שנה"ל **תשפ"ה**__, סמסטר _**א**_, מועד א___
שאלון בחינה בקורס: רשתות מחשבים מתקדם
מספר קורס: 156336.2.5785

- שם המרצה: ברק גונן, אבי טריסטמן, גיא סעדון
- תאריך הבחינה: 16.2.2025
- משך הבחינה (בדקות): 180
- חומר עזר מותר לשימוש: הכל למעט בינה מלאכותית ועזרה מחברים
- מחשבון: כן
- המבחן כולל סה"כ 2 שאלות, יש לענות על 2 שאלות.

תלמידים יקרים ויקרות,

1. **המבחן כולל את כל המידע הנדרש. קראו אותו היטב והעזרו בטיפים ובקבצי השלד**.
2. נוהל הבחינות של המרכז האקדמי לב מחייב אותך, באחריותך לקוראו ולהכירו - בחינה עלולה להיפסל על כל חריגה מהנוהל.
3. אם אינך מבין את כוונת המרצה בשאלה כלשהי, עליך לכתוב בראש התשובה כיצד הינך מבין את השאלה ולפתור בהתאם. המרצה ישקול האם יש מקום להבנה זו ואז ינקד בהתאם.

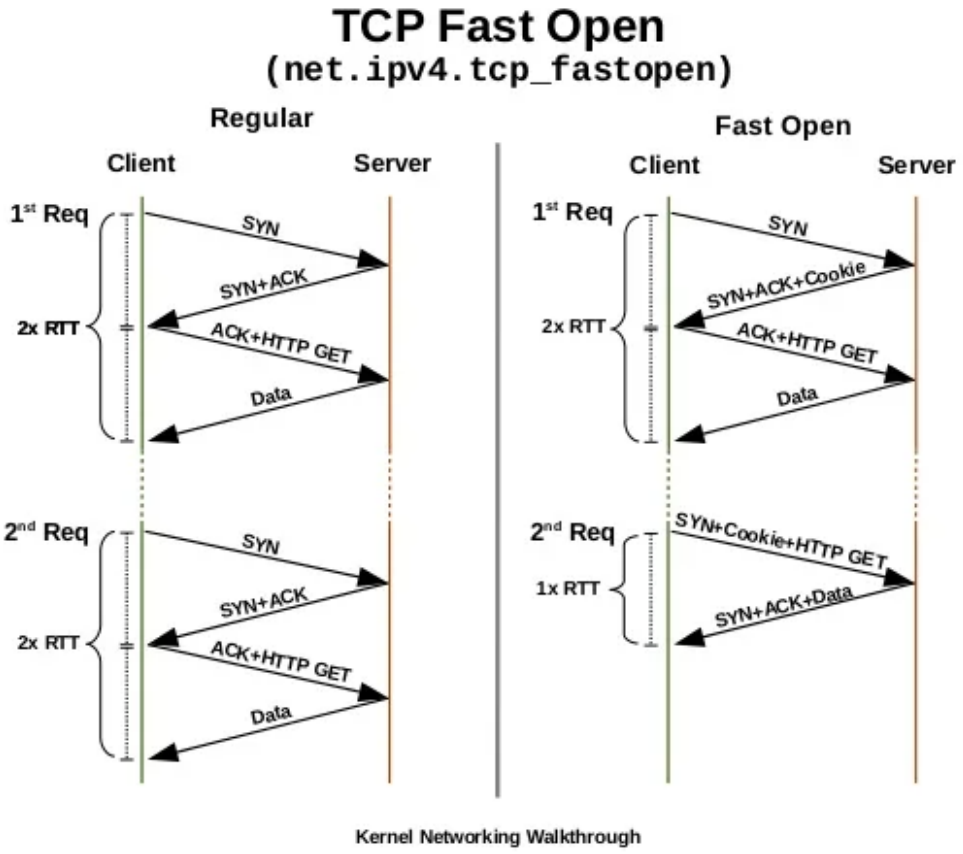
בהצלחה רבה !

שנה"ל **תשפ"ה**__, סמסטר _**א**_, מועד א__
שאלון בחינה בקורס: רשתות מחשבים מתקדם
מספר קורס: 156336.2.5785

שאלת תכנות – 75 נקודות - תכנות TCP Fast open

בקורס למדנו לבצע תכנות של TCP three way handshake באמצעות סקאפי. בשאלה זו תצטרכו לתכנת גרסה מעט שונה, הנקראת TCP Fast open או בקיצור TFO. נתחיל בהסבר תיאורטי קצר ולאחר מכן הדרכה לגבי המימוש.

מטרת ה-TFO לחסוך RTT בגלישה חוזרת לאתר, באמצעות ביטול הצורך ב- three way handshake נוסף. הדיאגרמה הבאה ממחישה כיצד זה מבוצע:



הצד השמאלי הוא שימוש רגיל ב-three way handshake, ביצוע בקשת HTTP GET וחזרה על התהליך בפניה השניה לשרת.

הצד הימני הוא שימוש ב-TFO. שימו לב שהשרת מחזיר cookie בתור מידע מעל פקטת ה-SYN ACK. בפעם הבאה כשהלקוח פונה לשרת הוא מעביר על פקטת ה-SYN גם את ה-cookie וגם את ה- HTTP GET. השרת עונה ב- SYN ACK יחד עם המידע המבוקש.

עד כאן התאוריה.

שנה"ל **תשפ"ה**__, סמסטר _**א**_, מועד א__
שאלון בחינה בקורס: רשתות מחשבים מתקדם
מספר קורס: 156336.2.5785

מימוש- יש צורך לממש 5 פקטות, חלקן בשרת וחלקן בלקוח. לטובת ההסבר, מצורף צילום מתוך wireshark שהקליט פתרון של השאלה. הגדילו את המסך והתבוננו בו היטב. קיראו את כל ההסבר ואת הטיפים למימוש לפני שתתחילו לתכנת.

No.	Time	Source	Destination	Protocol	Length	info
307	5.470303	192.168.1.174	127.0.0.1	TCP	54	12345 → 55555 [SYN] Seq=0 Win=8192 Len=0
308	5.520820	192.168.1.174	192.168.1.174	TCP	58	55555 → 12345 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=4
309	5.530932	192.168.1.174	127.0.0.1	TCP	54	12345 → 55555 [ACK] Seq=1 Ack=1 Win=8192 Len=0
371	6.545016	192.168.1.174	127.0.0.1	TCP	79	12346 → 55555 [SYN] Seq=0 Win=8192 Len=25
372	6.590151	192.168.1.174	192.168.1.174	HTTP	142	HTTP/1.1 200 OK (text/html)

<u>פקטה 1 – SYN – 15 נקודות</u>

הלקוח שתבנו ישלח פקטת SYN.

השרת שתבנו יקלוט אותה באמצעות פילטר מתאים. מספר הפורט שהשרת יאזין אליו נתון לבחירה.

<u>פקטה 2 – SYN ACK – 15 נקודות</u>

השרת ישלח פקטת SYN ACK יחיד עם cookie, מחרוזת כלשהי בגודל 64 ביט.

הלקוח יקלוט את הפקטה באמצעות פילטר מתאים.

<u>פקטה 3 – ACK – 15 נקודות</u>

הלקוח ישלח ACK, השרת יקלוט אותו באמצעות פילטר מתאים.

נניח שלאחר שליחת פקטה 3 התקשורת בין השרת והלקוח מתנתקת וצריך להתחיל מחדש.

<u>פקטה 4 – SYN TFO – 15 נקודות</u>

הלקוח ישלח פקטת SYN הכוללת את ה- cookie ובקשת HTTP GET תקינה לפי הפרוטוקול.
המשאב המבוקש על ידי הלקוח יהיה אחת משתי אפשרויות, אותן יבחר המשתמש:

/Name

/ID

השרת יקלוט את הפקטה ויבדוק אם ה- cookie שלה זהה ל-cookie שהוא העביר ללקוח.
השרת יענה בפקטה מספר 5 אך ורק אם ה- cookie זהה.

<u>פקטה 5 – SYN ACK TFO – 15 נקודות</u>

השרת ישלח ללקוח פקטת SYN ACK ועליה המידע המבוקש לפי פרוטוקול HTTP.

שנה"ל **תשפ"ה**__, סמסטר _**א**_, מועד א___
שאלון בחינה בקורס: רשתות מחשבים מתקדם
מספר קורס: 156336.2.5785

התשובה לבקשת משאב Name תהיה מחרוזת טקסט עם השם שלכם (יש להוסיף את השדות הנדרשים לפי הפרוטוקול).

התשובה לבקשת משאב ID תהיה מחרוזת טקסט עם ת.ז שלכם.

הלקוח יקלוט את התשובה וידפיס למסך את המידע, ללא השדות הנוספים שיש בפרוטוקול.

<u>תזכורת, טיפים למימוש והנחיות כלליות</u>

1. **חובה לקבוע ערכים מתאימים בשדות השונים של הפקטות שהנכם שולחים כדי לוודא שהן אכן מייצגות handshake תקין.** התבוננו היטב בצילום המסך של ההסנפה המצורף לשאלה. שימו לב ש-wireshark לא מתריע על duplicate, out of order וכיוצא בזה. כך צריכה להיות גם התוצאה של הרצת השרת והלקוח שבניתם. ניתן להתעלם מפקטות RST, אם יישלחו כאלה.
2. שלד השרת ושלד הלקוח נתונים
3. השתמשו ב-send בשילוב sniff.
4. שימו לב שכאשר מבצעים sniff עם פרמטר count=1 מתקבלת רשימה באורך 1. כדי לגשת לרשימה צריך לפנות לאינדקס 0, או פשוט להגדיר packet = packet[0].
5. הוספת מידע מעל פקטה מתבצעת באמצעות הוספת שכבת Raw. לדוגמה:
   p = TCP()/Raw(load="Hello")
6. אל תגדירו קבועים שעלולים לדרוס שמות שייבאתם מסקאפי. לדוגמה IP.
7. המתינו שניה בין קבלת פקטה לשליחה של פקטה 4 של הלקוח
8. השרת לא אמור לעבוד מול מספר לקוחות בו זמנית. אפשר להניח שהוא עונה ללקוח יחיד ובסוף שליחת תגובת ה- HTTP השרת נסגר.

יש להעלות כפתרון:

- קובץ שרת
- קובץ לקוח
- צילום מסך מתוך ה-wireshark שיציג את חמשת הפקטות כפי שנקלטו אצלכם

שנה"ל **תשפ"ה**__, סמסטר _**א**_, מועד א__
שאלון בחינה בקורס: רשתות מחשבים מתקדם
מספר קורס: 156336.2.5785

שאלת wireshark – 25 נקודות

בצעו הסנפה לאתר כלשהו שעובד עם TLS1.2. אם אינכם מכירים אתר כזה, נסו מספר אתרים עד שתגיעו לתוצאה הרצויה. ניתן לבחור כל אתר, התוכן אינו משנה רק ה-handshake. במקרים של מגבלות בגלישה מומלץ לבחור אתר בנושא יהדות. לשאלות הבאות *חובה* לצרף גם צילומי מסך. בצילומי המסך הדגישו בצבע את השורה שבה נמצאת התשובה לשאלה. **צילום מסך שלא ברור ממנו היכן התשובה לשאלה – לא יתקבל**.

1. מהו הדומיין אליו ביצעתם את הגלישה? ללא ניקוד
2. באיזה פילטר ניתן להשתמש כדי למצוא את ה-client hello של הלקוח אל השרת שמחזיק את הדומיין הנל? מה מספר הפקטה? 4 נקודות
3. מהו ה-client random? למה הוא משמש? 4 נקודות
4. מהם ה-cipher suites שהלקוח הציע לשרת? 4 נקודות
5. איזו שיטה נבחרה עבור: 4 נקודות
   - הצפנה סימטרית
   - Hash
   - חתימה
   - החלפת מפתחות
6. מהו ה-certificate chain של הדומיין המבוקש? הוכיחו זאת באמצעות צילומי מסך ("א' נתן ל-ב', ב' נתן ל-ג' ") 4 נקודות
7. באיזו שיטה הלקוח מאמת שהסרטיפיקט של השרת הוא בתוקף? CRL? OCSP? OCSP stapling? צרפו סימוכין לטענה. 5 נקודות
8. צרפו את קובץ ההסנפה. דאגו שקובץ ההסנפה יהיה בגודל מקסימלי 2MB (בין אם על ידי הסנפה קצרה ובין אם על ידי חיתוך הקובץ באמצעות hex editor). אין צורך לצרף קובץ מפתחות. ללא ניקוד.