו-"ServerFolder" בהתאמה, תיקיות אלו נמצאות בתיקיה ראשית בשם "RootFolder".
* תיקיות וקבצים אשר השרת פועל איתם אל הלקוח צריכות להימצא בתוך התיקייה שלו.
* הקבצים שהלקוח מקבל מהשרת יתקבלו אל התיקייה שלו.

## צד השרת:

1. צריך לשים באותה תיקייה את השרת והתיקייה הראשית כאשר בתוכה תיקיית השרת.
2. את קובץ הJar של השרת מריצים באמצעות קובץ ההרצה SERVER\_SFTP Launcher.

## צד הלקוח:

1. את הלקוח והתיקייה הראשית כאשר בתוכה תיקיית הלקוח צריך לשים באותה תיקייה.
2. את קובץ הJar של הלקוח מריצים באמצעות קובץ ההרצה CLIENT\_SFTP Launcher.

### הפקודות הממומשות לשימושו של הלקוח הן:

USER, ACCT, PASS, LIST <F|V>, CDIR, KILL, DONE, RETR

* הפקודות הן לא case-sensitive, וניתן להשתמש בהן בצורה הבאה:
  + USER – על המשתמש להזין את שמו בצירוף הפקודה USER. על מנת לבצע התחברות לשרת. לדוגמה USER user.
  + ACCT – על המשתמש להזין את שם המשתמש בצירוף הפקודה ACCT. על מנת לבצע התחברות לשרת. לדוגמה ACCT acct.
  + על המשתמש להזין את הסיסמה שלו בצירוף הפרודה PASS. על מנת לבצע התחברות לשרת. לדוגמה PASS pass.
  + LIST <F|V> - פקודה זו מחזירה את הרשימה של כל הקבצים והתיקיות בתוך התיקייה המבוקשת. אם לא נבחרה תיקייה ספציפית, התיקייה שתיבחר הינה התיקייה שהשרת מצביע עליה כרגע. LIST V – מאפשר צפייה מורחבת של הקבצים.  
     המאפיינים שיוצגו הינם: שם הקובץ, גודל הקובץ, רמת הגנה- כתיבה, קריאה, ביצוע, מתיי הפעם האחרונה שהקובץ שונה, מי הבעלים. LIST F – מאפשר צפייה מצומצמת בקבצים ובעצם מחזיר רק את שמות הקבצים והתיקיות.  
    לדוגמה LIST V Folder, LIST F Pics.
  + CDIR – פקודה זו משנה את התיקייה שהשרת נמצא בה כרגע לנתיב שמועבר בארגומנט. לדוגמה CDIR Pics. בנוסף הפקודה CDIR ~ תחזיר אותך לתיקייה הראשית.
  + KILL – פקודה זו מוחקת מהשרת את הקובץ שמועבר בארגומנט.  
     לדוגמה KILL text.txt.
  + DONE – פקודה זו תסיים את החיבור עם השרת ובעצם תבצע התנתקות.
  + RETR – פקודה זו תגרום לשרת להתחיל לבצע העברה של קובץ ממנו אל המשתמש. לדוגמה RETR text.txt.

### טבלת משתמשים

|  |  |  |
| --- | --- | --- |
| PASS | ACCT | USER |
| "" | "" | "admin" |
| "pass" | "acct" | "user" |

# ביבליוגרפיה

Debashis Ganguly, Darshan Shetty. "Simple File Transfer Protocol." (University of Pittsburgh) 2015.

Lottor, Mark K. "Simple File Transfer Protocol." *IETF Drafts* (Internet Engineering Task Force), 1984.

Plonka, Paul, and David Barford. "Characteristics of Network Traffic Flow Anomalies." 2001.