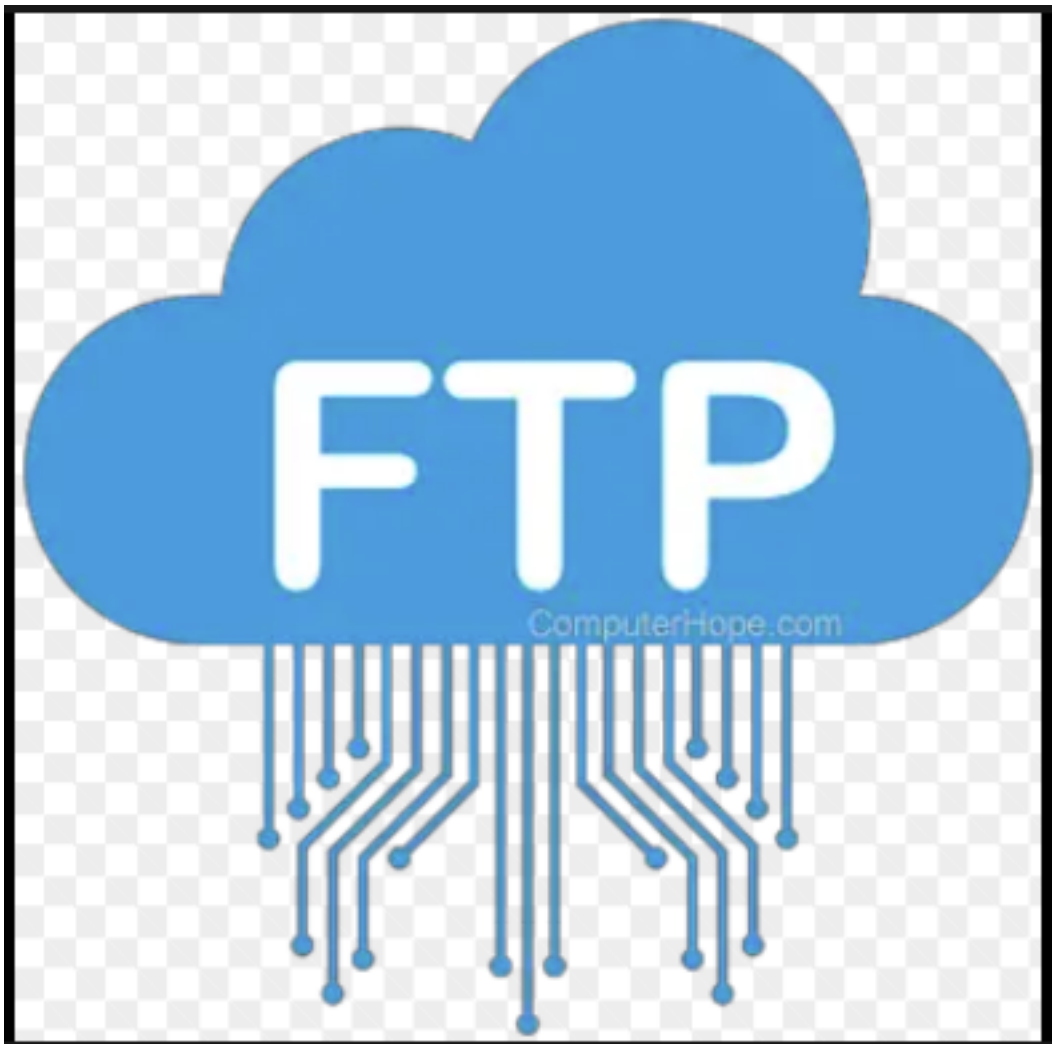
בס"ד

מגישים:
ניסים עטייה 207302027
אליה שלמה 205915663



בס"ד

מטרת הפרויקט

במטלה זו התבקשנו לבנות מערכת תקשורת בין קליינט לשרת dns לשרת dhcp ולשרת האפליקציה.

האפליקציה שאנו בחרנו היא אפליקציה ftp , כל לקוח שיתחבר לשרת זה יוכל להעלות קבצים אליו ולהוריד קבצים ממנו ולראות אילו קבצים יש בשרת זה.

Dynamic Host Configuration Protocol (DHCP(

פרוטוקול רשת המשמש להקצאה אוטומטית של כתובות ip ומידע אחר על תצורות רשת להתקנים ברשת.
dhcp מבטל את הצורך בהקצאה ידנית של כתובות ip ומאפשר למכשירים להתחבר בקלות לרשת ולתקשר עם מכשרים אחרים.

כאשר מכשיר מתחבר לשרת הdhcp הוא שולח הודעת שידור המבקשת כתובת ip, שרת הdhcp מקבל את הבקשה ומגיב עם כתובת ip זמינה ומידע אחר על תצורת רשת כגון מסיכת רשת וכתבות שרת ה dns ועוד..

dhcp מאפשר גם השכרה של כתבות ip, משמע שכתובת ip אינה מוקצת לצמיתות למכשיר, אלא מוקצת באופן זמני לתקופת זמן מוגדרת ולאחר זמן זה המכשיר צריך לבקש כתובת ip חדשה.

ניתן לחלק את תהליך תקשורת ה-DHCP בין השרת ללקוח לארבעה שלבים עיקריים: discover , offer request and ack.

Dynamic Host Configuration Protocol (DHCP) הוא פרוטוקול ניהול רשת המשמש להקצאה וניהול אוטומטי של כתובות IP , מסכות רשת משנה, שער ברירת מחדל ופרמטרים אחרים של תצורת רשת להתקנים ברשת.

DHCP מפשט את הניהול של כתובות IP ברשת. במקום להקצות ידנית כתובת IP לכל מכשיר ברשת, שרת DHCP יכול להקצות ולנהל באופן אוטומטי כתובות IP עבור כל המכשירים. זה חוסך זמן ומפחית את הסבירות לשגיאות בעת הגדרה וניהול של רשתות.

Discover:
כאשר לקוח מחובר לראשונה לרשת, הוא שולח הודעת DHCP Discover דרך כתובת שידור 255.255.255.255 כדי לבקש כתובת IP. הודעת Discover כוללת את כתובת ה-MAC של הלקוח יחד עם מידע נוסף כגון מסיכת רשת ועוד. אם ההודעה היא DHCP Discover, שרת ה-DHCP יפיק חבילת DHCP Offer המכילה את כתובת ה-IP שהשרת רוצה להקצות ללקוח. חבילה זו נבנית באמצעות ספריית ה-scapy, וכוללת את השדה yiaddr בשכבת BOOTP, המפרטת את כתובת ה-IP שהשרת רוצה להציע ללקוח

offer:
שרת ה-DHCP מקבל את הודעת Discover ומגיב בהודעת הצעה של DHCP, הכוללת כתובת IP זמינה שניתן להקצות ללקוח. הודעת offer כוללת גם את זמן החכירה (כמה זמן הלקוח יכול להשתמש

בכתובת ה-IP שהוקצתה), וכתובות שרת DNS. הודעת offer נשלחת לכתובת ה-MAC של הלקוח.
בס"ד

לאחר קבלת הודעת DHCP Discover מהלקוח, שרת ה-DHCP יוצר הודעת DHCP Offer כדי להציע תצורה ללקוח. ההצעה מכילה את כתובת ה-IP של השרת, מסיכת רשת המשנה, זמן החכירה של כתובת ה-IP ואפשרויות תצורה אחרות שהשרת יכול להציע ללקוח.

<u>request</u>:
עם קבלת הודעת offer, הלקוח שולח הודעת DHCP Request לשרת כדי לבקש רשמית את כתובת ה-IP המוצעת. הודעת הבקשה מאשרת גם את פרטי תצורת הרשת האחרים שהתקבלו בהודעת offer .
הודעת DHCP ACK דומה להודעת DHCP request , אך היא מכילה מידע נוסף, כגון משך החכירה וכתובת שרת ה-DNS , המוסכם בין הלקוח והשרת במהלך תהליך הבקשה. הודעת DHCP ACK נשלחת גם כהודעת שידור לרשת, אך היא נשלחת ישירות ללקוח המבקש במקום לכל הלקוחות

<u>ack</u>:
אם כתובת ה-IP המוצעת עדיין זמינה, השרת מגיב בהודעת DHCP Acknowledge (או Ack). הודעת ה-Ack מאשרת שכתובת ה-IP הוקצתה ללקוח ומספקת את זמן החכירה ומידע אחר על תצורת הרשת.
ברגע שהלקוח מקבל את הודעת ה-Ack , הוא מגדיר את הגדרות הרשת שלו ומתחיל להשתמש בכתובת ה-IP שהוקצתה. ACK משמש בפרוטוקולי רשת רבים, כולל TCP (פרוטוקול בקרת שידור) ו-UDP (פרוטוקול משתמש Datagram). ב-TCP, ACKs משמשים כדי להבטיח משלוח אמין של מנות. כאשר שולח שולח חבילה למקלט, המקבל ישלח ACK בחזרה לשולח כדי לאשר שהחבילה התקבלה. אם השולח לא יקבל ACK בתוך מרווח זמן מוגדר, הוא יניח שהחבילה אבדה וישלח אותה מחדש. תהליך זה חוזר על עצמו עד שהשולח מקבל ACK מהמקלט.

<u>Domain Name System -DNS</u>
פרוטוקול המשמש לתרגום שמות דומיינים הניתנים לקריאה על ידי אדם (כגון www.example.com) לכתובות IP הניתנות לקריאה במחשב (כגון 192.168.0.1). DNS הוא חלק חיוני מתשתית האינטרנט ומאפשר למשתמשים לגשת לאתרי אינטרנט ולמשאבי רשת אחרים באמצעות שמות דומיין שקל לזכור במקום לזכור כתובות IP מספריות.

ניתן לחלק את תהליך תקשורת ה-dns למספר שלבים:

1. כאשר משתמש מזין שם דומיין בדפדפן האינטרנט שלו הדפדפן שולח שאילתת dns, השאילתה כוללת את שם הדומיין שאליו המתממש רוצה לגשת.
2. אם לשרת ה-dns אין את כתובת ה-ip של שם הדומיין המבוקש במטמון שלו הוא מבצע שאילתה רקורסיבית כדי להשיג את כתובת ה-ip.
שאילתה רקורסיבית פירושה שהשרת מתחיל בשרתי ה-dns הבסיסיים ופועל במורד היררכיית ה-dns כדי לאתר את שרת ה-dns הסמכותי עבור שם הדומיין המבוקש.
3. שרתי שורש: שרתי ה-dns הבסיסיים הם רשת גלובלית של שרתים השומרת מידע על הדומיינים ברמה העליונה, ושרתי ה-dns המוסמכים המשויכים אליהם.
השאילתה הרקורסיבית מתחילה באחד משרתי השורש ומבקשת את כתבות ה-ip עבור הדומיין המבוקש.
4. שרתי TLD: ברגע ששרת השורש מגיב עם כתובת הip של שרת ה-dns הסמכותי עבור התחום המבוקש ברמה העליונה, השאילתה הרקורסיבית נשלחת לשרת ה(TLD (top level domain

בס"ד

שרת הtld מגיב עם כתובת ה-ip של שרת ה-dns הסמכותי לשלב הבא למטה בהיררכיית הdns

5. שרת סמכותי: השאילתה הרקורסיבית ממשיכה במורד היררכיית הdns עד שהיא מגיעה לשרת ה dns הסמכותי עבור שם הדומיין המבוקש. השרת הסמכותי מגיב עם כתובת הip המבוקשת לשם הדומיין המבוקש.
6. תגובת dns: שרת הdns מקבל את כתבות הip מהשרת הסמכותי ומחזיר אותה למשתמש מה שמאפשר למשתמש ליצור חיבור לשרת הדומיין המבוקש.

FTP- file transfer protocol

פרוטוקול סטנדרטי המשמש להעברת קבצים דרך האינטרנט. FTP מאפשר למשתמשים להעביר קבצים בין שתי מערכות מרוחקות, בדרך כלל לקוח ושרת. ל-FTP יש כמה פרצות אבטחה, כמו שליחת אישורי התחברות בטקסט רגיל, שיכולים להיות יורטים על ידי תוקפים, ואפשרות למשתמשים אנונימיים להעלות קבצים לשרת, מה שעלול להוביל להעלאת תוכנות זדוניות או תוכן זדוני אחר לשרת. מסיבה זו, חלופות כמו SFTP (Secure File Transfer Protocol) או FTPS (FTP over SSL) משמשות לעתים קרובות להעברת קבצים מאובטחת.

חיבור FTP:

השלב הראשון הוא ליצור חיבור בין לקוח ה-FTP לשרת ה-FTP. הלקוח שולח בקשה להתחבר לשרת באמצעות TCP/IP, הכוללת את כתובת ה-IP ומספר היציאה של השרת.

אימות:

ברגע שהלקוח והשרת מחוברים, הלקוח חייב לבצע אימות עם השרת על ידי מתן שם משתמש וסיסמה. אם האישורים נכונים, השרת מעניק גישה ללקוח ומאפשר לו להתחיל בהעברת קבצים.

העברת קבצים:

לאחר האימות, הלקוח יכול לשלוח פקודות לשרת כדי להעלות או להוריד קבצים.
FTP תומך בשני מצבים של העברת קבצים: ASCII ובינארי. מצב ASCII משמש עבור קבצי טקסט, בעוד שמצב בינארי משמש עבור קבצים שאינם טקסט, כגון תמונות או קובצי הפעלה.

פקודות:

FTP כולל קבוצה של פקודות שהלקוח יכול להשתמש בהן כדי ליצור אינטראקציה עם השרת. חלק מהפקודות הנפוצות ביותר כוללות:

GET: מוריד קובץ מהשרת ללקוח
PUT: מעלה קובץ מהלקוח לשרת
LS: מפרט את התוכן של הספרייה הנוכחית בשרת

סיום חיבור: לאחר השלמת העברת הקבצים, הלקוח יכול לסגור את החיבור לשרת.
הלקוח שולח בקשה לסגור את החיבור,והשרת מגיב באישור.
ראוי לציין שניתן לאבטח את ה-ftp באמצעות הצפנה, כדי להגן על העברת מידע רגיש כגון סיסמאות וקבצים. בנוסף לftp יש מספר מגבלות כמו היותו פגיע להתקפות כגון האזנה או חבלה בנתונים במהלך

העברת קבצים והוא אינו מתאים להעברת קצבים גדולים, או העברות תכופות של קבצים קטנים רבים.

בס"ד

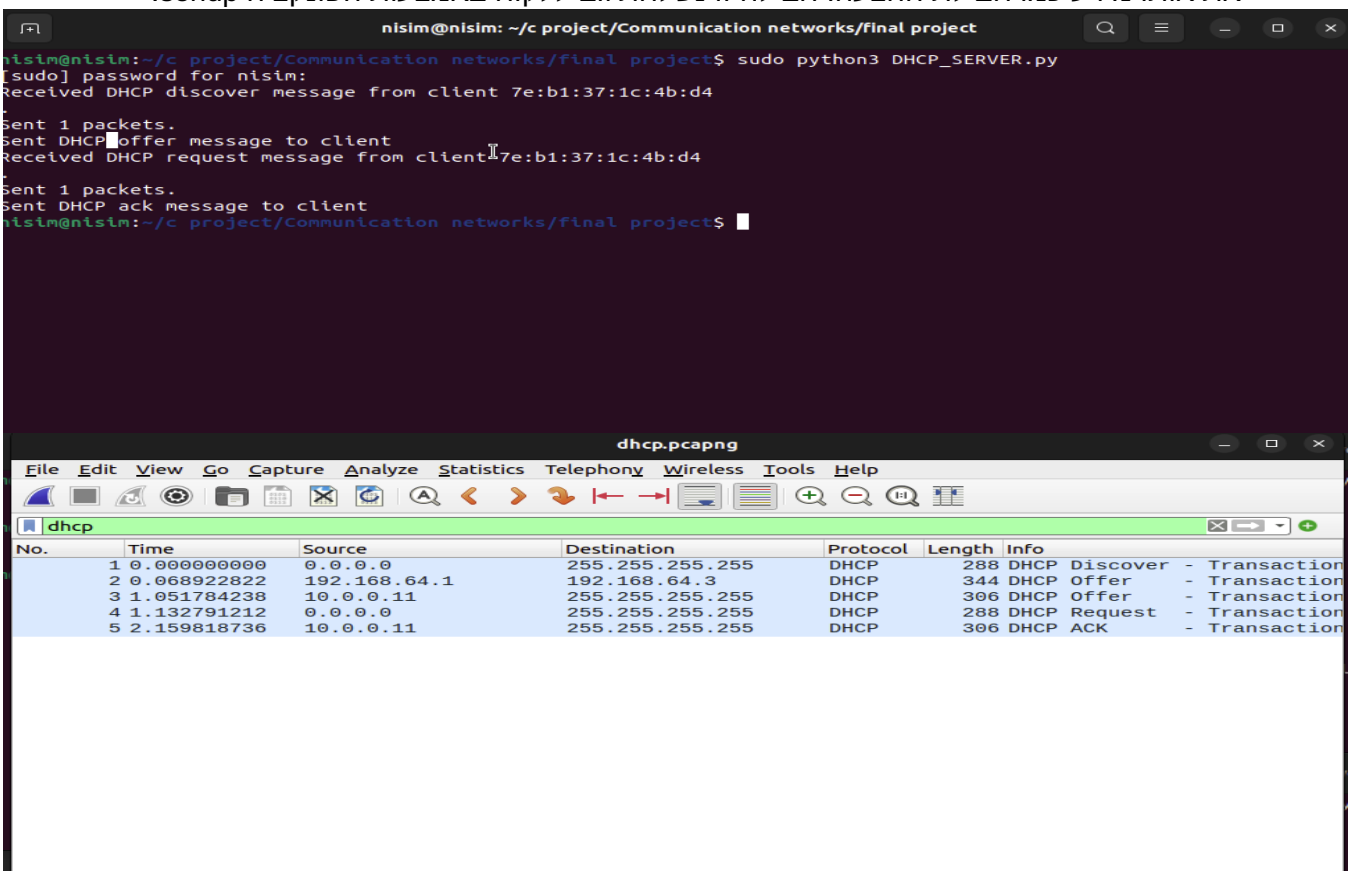
הקוד שלנו:

dhcp:

בקוד שלנו מימשנו שרת dhcp באמצעות ספריית scapy בפיתון.
כאשר התוכנית רצה היא מקשיבה לבקשות dhcp בממשק רשת enp0s1 עם פילטר של udp בפורט 67/68 . כאשר בקשה נתפסת היא נשלחת לפונקציה handle_dhcp_request.

הפונקציה handle_dhcp_request נקראת בכל פעם שנתפסת חבילה מתאימה לפי הפילטר שהגדרנו, אם ההודעה היא הודעת "DHCP discover", השרת שולח חבילת "DHCP offer" ללקוח, ומציעה כתובת IP. המשתנה dhcp_offer בקוד מכיל את המבנה של חבילת ההצעה, הכוללת את כתובת ה-IP של שרת ה-DHCP , מסיכת רשת המשנה, כתובת הנתב, זמן החכירה וכתובת שרת ה-DNS. חבילה זו נשלחת לאחר מכן ללקוח באמצעות הפונקציה sendp.

אם ההודעה היא הודעת "DHCP request", השרת שולח חבילת "DHCP ack" ללקוח, המאשרת את הקצאת החכירה של כתובת ה-IP. המשתנה dhcp_ack בקוד מכיל את המבנה של חבילת ה-ack, הכוללת את אותו מידע כמו חבילת ההצעה. חבילה זו נשלחת גם ללקוח באמצעות הפונקציה sendp.



בתחילה הרצנו את שרת הdhcp ולאחר מכן הרצנו את תוכנית הקליינט. ברגע שהרצנו את תוכנית הקליינט, נוכל לראות בתמונה הראשונה שהשרת קיבל את הודעת הdiscover של הקליינט והדפיס את כתובת הmac של הקליינט, שזוהי כתובת הmac של המחשב שלי והיא אכן 7e:b1:37:1c:4b:d4 בס"ד

נוכל לראות בנוסף שהבקשה הזאת אכן נתפסה בווירשארק ואכן הsource הינו 0.0.0.0 והdest הוא 255.255.255.255 וזו אכן הודעת dhcp discover. לאחר מכן נוכל לראות בתמונה הראשונה שהשרת שלח הודעת offer ואכן נוכל לראות בווירשארק שהשרת שכתובת הip שלו היא 10.0.0.11 שלח הודעה מסוג dhcp discovery בברודקאוסט.
נוכל לראות בווירשארק גם שהdhcp של הבית שלי קיבל את ההודעה שהלקוח שלח ושלח לו הודעת dhcp offer גם כן מפני שהווירשארק תפס עוד הודעת dhcp offer מכתובת 192.168.64.1.
לאחר מכן נוכל לראות בתמונה הראשונה שהשרת תפס הודעת dhcp request והדפיס את כתובת הmac של הלקוח ושלח dhcp ack, ואכן נוכל לראות בווירשארק שנשלחה dhcp request מכתבות 0.0.0.0 בברוקדאסט ואכן נשלחה הודעה מסוג dhcp ack מip של הdhcp בברדוקסט.

<u>dns</u>

קוד זה מיישם שרת DNS בסיסי שיירט בקשות DNS, מחפש את שם הדומיין המבוקש במטמון שלו, ומגיב עם כתובת ה-IP המתאימה אם הוא נמצא במטמון. אם שם הדומיין המבוקש לא נמצא במטמון, שרת ה-DNS שולח בקשה לשרת ה-DNS של גוגל לקבל את כתובת ה-IP, מאחסן אותה במטמון ומגיב ללקוח עם כתובת ה-IP.

הקוד משתמשת בספריית scapy, הוא מגדיר את ממשק הרשת (IFACE) ומילון שיכיל את מיפויי ה-DNS (dns_cache). המטמון מאחסן את כתובת ה-IP של שם דומיין שנשאל בעבר כדי למנוע שליחת בקשה לשרת ה-DNS של Google בכל פעם שלקוח מבקש את אותו שם דומיין.

הקוד מגדיר פונקציה (handle_dns_request) המטפלת בבקשות DNS נכנסות. הפונקציה נקראת על ידי הפונקציה sniff אשר מאזינה למנות UDP ביציאה 53 (יציאת ה-DNS הרגילה). כאשר מתקבלת חבילת בקשת DNS, הפונקציה handle_dns_request נקראת לעבד אותה.

הפונקציה handle_dns_request () מחלצת תחילה את שם הדומיין המבוקש משאילתת ה-DNS ובודקת אם כתובת ה-IP של שם הדומיין קיימת במטמון. אם כתובת ה-IP נמצאת במטמון, הפונקציה יוצרת חבילת תגובה של DNS ושולחת אותה בחזרה ללקוח עם כתובת ה-IP. אם כתובת ה-IP לא נמצאה במטמון, הפונקציה שולחת בקשה לשרת ה-DNS של גוגל לקבל את כתובת ה-IP, מאחסנת אותה במטמון ומגיבה ללקוח עם כתובת ה-IP.

הקוד שולח את חבילת תגובת ה-DNS בחזרה ללקוח באמצעות הפונקציה sendp() של Scapy. חבילת התגובה נבנית על ידי העתקת כותרות ה-Ethernet, IP ו-UDP מחבילת בקשת ה-DNS הנכנסת והוספת כותרת תגובה DNS עם כתובת ה-IP המתאימה.

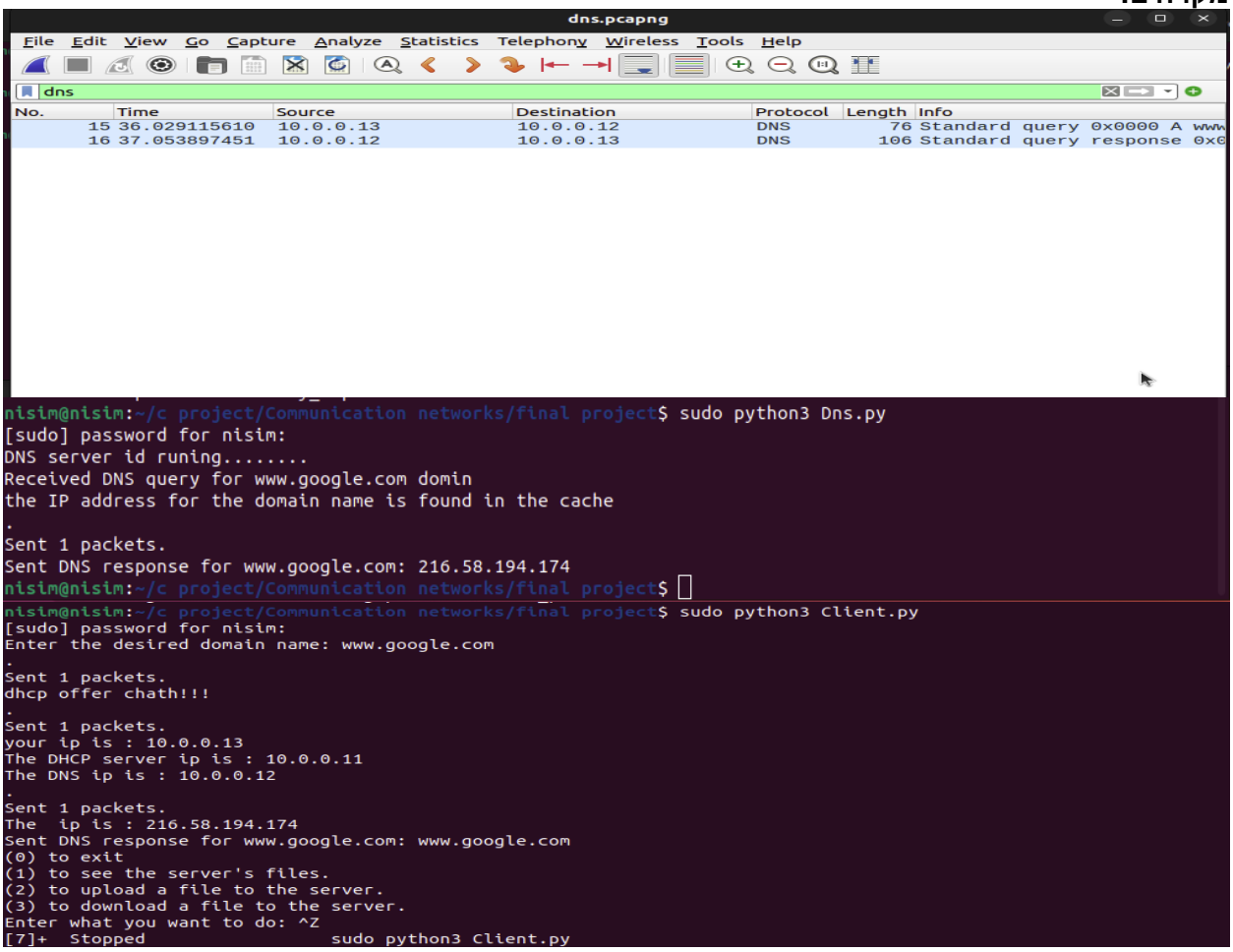
לבסוף, הקוד מפעיל את שרת ה-DNS על ידי קריאה לפונקציה sniff() עם המסנן "udp port 53" כדי להאזין לבקשות DNS ביציאה 53. הארגומנט prn מוגדר ל-handle_dns_request כדי לציין את פונקציית ה-callback שתיקרא עבור כל DNS חבילת בקשה התקבלה. ארגומנט הספירה מוגדר ל-1 כדי להגביל את מספר החבילות לעיבוד. ברגע שמתקבלת בקשת DNS, השרת מגיב עם כתובת ה-IP המתאימה ואז יוצא.

DNS* (מערכת שמות דומיין) היא מערכת שמות היררכית ומפוזרת, הממפה שמות דומיינים הניתנים לקריאה על ידי אדם לכתובות IP . במילים פשוטות יותר, זה כמו ספר הטלפונים של האינטרנט. כאשר אתה מקליד כתובת (Uniform Resource Locator) URL בדפדפן אינטרנט, הדפדפן שולח בקשה לשרת

בס"ד

ה-DNS לחפש את כתובת ה-IP המשויכת לאותה כתובת URL. לאחר פתרון כתובת ה-IP, הדפדפן שולח בקשה לשרת המארח את האתר עם כתובת ה-IP הזו, והאתר מוצג בדפדפן שלך.

מקרה 1:

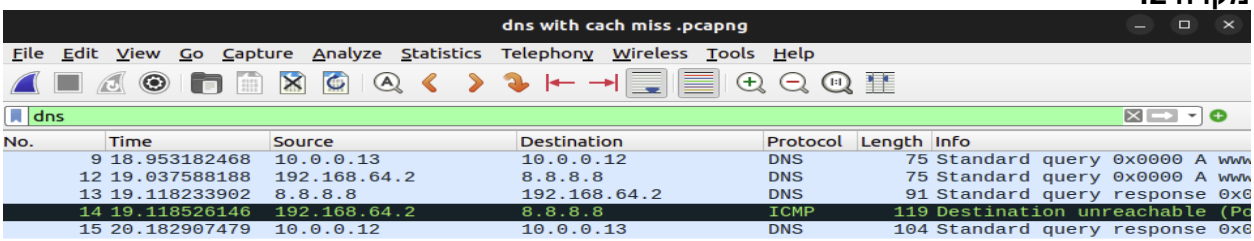


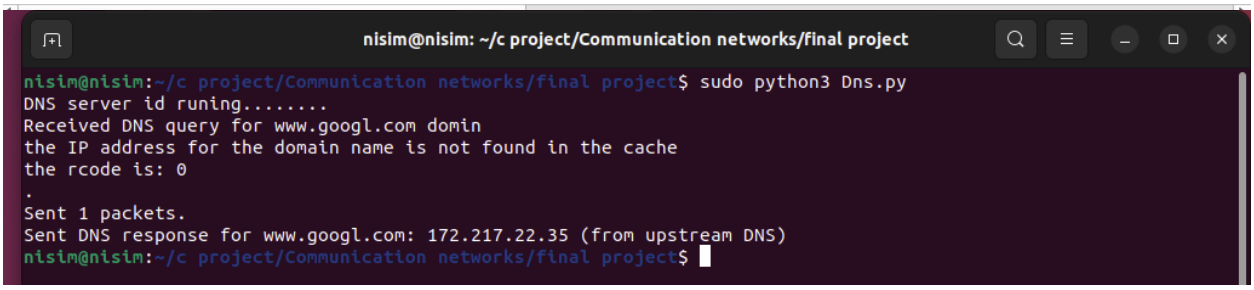
בתחילה הרצנו את התוכנית של השרת dns ולאחר מכן הרצנו את התוכנית של שרת הקליינט.
תחילה הקליינט מתבקש להכניס את שם הדומיין אליו הוא רוצה להתחבר.
לאחר מכן כמו שפירטנו לעיל הוא מקבל קונפיגורציה משמרת הdhcp,ורק לאחר מכן שולח dns request לשרת הdns עם שם הדומיין אותו הכניס הלקוח בתחילה.
נוכל לראות בתמונה השניה ששרת הdns רץ ומחכה להודעות וברגע שהוא תפס הודעה הוא מדפיס שהוא תפס הודעה, ואת שם הדומיין אותו הלקוח מחפש.

הוא מדפיס בנוסף מידע אשר אומר שהדומיין נמצא בcach שלו ושולח את כתובת הip של הדומיין, נוכל לראות בתמונה השלישית שהלקוח מכניס את שם הדומיין www.google.com ומקבל קונפיגורציה מה dhcp ומדפיס את כתובת הip שהוא קיבל- של שרת הdhcp ואת כתובת הip ל שרת הdns. לאחר מכן שולח הודעת dns query עם שם הדומיין שמצוין לעיל ומקבל את כתובת הip של שם הדומיין הזה ומדפיס אותה.
בס"ד

נוכל לראות בתמונה הראשונה מהויירשארק שנשלח הודעת dns query מכתובת 10.0.0.13 שזוהי כתובת הip של הלקוח אל כתובת 10.0.0.12 שזה כתבות הip של שרת הdns.
ונשלחת הודעה חזרה מהכתבות ip של הdns אל כתובת הip של הלקוח וזאתי הודעת הresponse.

מקרה 2:





בדומה למקרה 1 בתחילה הרצנו את התוכנית של השרת dns ולאחר מכן הרצנו את התוכנית של שרת הקליינט.
בס"ד

תחילה הקליינט מתבקש להכניס את שם הדומיין אליו הוא רוצה להתחבר. לאחר מכן כמו שפירטנו לעיל הוא מקבל קונפיגורציה משמרת הdhcp ורק לאחר מכן שולח dns request לשרת הdns עם שם הדומיין אותו הכניס הלקוח בתחילה.
נוכל לראות בתמונה השנייה ששרת הdns רץ ומחכה להודעות וברגע שהוא תפס הודעה הוא מדפיס שהוא תפס הודעה, ואת שם הדומיין אותו הלקוח מחפש.

הוא מדפיס בנוסף מידע אשר אומר שהדומיין לא נמצא בcach שלו ושולח את שאילתה לdns של גוגל בכתובת 8.8.8.8 עם שם הדומיין שהלקוח שאל לגביו, נוכל לראות בתמונה השלישית שהלקוח מכניס את שם הדומיין www.googl.com ומקבל קונפיגורציה מהdhcp ומדפיס את כתובת הip שהוא קיבל של שרת

הdhcp ואת כתובת הip ל שרת הdns. לאחר מכן שולח הודעת dns query עם שם הדומיין שמצוין לעיל ומקבל את כתובת הip של שם הדומיין הזה ומדפיס אותה.
נוכל לראות בתמונה הראשונה מהויירשארק שנשלח הודעת dns query מכתובת 10.0.0.13 שזוהי כתובת הip של הלקוח אל כתובת 10.0.0.12 שזה כתבות הip של שרת הdns.
לאחר מכן משום שלdns אין את שם הדומיין במטמון נוכל לראות בויירשארק שהמחשב שולח dns query לשרת הdns של גוגל, שרת הdns של גוגל מחזיר תשובה עם כתובת הip של שם הדומיין הנדרש, ולאחר מכן נשלחת הודעה מכתבות של שרת הdns שלנו אל הip של הלקוח.
ונשלחת הודעה חזרה מהכתבות ip של הdns אל כתובת הip של הלקוח וזאתי הודעת הresponse.

:my_ftp
לקוד של שרת זה יש שלושה פונקציות עיקריות: put,get,ls
הספריות המיובאות בתחילת הקוד הן time,scapy,pwd.
לאחר יבוא הספריות הדרושות מוגדרים משתנים קבועים כגון: כתבות הmac, כתבות הip של השרת ועוד..
עם הרצת הקוד יודפס למשתמש שהmy_ftp server מחובר.
לאחר מכן בעזרת פונקציית sniff נסניף פקאטות udp שהsrc_port שלהן הוא 30663.

כל פקאטה שתוסנף על ידי הפילטר המתאים תיכנס לתוך פונקציה extract_wahttodo.
בפוקנציה זאת נדפיס שהסקאטה נתפסה, נחלץ ממנה את מה שהמשתמש רוצה לעשות בשרת לדוג:
האם הוא מעוניין לעלות קבצים לשרת, להוריד קבצים מהשרת או שמה לראות אילו קבצים יש לשרת להציע.
לאחר מכן יודפס הודעה מתאימה אשר אומרת מה המשתמש בחר לעשות.
אם המשתמש בחר לעלות קבצים לשרת נשלח למשתמש פקאטה request המכילה ack, ולאחר מכן תרוץ הפונקציה put.
אם המשתמש בחר להוריד קבצים לשרת נשלח למשתמש פקאטה request המכילה ack, ולאחר מכן תרוץ הפונקציה get.
אם המשתמש בחר לראות אילו קבצים יש לשרת להציע תישלח למשתמש פקאטה request המכילה ack , ולאחר מכן תרוץ הפונקציה ls.

בס"ד

-()Put

הפונקציה תסניף על ידי הפילטר המתאים המפורט לעיל פקאטה ראשונה אשר תכיל את שם הקבוץ אותו המשתמש ירצה להעלות לשרת, בנוסף לזה היא תסניף פקאטה נוספתץ על ידי הפילטר המתאים אשר תכיל א גודל הקובץ ולאחר מכן יודפס הודעה מתאימה: שם הקובץ שהמתמש רוצה להעלות הוא... וגודל הקובץ הוא... בבתים.
לאחר מכן הפונקציה תסניף את תוכן הקובץ, ותשמור אותו במשתנה file_data.
לאחר מכן הסונצקיה תיפתח קובץ בשם אשר קיבלה מהפקאטה הראשונה בתוך התיקייה של השרת, תכתוב את file_data לתוך הקובץ ותסגור את הקובץ.
לאחר מכן הפוקנציה תשנה את permissions הקובץ ל 0o777 אשר יאפשר לכתוב ולקרוא מהקובץ.
בנוסף לכך הפונקציה תשנה את הowner של הקובץ לבעל השרת, ותדפיס הודעה אשר תגיד שהקובץ..
הפונקציה put () היא שיטה שאחראית לטיפול בהעלאות קבצים לשרת. זה נקרא כאשר הלקוח שולח פקודת "put " לשרת. מטרת שיטה זו היא לקבל את הקובץ שנשלח על ידי הלקוח, לשמור אותו במערכת הקבצים של השרת, לשנות את ההרשאות שלו ולהודיע ללקוח שהקובץ התקבל בהצלחה

get()

הפקונציה תסניף על ידי פילטר מתאים המפורט לעיל פקאטה ראשונה אשר תכיל את שם הקובץ אותו המשתמש רוצה להוריד מהשרת ותחלץ מהפקאטה את השם ותשמור תו במשתנה בשם name.
לאחר מכן בלולאת while כל עוד הפונקתציה לא קיימת בשרת תישלח למשתמש פקאטה בשם ack_nack
אשר תכיל nack - מה שיצביע למתשמש שהשרת לא מאשר הורדה כזאת משום לשרת אין קובץ כזה להציע, והפוקנציה תסניף ותצפה שוב פעם לפקאטה אשר תכיל את שם הקובץ אותו המשתמש ירצה להוריד מהשרת ותחלץ את שם הקובץ מהפקאטה למשתמש בשם name וכך תבצע שוב ושוב עד אשר המשתמש יכיס שם של קובץ תקין.
לאח שהמשתמש הכניס שם של קובץ תקין הפונקציה תבנה שוב פקאטה בשם ack_nack אך הפעם היא תכיל את המילה ack מה שיצביע למשתמש שהשרת מאשר הורדה, וידי למתשמש להיות מוכן לכך שהשרת מתחיל לשלוח את תוכן הקובץ.
הפוקנציה תפתח את הקובץ, תקרא את תוכן הקובץ לתוך משתנה בשם file_data ותסגור אותו.
היא תבנה פקאטה אשר תכיל את תוכן הקובץ ותשלח אותו, תדפיס הודעה מתאימה אשר תעדכן שהקובץ הורד בהצלחה.

הפונקציה get() משמשת להורדת קובץ משרת ה-FTP. תחילה הוא מרחרח את הרשת אחר חבילה המכילה את שם הקובץ להורדה. שם הקובץ חולץ מהמטען של החבילה, והפונקציה בודקת אם הקובץ קיים בספרייה המקומית.

ls()

הפוקנציה משתמשת בספריית os ובתוך סיפריה זו בפונקציה listdir אשר מקבלת את הpath היכן שמורים כל הקבצים של השרת, ומחזירה הlist של כל הקבצים.
את הlist הזה נשמור במשתנה file_name.
לאחר מכן נמיין את הקבצים על ידי פונקציה sorted אשר מקבלת את הlist של כל הקבצים ונשמור את הקבצים הממויינים במשתנה בשם sorted_files_names.
לאחר מכן על ידי פוקנציית join המקבלת את הlist של הקבצים הממויינים יוצרת מחרוזת על ידי צירוף כל האלמנטים ברשימה שנקראת sorted_file_names, עם ' ,' כמפריד בין כל אלמנט.
לאחר מכן הפונקציה תיצור פקאטה אשר תכיל את המחזורת של שמות הקבצים הממויינים לפי סדר הא"ב שיש לשרת להציע ותשלח למשתמש ותדפסי הודעה מתאימה אשר תדעכשהקבצים שנלחו להצגה למשתמש בהצלחה.

בס"ד

client

בתחילת הקוד, הספריות הדרושות מיובאות, הספריות שבהן נעשה שימוש הן os-ו pwd, scapy, time. ספריות אלו חיוניות לביצוע פעולות רשת שונות, כגון שליחה וקבלה של מנות.

הסקריפט מגדיר מספר משתנים גלובליים, כגון IFACE, CLIENP_IP, DNS_IP, DHCP_IP, AP_IP, domain_name ו-mac . משתנים אלו מכילים מידע חשוב המשמש לאורך הסקריפט.
IFACE מאחסן את שם ממשק הרשת, CLIENP_IP מאחסן את כתובת ה-IP של הלקוח, DNS_IP מאחסן את כתובת ה-IP של שרת ה-IP_DHCP , DNS מאחסן את כתובת ה-IP של שרת ה-IP_AP, DHCP מאחסן את כתובת ה-IP של FTP_MY, domain_name, ו-mac מאחסן את כתובת ה-MAC.

הסקריפט מגדיר שתי פונקציות, get_ips ו-get_ip_domain. הפונקציה get_ips נקראת כאשר הסקריפט מקבל DHCP offer משרת ה-DHCP.
הפונקציה מחלצת את כתובות ה-IP של שרתי הלקוח, ה-DNS וה-DHCP מחבילת offer של DHCP ומאחסנת אותם במשתנים הגלובליים CLIENP_IP, DNS_IP ו-DHCP_IP, בהתאמה.
get_ip_domain נקראת כאשר הסקריפט מקבל DNS response משרת ה-DNS.
הפונקציה מחלצת את כתובת ה-IP של הדומיין מחבילת התגובה של ה-DNS ומאחסנת אותה במשתנה הגלובלי AP_IP.

לאחר מכן, הסקריפט מגדיר חבילת dhcp discover
חבילה זו משמשת לגילוי שרת ה-DHCP ברשת. הסקריפט שולח את חבילת הגילוי של DHCP באמצעות sendp.

לאחר מכן, הסקריפט משתמש בפונקציה sniff כדי להמתין ל DHCPoffer משרת ה-DHCP.
הפונקציה sniff משמשת ללכידת מנות רשת התואמות למסנן ספציפי.
במקרה זה, המסנן המשמש הוא "udp port 67". כאשר הסקריפט מקבל תגובת DHCP offer משרת ה-DHCP, הוא קורא לפונקציה get_ip כדי לחלץ את כתובות ה-IP של הלקוח, כתובת ה ip של שרתי ה-DNS וה-DHCP מחבילת offer של DHCP.

לאחר מכן, הסקריפט מגדיר חבילת DHCP recuest.
חבילה זו משמשת לבקשת כתובת IP משרת ה-DHCP. הסקריפט שולח את חבילת הבקשות של DHCP באמצעות sendp.
לאחר מכן, ממתין לתגובת DHCP ack משרת ה-DHCP באמצעות פונקציית sniff עם מסנן " udp port 67" . כאשר הסקריפט מקבל תגובת אישור DHCP, הוא מדפיס את כתובות ה-IP של שרתי הלקוח, ה-DNS וה-DHCP.

לאחר מכן, הסקריפט מגדיר חבילת query
חבילה זו משמשת לשאילתה בשרת ה-DNS עבור כתובת ההדומיין
לאחר מכן, הסקריפט משתמש בפונקציה sniff כדי להמתין ל dns response משרת ה-dns.
הפונקציה sniff משמשת ללכידת מנות רשת התואמות למסנן ספציפי.
במקרה זה, המסנן המשמש הוא "udp and port 5353" כאשר הסקריפט מקבל תגובת dns response משרת ה-DHCP, הוא קורא לפונקציה get_ip_domain כדי לחלץ את כתובות ה-IP של הדומיין.

בס"ד

הקוד ממשיך בהגדרת port המקור והיעד, ולאחר מכן מציג תפריט של אפשרויות לבחירת המשתמש. לאחר מכן הבחירה של המשתמש מאוחסנת במשתנה ובלולאת while משמשת לביצוע שוב ושוב את האפשרות שנבחרה עד שהמשתמש בוחר לצאת (0)

הפעולות המשתמש יכול לבחור:

1. הצג את קבצי השרת.
2. העלה קובץ לשרת.
3. הורד קובץ מהשרת

עבור כל אפשרות, התוכנית שולחת מסגרות Ethernet וחבילות IP דרך פרוטוקולי UDP או TCP כדי לתקשר עם השרת. החבילות הספציפיות שנשלחות ומתקבלות תלויות באפשרות שנבחרה על ידי המשתמש.

אם המשתמש בוחר באפשרות 1 לצפייה בקבצי השרת, התוכנה שולחת מנה עם ההודעה "ls" לשרת דרך UDP . השרת מגיב עם רשימה של קבצים זמינים, אשר מודפסת למשתמש.

אם המשתמש בוחר באפשרות 2 להעלות קובץ לשרת, התוכנה מבקשת מהמשתמש להעלות את שם הקובץ. לאחר מכן, התוכנית שולחת סדרה של מנות לשרת באמצעות UDP ו-TCP כדי להעביר את הקובץ. ראשית, הוא שולח חבילה עם ההודעה "put " כדי לציין שהוא רוצה להעלות קובץ. השרת מגיב עם חבילת אישור, ולאחר מכן התוכנה שולחת את השם והגודל של הקובץ לשרת. לבסוף, התוכנית שולחת את נתוני הקובץ בחבילת TCP.

אם המשתמש בוחר באפשרות 3 להורדת קובץ מהשרת, התוכנה תבקש מהמשתמש את שם הקובץ להורדה.

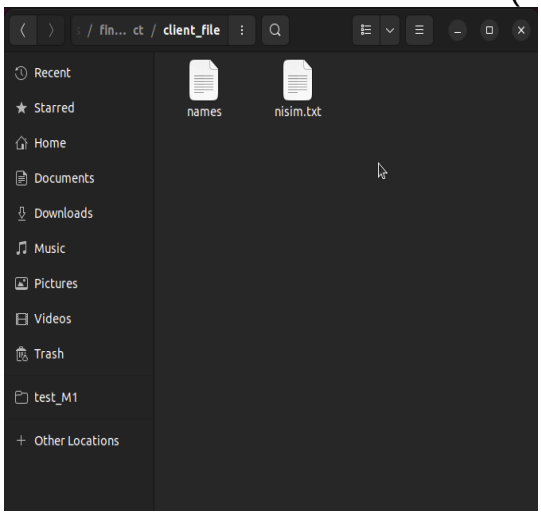
לאחר מכן, התוכנית שולחת סדרה של מנות לשרת באמצעות UDP ו-TCP כדי לבקש ולקבל את הקובץ. ראשית, הוא שולח חבילה עם ההודעה "get " כדי לציין שהוא רוצה להוריד קובץ. השרת מגיב עם חבילת אישור, ולאחר מכן התוכנית שולחת את שם הקובץ להורדה. השרת מגיב עם חבילת ack כדי לציין שהקוב קיים או חבילת "nack " כדי לציין שהקובץ אינו קיים. אם הקובץ קיים, התוכנה מקבלת את נתוני הקובץ בחבילת TCP ושומרת אותם בקובץ במחשב של הלקוח.

וחוזר חלילה עד אשר המשתמש בוחר לצאת ואם המשתמש בחר לצאת מודפסת לו הודעת bye bye

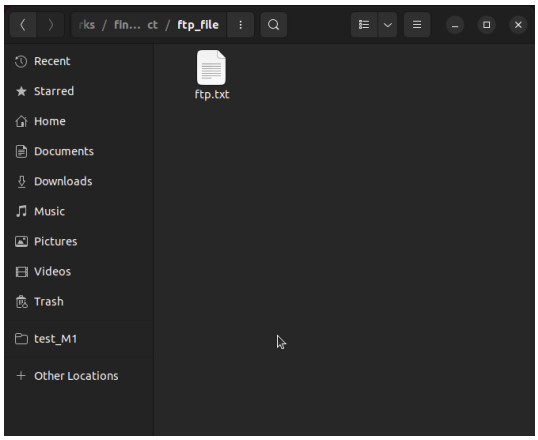
בס"ד

<u>ריצה לדוגמא:</u>

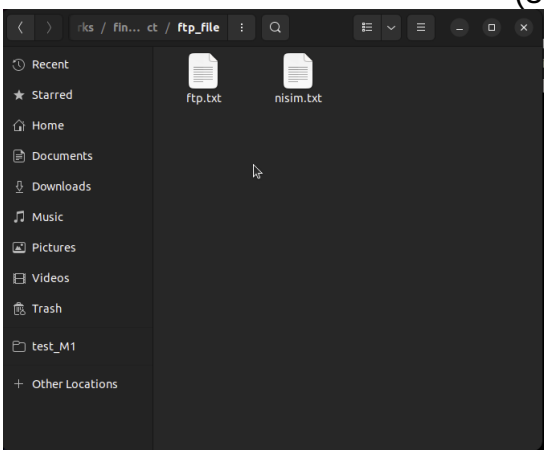
1)



2)

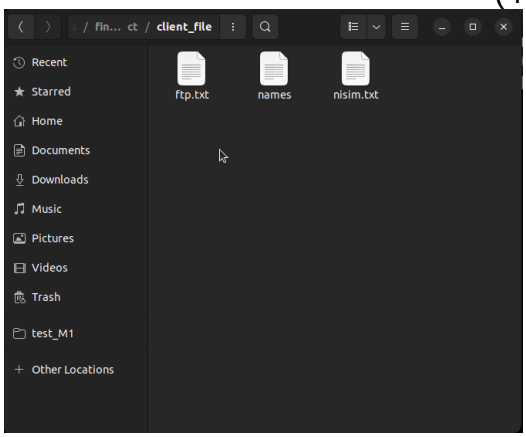

rks / fin... ct / ftp_file
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
test_M1
Other Locations
ftp.txt

(3


rks / fin... ct / ftp_file
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
test_M1
Other Locations
ftp.txt
nisim.txt

בס"ד

(4


/ fin... ct / client_file
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
test_M1
Other Locations
ftp.txt
names
nisim.txt

(5

```
nisim@nisim: ~/c project/Communication networks/
.
Sent 1 packets.
dhcp offer chath!!!
.
Sent 1 packets.
your ip is : 10.0.0.13
The DHCP server ip is : 10.0.0.11
The DNS ip is : 10.0.0.12
.
Sent 1 packets.
The  ip is : 10.0.0.18
Sent DNS response for www.my_ftp.com: 10.0.0.18
(0) to exit
(1) to see the server's files.
(2) to upload a file to the server.
(3) to download a file to the server.
Enter what you want to do: 1
.
Sent 1 packets.
The files inside the ftp server are:
ftp.txt
(0) to exit
(1) to see the server's files.
(2) to upload a file to the server.
(3) to download a file to the server.
Enter what you want to do: 3
.
Sent 1 packets.
Enter the name of the file you want to download: ftp
.
Sent 1 packets.
The server has no file named ftp
Enter the name of the file you want to download: ftp.txt
.
Sent 1 packets.
A file ftp.txt has been added to the ftp server successfully
(0) to exit
(1) to see the server's files.
(2) to upload a file to the server.
(3) to download a file to the server.
Enter what you want to do: 2
.
Sent 1 packets.
Enter the name of the file you want to upload: nisim.txt
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
The file has been uploaded successfully.
(0) to exit
(1) to see the server's files.
(2) to upload a file to the server.
(3) to download a file to the server.
Enter what you want to do: 1
.
Sent 1 packets.
The files inside the ftp server are:
ftp.txt,nisim.txt
(0) to exit
(1) to see the server's files.
(2) to upload a file to the server.
(3) to download a file to the server.
```

בס"ד

(6

```
nisim@nisim: ~/c project/Communication networks/f
.
Sent 1 packets.
['ftp.txt']
.
Sent 1 packets.
The files has been sent to shown successfully.
packet catch
the command that the server will aplay is get
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
The file has been download successfully.
packet catch
the command that the server will aplay is ftp
.
Sent 1 packets.
['ftp.txt']
.
Sent 1 packets.
The files has been sent to shown successfully.
packet catch
the command that the server will aplay is ftp.txt
.
Sent 1 packets.
['ftp.txt']
.
Sent 1 packets.
The files has been sent to shown successfully.
packet catch
the command that the server will aplay is put
.
Sent 1 packets.
The name of the file to upload is: nisim.txt, and its size is: 1294 bytes
A file nisim.txt has been added to the ftp server successfully
packet catch
the command that the server will aplay is nisim.txt
.
Sent 1 packets.
['ftp.txt', 'nisim.txt']
.
Sent 1 packets.
The files has been sent to shown successfully.
packet catch
the command that the server will aplay is 1294
.
Sent 1 packets.
['ftp.txt', 'nisim.txt']
.
Sent 1 packets.
The files has been sent to shown successfully.
packet catch
the command that the server will aplay is ls
.
Sent 1 packets.
['ftp.txt', 'nisim.txt']
.
Sent 1 packets.
The files has been sent to shown successfully.
^Z
[1]+  Stopped                 sudo python3 My_ftp.py
nisim@nisim:~/c project/Communication networks/final projectS
```

בס"ד

7)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port==20027||tcp.port==30663||udp.port==20027||udp.port==30663

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0.0.0.0	255.255.255.255	DHCP	286	DHCP Discover - Transaction ID 0x0
2	0.056770989	192.168.64.1	192.168.64.3	DHCP	342	DHCP Offer - Transaction ID 0x0
3	1.056969649	10.0.0.11	255.255.255.255	DHCP	304	DHCP Offer - Transaction ID 0x0
4	2.102342074	0.0.0.0	255.255.255.255	DHCP	286	DHCP Request - Transaction ID 0x0
5	3.148182350	10.0.0.11	255.255.255.255	DHCP	304	DHCP ACK - Transaction ID 0x0
6	4.194554061	10.0.0.13	10.0.0.12	DNS	74	Standard query 0x0000 A www.my_ftp.com
7	5.248445950	10.0.0.12	10.0.0.13	DNS	104	Standard query response 0x0000 A www.my_ftp.com A 10.0.0.18
8	15.699793952				107	<Ignored>
9	15.700444422				87	<Ignored>
10	16.779775179	10.0.0.13	10.0.0.18	UDP	44	30663 → 20027 Len=2
11	17.576320055				90	<Ignored>
12	17.749306637				90	<Ignored>
13	17.821589792	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
14	18.861694082	10.0.0.18	10.0.0.13	UDP	49	20027 → 30663 Len=7
15	22.794657325				42	<Ignored>
16	22.795720389				42	<Ignored>
17	27.536940679				101	<Ignored>
18	27.537072838				81	<Ignored>
19	29.406958378	10.0.0.13	10.0.0.18	UDP	45	30663 → 20027 Len=3
20	30.446128807	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
21	51.870403669	10.0.0.13	10.0.0.18	UDP	45	30663 → 20027 Len=3
22	52.918643279	10.0.0.18	10.0.0.13	UDP	46	20027 → 30663 Len=4
23	61.662386266	10.0.0.13	10.0.0.18	UDP	49	30663 → 20027 Len=7
24	62.709977231	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
25	63.766545891	10.0.0.18	10.0.0.13	TCP	76	20027 → 30663 [SYN] Seq=0 Win=8192 Len=22
26	64.825657883	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
27	65.870260067	10.0.0.18	10.0.0.13	UDP	49	20027 → 30663 Len=7
28	66.921882790	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
29	67.973458016	10.0.0.18	10.0.0.13	UDP	49	20027 → 30663 Len=7
30	73.644478484				142	<Ignored>
31	73.660385586				110	<Ignored>
32	74.601244477				110	<Ignored>
33	78.362459946	10.0.0.13	10.0.0.18	UDP	45	30663 → 20027 Len=3
34	78.430787822				445	<Ignored>
35	78.430788072				465	<Ignored>
36	79.386451812	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
37	87.542073426	10.0.0.13	10.0.0.18	UDP	51	30663 → 20027 Len=9
38	88.593966271	10.0.0.13	10.0.0.18	UDP	46	30663 → 20027 Len=4
39	89.657990949	10.0.0.13	10.0.0.18	TCP	1348	30663 → 20027 [SYN] Seq=0 Win=8192 Len=1294
40	90.710017564	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
41	91.770818584	10.0.0.18	10.0.0.13	UDP	59	20027 → 30663 Len=17
42	92.826129811	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
43	93.894023568	10.0.0.18	10.0.0.13	UDP	59	20027 → 30663 Len=17
44	104.542795117	10.0.0.13	10.0.0.18	UDP	44	30663 → 20027 Len=2
45	105.565462367	10.0.0.18	10.0.0.13	UDP	45	20027 → 30663 Len=3
46	106.609296361	10.0.0.18	10.0.0.13	UDP	59	20027 → 30663 Len=17

בתמונה 1 ניתן לראות את הקבצים שיש ללקוח בתקיה לפני הרצת הקוד.

בתמונה 2 ניתן לראות את הקבצים שיש לשרת הftp בתקיה לפני הרצת הקוד.

בתמונה 5 ניתן לראות את הבחירות שעשה המשתמש ואת ההכנסות שעשה המשתמש.
בתחילה ניתן לראות שהמשתמש מקבל משרת הdhcp את כתובת הip שלו כתובת הip של שת הdns וכתובת הip של שרת הdhcp , לאחר מכן הוא שולח לשרת הdns שהוא רוצה את כתובת הip של הדומיין www.my_ftp.com ושרת הdns מחזיר לו את כתובת הip של דומיין זה שהיא 10.0.0.18.
לאחר מכן יש למשתמש אפשרות בחירה: 0 כדי לצאת, 1 כדי לראות אילו קבצים יש לשרת הftp להציע, 2 להעלות קבצים לשרת ו3 כדי להוריד קבצים מהשרת.
בחירה ראשונה של המשמש הייתה 1 ואכן הודפס שיש לשרת קובץ יחיד בשם ftp.txt ולאחר מכן שוב למשתמש יש אפשרות לבחור מה לעשות בשלב זה המשתמש בחר להוריד קובץ ולכן הוא הכניס 3 והתוכנה שאלה את המשתמש איזה קובץ הוא רוצה להכניס והוא הכניס שהוא רוצה להוריד את הקובץ ftp, מכיוון שלשרת הftp אין קובץ בשם זה השרת שלח למשתמש שאין אישור להוריד את הקובץ הזה (מכיוון שאין לא אחד כזה) ולכן הודפס למשתמש שאין קובץ כזה לשרת ובקשה נוספת להכניס שם של קובץ.

בס"ד

המשתמש הכניס שהוא רוצה להוריד קובץ בשם ftp.txt ומפני שיש לשרת קובץ בשם זה השרת שלח למשתמש אישור והוא מתחיל בשליחת התוכן של הקובץ, מצד הלקוח הוא מקבל את תוכן הקובץ.

לאחר שכל הקובץ התקבל בהצלחה ונשמר בתקיה הלקוח הודפסה הודעה מתאימה שהקובץ הורד בהצלחה.
לאחר מכן שוב למשתמש יש אפשרות בחירה והוא בחר להעלות קובץ לשרת ולכן הוא הכניס את המספר 2 ונשלחה בקשה לשרת להעלות קובץ, השרת אישר ולכן הודפס למשתמש הודעה שאליה הוא צריך להכניס את שם הקובץ שהוא רוצה להעלות לשרת, הוא הכניס את שם הקובץ nisim.txt והקובץ נשלח לשרת ולאחר שהתקבל בשרת ונשמר אצלו המשתמש קיבל הודעה שהקובץ הועלה בהצלחה.
ושוב למשתמש יש אפשרות בחירה מה לעשות הוא בחר לראות אילו קבצים יש לשרת אז הוא הכניס 1 ואכן לאחר העלאת הקובץ nisim.txt כעת יש לשרת גם את הקובץ ftp.txt וגם את הקובץ nisim.txt ולכן הודפס את שתי שמות הקבצים הללו למשתמש.
לאחר מכן(לא נכנס לתמונה) הלקוח הכניס את המספר 0 והתוכנה קיבלה שהמשתמש בחר לסיים את הריצה והדפיסה לו הודעה מתאימה והתוכנה נסגרה.

לאחר ההסבר של תמונה 5 והרצת התוכנית לפי תמונה 5 נוכל להסביר את תמונות 3 ו4.
אשר בתמונה 3 יש את הקבצים שיש לשרת הftp להציע, לכן בגלל שהלקוח העלה את הקובץ nisim.txt והיה לשרת מבעוד מועד את הקובץ ftp.txt כעת יש לשרת את שני הקבצים הללו.
בתמונה 4 יש את הקבצים של הלקוח ולכן יש לו את הקבצים שהיו לו במעוד מועד ובגלל שהוא הוריד מהשרת את הקובץ ftp.txt כעת יש לו גם את הקובץ הזה וגם את שני הקבצים שהיו לו מבעוד מועד.

בתמונה 6 נוכל לראות את הפלטים של השרת ftp שבנינו, תחילה (לא נכנס לתמונה) יש הדפסה של השרת שהוא קיבל את הבקשה של המשתמש להראות אילו קבצים יש לו, הוא שולח אישור לזה ולכן הוא מדפיס את רשימת הקבצים שיש לו גם כן ושולח אותה ומדפיס לאחר מכן שהוא שלח את הרשימה בהצלחה.
לאחר מכן הוא מצפה לקבל עוד בקשה מה לעשות מהמשתמש, הוא מדפיס שהוא קיבל בקשה להורדה מהמשתמש(get) לאחר מכן הוא מצפה לקבל את שם הקובץ אותו המשתמש רוצה להוריד וכפי שהזכרתי למעלה הוא הכניס שם קובץ אשר לשרת אין קובץ בשם כזה ולכן הוא שולח הודעה מתאימה אשר אומרת למשתמש שאין לו קובץ בשם הזה (nack).
לאחר מכן הוא שוב מצפה לקבל שם של קובץ מהמשתמש אותו המשתמש רוצה להוריד מהשרת ומפני שהמשתמש הכניס שם של קובץ מתאים הוא שולח אישור למתשמש(ack) ומתחיל לשולח את הקובץ.
לאחר שהוא סיים לשלוח את הקובץ הוא מדפיס הודעה מתאימה אשר אומרת שהוא שלח את הקובץ בהצלחה.
לאחר מכן הוא מצפה לקבל מהמשתמש שוב מה הוא רוצה לעשות והוא קיבל שהמשתמש רוצה להעלות קובץ(put) ולכן הוא מדפיס הודעה מתאימה, השרת מצפה לקבל את שם הקובץ ואת גודל הקובץ ולאחר שקיבל את שני אלה הוא שולח אישור למשתמש שהוא מוכן לקבל את הקבץ ומדפיס הודעה מתאימה שאומרת שהמשתמש רוצה להעלות קובץ בשם nisim.txt ושהגודל של הקובץ הוא 1294 בטים, ולאחר שהוא קיבל אותו הוא מדפיס שהקובץ הועלה בהצלחה.
לאחר מכן הוא מצפה לקבל עוד פאקטה שמכילה פקודה מהמשתמש אשר תעדכן את השרת מה המשתמש רוצה לעשות ולאחר שהוא קיבל הוא מדפיס הודעה מתאימה שהמשתמש רוצה לראות אילו קבצים יש לו להציע(ls) ולכן הוא מדפיס הודעה מתאימה.
הוא שולח שהוא מאשר את הבקשה מדפיס את רשימת הקבצים שיש לו ושולח אותה למשתמש ולאחר מכן הוא מדפיס שהקבצים נשלחו להצגה בהצלחה.

בס"ד

בתמונה 7 נוכל לראות את ההסנפה של תוכנת הווירשארק.
פריים 1-5 זה החלק שהלקוח מקבל קונפיגורציה משרת הdhcp לגבי כתובת הip של הלקוח עצמו וגם את כתובת הip של השרת dhcp וגם את כתובת הip של שרת הdns.

אפשר לראות הסבר מפורט יותר בחלק של Dynamic Host Configuration Protocol ובחלק של הקוד שלנו בתת חלק dhcp.

פריים 6-7 זה החלק שהלקוח מבקש ומקבל את כתובת הip של הדומיין המבוקש.
אפשר לראות הסבר מפורט יותר בחלק של Domain Name System -DNS ובחלק של הקוד שלנו בתת חלק dns.
בפריים 10 זה פאקטה שנשלחה מהלקוח לשרת (נוכל לזהות את זה לפי כתובת הip מקור וip יעד בהתאמה וכך לכל שאר הפריימים) וזאת פאקטת הבקשה של המשתמש לפקודת (ls).
בפריים 13 זאת פאקטת האישור על הבקשה של המשתמש לפקודת (ls) אשר נשלחת מהשרת ללקוח.
פריים 14 היא הפאקטה שמכילה את רשימת הקבצים שיש לשרת.
פריים 19 זה פאקטה שנשלחה מהלקוח לשרת וזאת פאקטת הבקשה של המשתמש לפקודת (get).
פריים 20 זו היא פאקטת האישור לפקודת הget
פריים 21 זה פאקטה המכילה את שם הקובץ השגוי להורדה.
פריים 22 זאת פאקטת הnack.
פריים 23 זה פאקטה המכילה את שם הקובץ להורדה(בלי שגיאה).
פריי 24 זה פאקטת הack.
פריים 25-29 זה השליחה של התוכן הקובץ מהשרת ללקוח.
פריים 33 זאת בקשה נוספת של הלקוח לשרת להעלות קבצים לשרת (put).
פריים 36 זאת פאקטת האישור לבקשה.
פריים 37 זאת פאקטה המכילה את שם הקובץ שהמשתמש רוצה להעלות.
פריים 38 זאת פאקטה המכילה את גודל הקובץ שהמשתמש רוצה להעלות.
פריים 39 זה השליחה של התוכן הקובץ מהלקוח לשרת.
פריים 40-43 זה האישורים של תוכן הקובץ שהועלה לשרת נשלחים מהשרת ללקוח.
בפריים 44 זה פאקטה שנשלחה מהלקוח לשרת וזאת פאקטת הבקשה של המשתמש לפקודת (ls).
בפריים 45 זאת פאקטת האישור על הבקשה של המשתמש לפקודת (ls) אשרת נשלחת מהשרת ללקוח.
פריים 46 היא הפאקטה שמכילה את רשימת הקבצים שיש לשרת.

בס"ד

<u>שאלות של הpdf:</u>

1.מנה לפחות 4 הבדלים עקריים בין פרוטוקול tcp ל-quic
2.מנה לפחות 2 הבדלים עקריים בין quic ל-vegas
3.הסבר מהא פרוטוקול bgp, במהוא הוא שונה מ-ospf והאם הוא עובד לפי מסלולים קצרים
.4
5. הסבירו את ההבדלים בין פרוטוקולARP ל-dns

<u>תשובות לשאלות של הpdf:</u>

.1
tcp ו-quic הם שניהם פרוטוקולי שכבת תחבורה, ישנם מספר הבדלים עיקריים ביניהם, נציג ארבעה:

1) אמינות:TCP הוא פרוטוקול אמין, כלומר הוא מבטיח שכל הנתונים מועברים ליעד בסדר הנכון וללא שגיאות. לעומת זאת, QUIC משתמש במנות (UDP) שלעצמן הן לא מהימנות, אך הוא כולל מנגנוני מהימנות משלו כדי להבטיח אספקת נתונים.
משמע tcp הוא פרוטוקול אמין בעוד שquic משתמש בudp אך יש לו מנגנוני מהמנות.

2) tcp דורש לחיצת יד תלת כיוונית כדי ליצור חיבור בין שתי נקודות קצה לפני שניתן יהיה להעביר נתונים כלשהם.
quic לעומת זאת משתמש כאמור בפרוטוקול udp אזי החיבור הוא יותר מהיר ומאפשרת ללקוח לשלוח נתונים מיד עם החיבור.
משמע tcp דורש פתיחת קשר בעוד שquic לא.

3) TCP משתמש באלגוריתם בקרת גודש שמאט את הקצב שבו נשלחים נתונים כאשר מגלים עומס ברשת.
quic משתמש במנגנון בקרת גודש דומה, אך הוא גמיש יותר ומותאם לתנאי הרשת.
משמע tcp לא מתייחס לתנאי רשת בעוד שquic כן

4) ל-TCP אין תכונות אבטחה מובנות והוא מסתמך על פרוטוקולים ברמה גבוהה יותר.
QUIC לעומת זאת, כולל מנגנוני הצפנה ואימות מובנים, מה שהופך אותו כברירת מחדל לפרוטוקול מאובטח יותר.
לtcp אין תוכנות אבטחה מובנות ולquic יש.

.2
quic וvegas הם שני פרוטוקולים שונים המשמשים ברשתות מחשבים.
שני הבדלים עיקריים ביניהם הם:

1) quic הוא פרוטוקול שכבת תעבורה שרץ על פרוטוקולUDP בעוד הvegas הוא אלגוריתם בקרת גודש המשמש בשכבת התעבורה של TCP
משמע quic פרוטוקול בשכוות התעבורה בעוד שvegas הוא אלגוריתם בקרת גודש בפרוטוקול tcp.

בס"ד

2) quic משתמש בגישת בקרת גודש המבוססת על אובדן מנות, בעוד vegas משתמשת בגישת בקרת גודש המבוססת על מדידת זמן השהייה ברשת.
quic מנטר אובדן מנות ומתאים את גודל חלון הגודש כדי לשלוט בכמות הנתונים הנשלחת,
בעוד vegas משתמשת בגישה מבוססת-השהייה כדי לשלוט על הגודש על ידי מדידת זמן הלוך ושוב של מנות.
משמע quic מודד אובן מנות ברשת וכך שולט על הגודש בעוד שvegas מסתכל על הזמן נסיעה הלוך ושוב של מנות כדי לזהות גודש.

.4

applicition	port src	port des	ip port	ip des	mac src	mac des
my_ftp	20027	30663	10.0.0.18	10.0.0.13	7e:b1:37:1c:4b:d4	7e:b1:37:1c:4b:d4
dhcp	67	68	10.0.0.11		7e:b1:37:1c:4b:d4	7e:b1:37:1c:4b:d4
dns	53	5353	10.0.0.12		7e:b1:37:1c:4b:d4	7e:b1:37:1c:4b:d4

במקרה של nat יתווספו עוד פאקטות נוספות ובמקרה של quic לא ישתנה שום דבר מהטבלה.

.3
bgp הוא פרוטוקול הבוחר את הנתיב הטוב ביותר ליעד בין מערכות אוטונומיות שונות על סמך המידע שהוא מקבל מהתקני הרשת האחרים.

ospf הוא פרוטוקול המוצא את הנתיב הקצר ביותר בין נתבים באותה רשת.

אחד ההבדלים העיקריים בין bgp ל ospf הוא זה שbgp פועל על מערכות אוטונומיות שונות וospf פועל באותה רשת.
עוד הבדל הוא שbgp מוצא את הניתוב הטוב ביותר ולאו דווקא הקצר ביותר לעומת זאת, ospf מוצא את הניתוב הקצר ביותר.
כאמור לעילBGP לא בהכרח עובד לפי מסלולים קצרים. במקום זאת, הוא בוחר את הנתיב הטוב ביותר על סמך גורמים שונים.

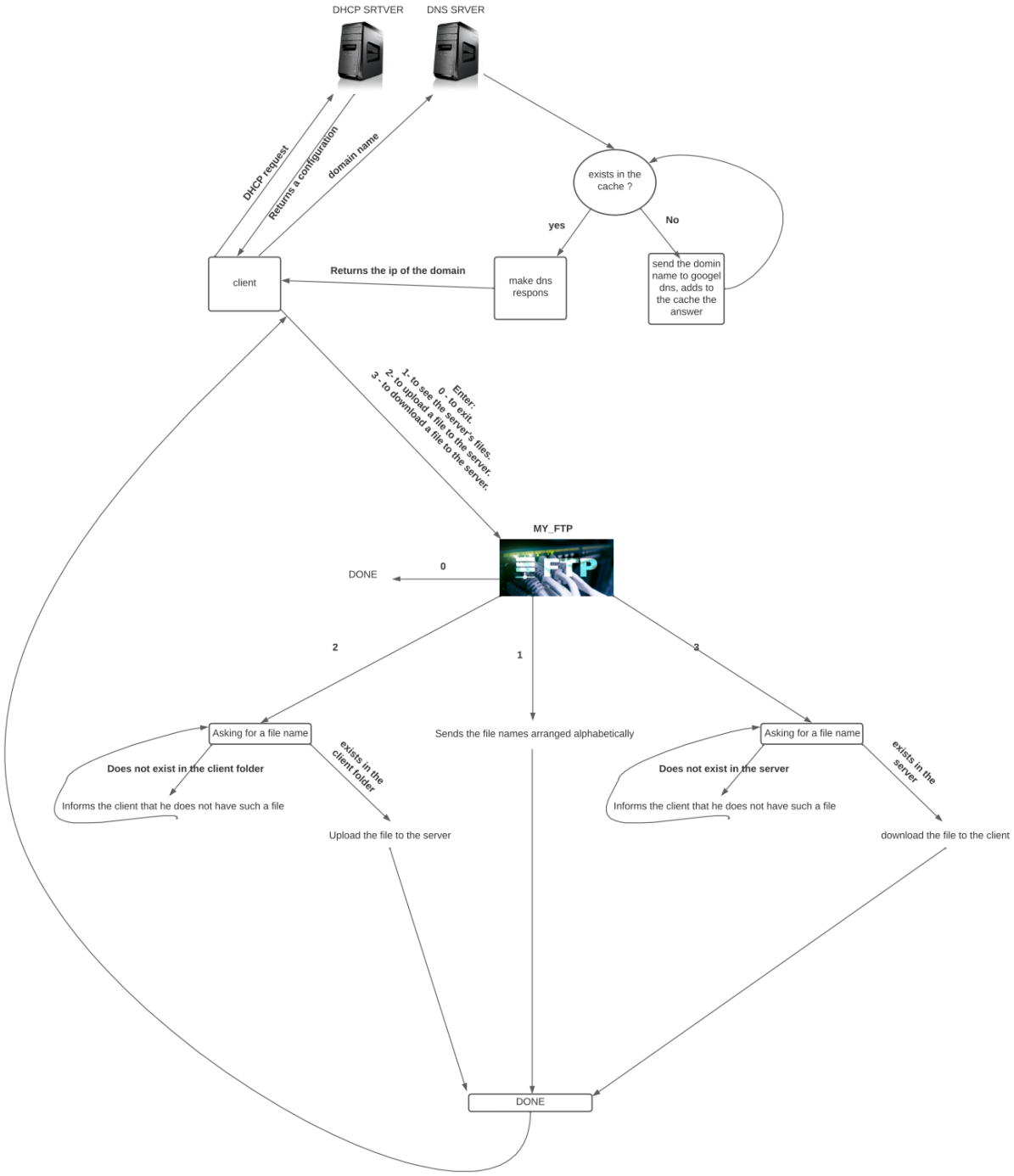
.5
ARP משמש למיפוי כתובת IP לכתובת פיזית כתובת MAC.

כאשר מכשיר ברשת רוצה לתקשר עם מכשיר אחר, הוא צריך לדעת את הכתובת הפיזית של המכשיר האחר.
ARP משמש לפתרון מיפוי זה על ידי שידור הודעת בקשה לכל המכשירים ברשת, בקשת מהמכשיר עם כתובת הרשת שצוינה להשיב עם הכתובת הפיזית שלו. לאחר קבלת המיפוי, המכשירים יכולים לתקשר אחד עם השני.

DNS לעומת זאת, משמש למיפוי שם דומיין לכתובת IP.
כאשר משתמש מקליד שם דומיין בדפדפן אינטרנט, הדפדפן צריך לדעת את כתובת ה-IP של שרת האינטרנט המארח את האתר.
DNS משמש לפתרון מיפוי זה על ידי שאילתה בשרת DNS עבור כתובת ה-IP המשויכת לשם התחום.
לאחר מכן שרת ה-DNS מחזיר את כתובת ה-IP ללקוח, ומאפשר ללקוח ליצור חיבור עם שרת האינטרנט.

בס"ד

<u>דיאגרמת מצבים:</u>



בס"ד

הוראות הפעלה:

0. הריצו על מחשבי לינוקס, החליפו את כתובת המק שרשומה בקובץ my_ftp.py ובקובץ client.py לכתובת המק של המחשב שלכם.
1. צרו תיקיה.
2. הורידו את הקבצים הבאים לתוך התיקיה שיצרתם: client.py dhcp_server.py dns.py my_ftp.py .
3. הורידו את התיקיות הבאות לתוך התיקיה שיצרתם: client_file ftp_file.
4. הכניסו כרצונכם קבצי טקסט עד גודל 1.3KB לתוך התקיות משלב 3.
5. פתחו 4 טרמנלים דרך התקיה שיצרתם בשלב 1.
6. הריצו תחילה את dhcp_server.py ולאחר מכן את dns.py ולאחר מכן את my_ftp.py ולבסוף את client.py בכל טרמינל בנפרד על ידי הפקודה הבאה sudo python3 file_name.py
7. הכניסו את שם הדומיין הרצוי(בכדי להשתמש באפליקציה הכניסו (www.my_ftp.com).

הסבר לסרטון הדרכה:

בתחילת הסרטון היראתי שיש לי תיקיה ואני הכנסתי לתוכה את ארבעת הקבצים כולל שתי התיקיות (כמו בהוראות).
לאחר מכן היראתי שבתיקיה cilent_file יש 2 קבצים names.txt ו-nisim.txt ובתקיה של השרת יש קובץ בשם ftp.txt.
לאחר מכן פתחתי 4 טרמינלים הרצתי את התוכנות לפי ההוראות, תחילה את dhcp_server.py ולאחר מכן את dns.py ולאחר מכן את my_ftp.py ולבסוף את client.py.
בכנסתי את שם הדומיין של האפליקציה (www.my_ftp.com).
תחילה רציתי לראות אילו קבצים יש לשרת ולכן הכנסתי 1, לאחר שקיבלתי את רשימת הקבצים שיש לשרת להציע רציתי להוריד מהשרת לכן הכנסתי 3.
כשהאפליקציה שאלה אותי מה שם הקובץ אותו רציתי להוריד הכנסתי את שם הקובץ ולאחר שהקובץ הורד קיבלתי הודעה מהטרמינל שהקובץ הורד בהצלחה.
לאחר מכן רציתי להעלות קובץ לשרת לכן הכנסתי 2 , שוב הכנסתי את שם הקובץ שרציתי להעלות ולאחר שהקובץ הועלה קיבלתי הודעה מהטרמינל שהקובץ הורד בהצלחה.
לבסוף רציתי לראות אילו קבצים יש לשרת כרגע לכן הכנסתי שוב 1 וראיתי שיש לו את הקובץ שהיה לו וגם את הקובץ שאני העלתי לו
לסיום רציתי לצאת לכן הכנסתי 0 והתוכנה הסתיימה והדפיסה לי bye bye.
בסוף הסרטון נכנסתי לתיקיה בכדי להראות בעוד דרך שאכן הקובץ שהעלתי לשרת נמצאת אצלו וגם הקובץ שהורדתי מהשרת נמצאת אצל הלקוח.

בס"ד

<u>ביבליוגרפיה:</u>

1. stackoverflow
2. techieselight
3. קורס רשתות תקשורת אוני' אריאל
    1. הקלטות של אנה
    2. הקלטות של אלמוג שור
    3. הקלטות של עמית דביר