שיעור 6 – שכבת התעבורה - UDP

**תרגיל: הסנפות מתקדמות**

בתרגיל זה נעיין בהסנפות תוך התמקדות בשתי השכבות אותן אנחנו מכירים - שכבת התעבורה ובשכבת האפליקציה.

**1. פרוטוקולים**

1. התחילו הסנפה חדשה ב-Wireshark, גלשו לאתר **bizportal.co.il,** ולאחר שהאתר נטען עצרו את ההסנפה.
2. השתמשו בפילטר udp או tcp כדי לסנן חבילות אשר כוללות בתוכן את אחד מהפרוטוקולים בשכבת התעבורה.
3. רשמו ארבעה פרוטוקלים שאתם מוצאים בטבלה, ומלאו את המידע שאתם מוצאים על כל פרוטוקול (את כל המידע ניתן להסיק מההסנפה בלבד ללא צורך בגוגל).

(שימו לב – חבילות שבשדה פרוטוקול כתוב רק UDP או TCP – התעלמו מהן בשלב זה. אלו חבילות ללא שכבת אפליקציה או שהפרוטוקול אינו מזוהה).

|  |  |  |  |
| --- | --- | --- | --- |
| פרוטוקול | טקסטואלי / בינארי? | בשכבת התעבורה יש.. | Port (בצד השרת) |
|  |  | TCP |  |
|  |  | TCP |  |
|  |  | UDP |  |
|  |  | UDP |  |

סביר שתמצאו בהסנפה לפחות את פרוטוקול DNS ו-HTTP עליהם למדנו כבר. HTTP כזכור משתמש ב-TCP והוא טקסטואלי (פורט 80 בצד השרת), בעוד ש-DNS משתמש ב-UDP, הוא בינארי (פרט לדומיין עצמו) ומשתמש בפורט 53 בצד השרת.

1. הסבירו כיצד יתכן שבחבילה אחת יש גם את פרוטוקול UDP וגם פרוטוקול נוסף?

|  |
| --- |
| זה היופי של מודל השכבות!  לשני הפרוטוקולים תפקידים שונים בתכלית. אם ניקח את הדוגמה של HTTP ו-TCP, אז תפקידו של פרוטוקול HTTP הוא להעביר מידע המכיל דפי אינטרנט בין שרת ה-WEB לדפדפן של הלקוח, בעוד שתפקיד פרוטוקול TCP הוא לאפשר את הקיום של השיחה, וידוא תקינות המידע ואפשור של שיחות נוספות במקביל (כאשר TCP לא מודע למידע עצמו שעובר). כמו שלמדנו על עיקרון הכימוס, שכבת האפליקציה מגדירה את המידע שעובר בה ומקודדת אותו בסופו של דבר לביטים (במקרה של HTTP, בעזרת קידוד ASCII של הטקסט), ושכבת התעבורה לוקחת את הביטים האלה (בלי להבין מה יש בהם בכלל) ודואגת לניהול השיחה שמאפשרת את ההעברה שלהם בין שני הצדדים. |
| תשובה |

1. באיזו שכבה כל הפרוטוקולים שרשמתם בעמודה הימנית בטבלה?

|  |
| --- |
| כל הפרוטוקולים הללו הם בשכבת האפליקציה והם משתמשים בפרוטוקול UDP או TCP בתור פרוטוקול התעבורה שלהם. |
|  |
| תשובה |

**2. DNS**

1. הציגו רק את חבילות ה-DNS שיצאו מהמחשב שלנו. כתבו שני פילטרים שונים שיבצעו את הסינון הזה:

|  |
| --- |
| כמה פילטרים אפשריים לדוגמה (אפשר לחשוב על פילטרים נוספים כמובן):  dns and ip.src == <MY\_IP>  udp.dstport == 53 and ip.src == <MY\_IP>  dns and udp.dstport == 53 |
| תשובה |

1. כתבו 5 דומיינים שונים אותם ביקש הדפדפן שלנו (ב-DNS):

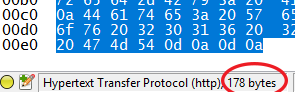
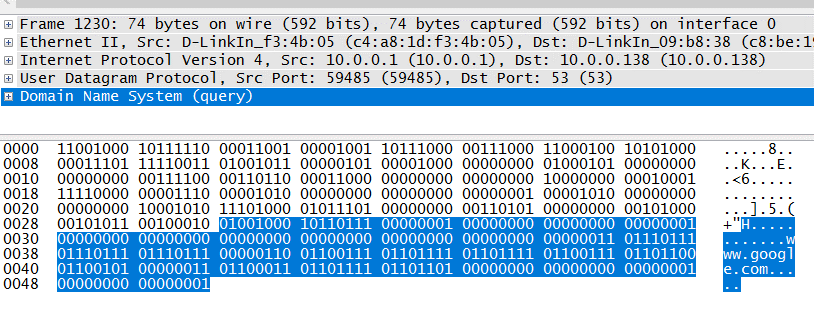
|  |
| --- |
| יש מלא דומיינים שמופיעים מכיוון שהאתר משתמש בכל מיני תוספים מאתרים שונים, מציג פרסומות עוד ועוד.  bizportal.co.il adservices.google.com taboola.com hotjar.com bit.ly ועוד ועוד... |
| תשובה |

1. בחרו את אחת מחבילות ה-DNS. כתבו את האורך שלה המופיע תחת העמודה Length:

|  |
| --- |
|  |
| תשובה |

1. באיזה פרוטוקול בחר DNS להשתמש בשכבת התעבורה?

|  |
| --- |
| DNS תמיד משתמש ב-UDP. מכיוון שההודעות הן קצרות והמהירות חשובה. במקרה שחבילה לא תגיע, מנגנון ה-DNS יבקש אותה שוב פשוט. |
| תשובה |

1. כאשר אנחנו מסמנים את אחת השכבות במסך השכבות, בשורה התחתונה מוצג האורך של השכבה הזו בלבד (בתוך החבילה).
2. בחבילה שבחרתם, כתבו מהו האורך (בבתים) של כל אחת מהשכבות:

|  |
| --- |
| אפליקציה =  תעבורה =  רשת = קו = |
| תשובה |

1. מהי השכבה הארוכה ביותר? האם זה מצב הגיוני לדעתכם? מדוע?

|  |
| --- |
| ברוב המקרים (אלא אם כן ההודעה שנשלחת היא קטנה במיוחד), השכבה שבה יעבור רוב המידע היא שכבת האפליקציה. אנחנו שמחים שזה המצב, אחרת רוב המידע שהיינו מעבירים בשיחה היה מידע על השיחה, וכל שיחה הייתה לוקחת הרבה זמן.  בסופו של דבר, לכל חבילה יש גודל מקסימלי שנקבע בגלל האילוצים של התווך שעליו היא עוברת. הפרוטוקולים של השכבות הנמוכות, שאחראיים על וידוא שהמידע עובר כמו שצריך, צריכים לתפוס כמה שפחות מהגודל של החבילה כדי שנצטרך בסופו של דבר כמה שפחות חבילות כדי להעביר את המידע שלנו.  המידע שעובר בשכבת האפליקציה הוא הסיבה לשיחה כולה, ולכן טוב שהמידע הזה הוא רוב החבילה... |
| תשובה |

1. מהו פורט המקור (Src Port) שבחר המחשב שלי לשיחת DNS זו? כיצד הוא בחר את ה-Port הזה?

|  |
| --- |
| סביר מאוד שהמחשב שלכם בחר פורט גבוה (מעל 1024) בשביל השיחה. כמו שלמדנו, הפורטים הנמוכים שמורים לפרוטוקולים מוגדרים מראש (כמו שפורט 53 מוגדר ל-DNS) ושאר הפורטים הם חופשיים לשיחות שהמחשב רוצה ליזום מול שרתים שונים.  המחשב שלכם בחר פורט אקראי כלשהו שהיה פנוי באותו הזמן, הקצה אותו לשיחת ה-DNS, וברגע שהשיחה נגמרה הוא יכול להקצות את הפורט לשיחות אחרות שיבואו בעתיד. |
| תשובה |

1. לחצו פעמיים על השכבה של UDP במסך השכבות כדי לפתוח את הפירוט שלה. הסבירו על כל אחד מארבעת השדות שכולל ההדר – מה שמו, מה האורך שלו ומה התפקיד שלו.

|  |
| --- |
| בהתחלת ה-header יש את פורט המקור ופורט היעד, שהצד השני משתמש בהם כדי לשייך את ההודעה לשיחה המתאימה. כל אחד מהשדות הללו הוא באורך 2 בתים כמו שלמדנו בשיעור.  לאחר מכן, יש שדה שמתאר את האורך של ה-header והמידע שבא אחריו (בקשת ה-DNS) ביחד. המספר שמתואר בו הוא האורך בבתים, והוא משמש את הצד המקבל בשביל לדעת מתי המידע נגמר ומתחילה הודעה אחרת. אורך השדה הזה הוא 2 בתים.  לבסוף, ישנו שדה ה-checksum, שמאפשר וידוא שהמידע שהתקבל הוא אכן תקין ולא היו שגיאות בדרך (מה שיגרור התעלמות מכל החבילה במקרה של DNS). גם האורך כאן הוא 2 בתים. |
| תשובה |

1. מה המספר המקסימלי שיכול להיכנס בשדה Length? הסבירו כיצד חישבתם זאת.

|  |
| --- |
| שדה האורך מוגבל ע"י 2 בתים, שזה אומר 16 ביטים. כמו שחישבנו בעבר, במקרה כזה כמות המספרים שמוכלים ב-16 ביטים הם מאפס עד 2 בחזקת 16 (פחות אחד), ולכן המספר המקסימלי יהיה 65535!  זה אומר שעבור UDP, לא ניתן להעביר בחבילה אחת יותר מ-65527 בתים של מידע משכבת האפליקציה... |
| תשובה |

1. סמנו במסך השכבות את השדה Source Port. שימו לב לקטע ההקסא שמסומן במסך התחתון. העתיקו את הקטע לאתר שממיר ממספר הקסא למספר דצימלי (hex to dec). מהו המספר שקיבלתם? האם הוא המספר שציפיתם?

|  |
| --- |
| Hex = Dec = |
| תשובה |

**3. HTTP**

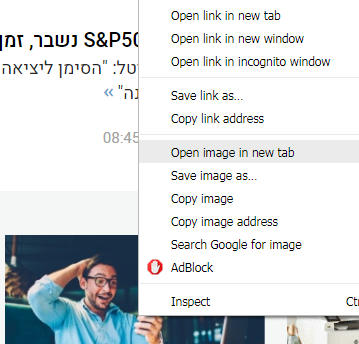
1. כעת נבצע סינון לחבילות HTTP בלבד.
2. באיזה פרוטוקול משתמש HTTP בשכבת התעבורה?

|  |
| --- |
| HTTP משתמש בפרוטוקול TCP בתור הפרוטוקול של שכבת התעבורה שלו (נלמד על TCP בהרחבה בשבוע הקרוב). |
| תשובה |

1. מדוע לדעתכם הוחלט ש-HTTP ישתמש דווקא בפרוטוקול זה?

|  |
| --- |
| עבור HTTP חשובה לנו מאוד האמינות של המידע שעובר, ולכן נהיה מוכנים לסבול עיכוב קל בהעברת המידע בשביל לוודא שהוא מגיע כמו שצריך.  אפשר לחשוב שאנחנו גם נרצה אמינות של המידע של DNS ולכן למה שהוא לא יעבור ב-TCP גם כן? ובכן, ב-DNS אנחנו מעבירים כמות קטנה מאוד של מידע, ולכן לא נרצה שרוב המידע שמועבר יהיה עבור הקמת השיחה, כשאנחנו יכולים לסגור את השיחה בחבילה אחת... אז אנחנו פשוט משתמשים ב-checksum כדי להבין אם המידע עבר תקין, ואם הוא לא תקין אז אנחנו פשוט נשלח בקשת DNS חדשה ונקווה שבפעם הבאה הוא יעבור תקין (וזה עדיין יהיה יותר יעיל מבחינת נפח התעבורה). |
| תשובה |

1. באתר הראשי של bizportal ישנן תמונות הנלוות לכתבות. בעזרת כפתור ימני על התמונות ו"פתיחת תמונה בכרטיסיה חדשה", חקרו מהו המאפיין שחוזר בשם הקובץ שלהן.

  
  
כתבו פילטר חכם המציג את הבקשות של כל התמונות באתר בלבד.  
בדקו שהוא עובד על ידי טעינה מחדש של האתר (מומלץ עם Ctrl+F5).

|  |
| --- |
| אפשר לראות שכל התמונות נמצאות בתיקייה rsPhoto, אז פילטר שיעבוד הוא למשל  http and frame contains "rsPhoto" |
| תשובה |

1. גלו כיצד ניתן לסנן HTTP ע"פ Response Code. ניתן לעשות זאת בעזרת גוגל או ע"י כתיבת http ולאחריו נקודה בשדה הפילטר. כתבו פילטר אשר מציג רק תשובות של 200 OK:

|  |
| --- |
| דרך נוחה לזהות איך ניתן לפלטר על פי שדה כלשהו שנמצא בפקטה היא ללחוץ על השדה הרלוונטי בלחיצה ימנית ב-Wireshark, ואז ללחוץ על “Apply As Filter”. זה יפלטר על בסיס השדה הזה ונוכל לראות בחלון הפילטור מה בדיוק הפילטר המתאים. במקרה שלנו:  http.response.code == 200 |
| תשובה |

1. כתבו פילטר אשר מציג רק את התגובות שאינן 200:

|  |
| --- |
| http and http.response.code != 200  ניתן למצוא פילטרים נוספים שיעבדו כמובן. |
| תשובה |

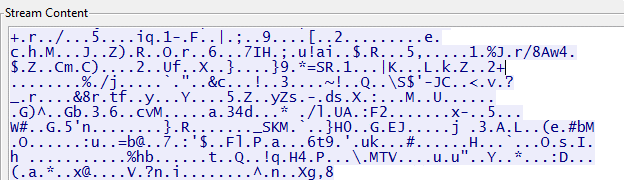
1. חזרו לפילטר שמציג את התשובות של 200. כעת אנו למעשה רואים את כל הקבצים אשר קיבלנו מהשרת. כמה קבצים מוצגים?

|  |
| --- |
|  |
| תשובה |

1. אילו סוגי קבצים אתם רואים בבקשות HTTP? כתבו לפחות 5.

|  |
| --- |
| בבקשות התגובה של HTTP קבצים שונים מסוג טקסט (html, javascript, css וכו'), וגם תמונות מסוגים שונים (jpeg, png). |
| תשובה |

\* הידעת? עד היום לא דיברנו על כך, אבל כאשר HTTP מעביר תמונות או קבצים אחרים הוא מעביר אותן בצורה בינארית. כך ש-HTTP, למרות שבמהותו הוא פרוטוקול טקסטואלי, מסוגל להעביר גם מידע בינארי לפעמים.

1. **בונוס:** ידוע לנו ש-HTTP הוא פרוטוקול טקסטואלי, ובאתר bizportal לא מתבצעת הצפנה (אין סימן של מנעול בגלישה, והפורט הוא 80 ולא 443 של HTTPS). אם כך מדוע התשובות של השרת מגיעות אלינו בצורה בינארית, גם כאשר אנחנו מבקשים קובץ טקסטואלי כמו HTML?   
   רמז – התשובה נמצאת בהדרים...

|  |
| --- |
| אם נסתכל על ה-header של התגובה מצד השרת, נראה שדה שנקרא content-encoding. הוא מתאר בעצם את צורת הקידוד שהשרת שולח לנו את המידע בה, והדפדפן שלנו צריך להיות מסוגל לפענח את הקידוד ולהציג לנו את המידע כמו שהוא באמת...  במקרה הזה, השרת שולח לנו את המידע בקידוד gzip, שאמור לכווץ את הגודל של הטקסט ולחסוך בכמות רוחב הפס שאנחנו מתשמשים בו עבור השיחה. הבעיה היא ש-Wireshark לא מזהה את התעבורה בתור gzip (אין לו אופציה של קידוד כזה במסך ה-Follow TCP Stream) ולכן המידע נראה לנו משובש (או "מוצפן"). בפועל, אם נבצע decode למידע, נראה שהוא טקסטואלי לחלוטין ולא מוצפן בכלל... |
| תשובה |

1. **בונוס:** בצעו כמה טעינות רצופות של bizportal עם Reload. לאחר מכן הסניפו טעינה נוספת של bizportal. כעת הדפדפן שלכם אמור לזכור חלקים מהאתר (למשל תמונות) ולא להוריד אותם מחדש, כדי לחסוך בתעבורה. מצאו תגובות HTTP שתומכות בטענה זו, והסבר מה הן אומרות.

|  |
| --- |
| תגובות 304 Not modified הן תגובות אשר מעידות שהשרת אמר לנו: "הקובץ לא השתנה מאז הפעם האחרונה שביקשתם אותו. כך שאתם לא צריכים להוריד אותו מחדש". הוא מלמד אותנו שהדפדפן שלנו גם הוא מנהל זכרון cache (מטמון) כדי להאיץ את הגלישה שלנו באינטרנט (קחו למשל את ספרות התאריך, הן לא השתנו מאז שהאתר ynet הוקם כך שאין צורך להוריד אותן כל פעם מחדש). |
| תשובה |