סדנה באבטחת מידע – תרגיל בית 4

לכל בעיה, נא לפנות למייל secws16@gmail.com
תאריך הגשה: 9/12/18
הנחיות:

- שאלות בנוגע לתרגיל יש להפנות למייל הרשום מעלה עם הכותרת hw4question
- ההגשה ביחידים

מטרת התרגיל

- להמשיך ולפתח את חומת האש שלנו.
- להצליח לממש stateful inspection בחומת האש.
- להכיר את ה-payload של פקטות של פרוטוקולים מוכרים ולהיות מסוגלים לחסום או לאפשר אותם מבלי להיחשף לסכנות מיותרות.
- לייצר חומת אש המסוגלת לעבוד בעולם האמיתי.

תיאור התרגיל

חומות האש של היום משתמשות בפונקציות ושיטות מורכבות על מנת לזהות תנועה זדונית ברשת. בתרגיל זה נוסיף ונפתח את חומת האש שלנו, על מנת לטעום מהמורכבות הנדרשת מחומת אש מודרנית. לאחר שהוספנו את האפשרות להגן על הרשת שלנו בצורה בסיסית ע"י stateless packet filtering, נוסיף כעת את האפשרות להגן על הרשת גם ע"י חקירת המידע העובר בפקטה, הסתכלות על ה-payload של הפקטה ולא רק ב-headers. בנוסף ניצור גם טבלת חיבורים דינאמית אשר שומרת את מצבם של החיבורים הקיימים במערכת ובוחנת פקטות על פי החוקים הנכתבים דינאמית לטבלה זו. חומת אש כזו היא חומת אש מסוג stateful inspection.

חלק ראשון: טבלת החיבורים – Connection Table

ההבדל הגדול ביותר בין stateless packet filtering ל-stateful inspection הוא זה של-stateless packet filtering אין "זיכרון" לגבי הפקטות שכבר עברו את חומת האש. ב-stateless packet filtering כל פקטה שעוברת היא אינדיבידואלית ונבחנת אל מול טבלת החוקים, ללא קשר אם היא חלק מחיבור שכבר נפתח או חיבור חדש לחלוטין. stateful inspection לעומת זאת, עוקב אחר מצב החיבורים השונים ברשת: חיבורים חדשים שנוצרים, חיבורים קיימים וכאלה שמסתיימים. בכל פעם שנוצר חיבור **TCP** חדש, אנו נבחן את הפקטה הזו דרך טבלת החוקים הסטטית ונבדוק שהחיבור חוקי. במידה שהחיבור חוקי **נוסיף את החיבור הזה לטבלת החיבורים** עם הפרטים הבאים:

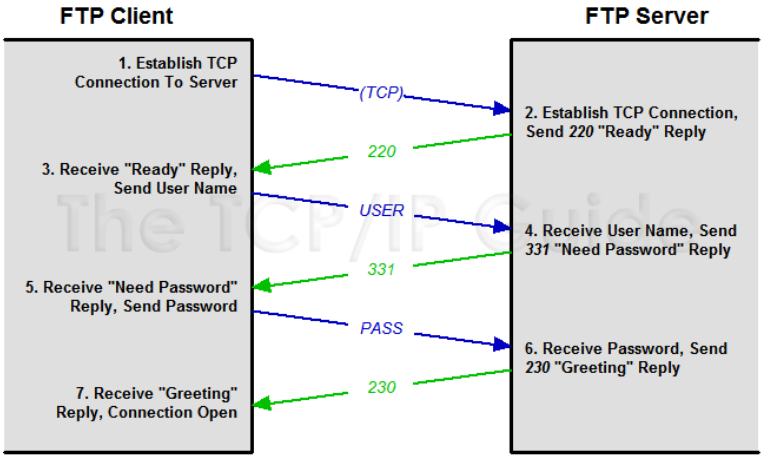
- Source IP address – כתובת המקור
- Source port – פורט המקור
- Destination IP address – כתובת היעד
- Destination port – פורט היעד
- Protocol\State – שדה זה בא לפרט את סוג התקשורת בה אנו עוסקים ובעיקר באיזה מצב נמצא החיבור. לדוגמא, עלינו לדעת אם מדובר בחיבור FTP שכרגע נמצא בשלב לחיצת היד המשולשת, או אולי אנחנו מורידים קובץ מהשרת וזה חיבור של פורט ה-DATA? כל חיבור נמצא במצב אחר, ועלינו לדעת את מצב החיבור על מנת לאפשר תקשורת רלוונטית בלבד. שימו לב שעליכם לתכנן ולתחזק מבני נתונים וטבלאות על מנת שיתמכו במנגנון זה, שכן, לפרוטוקולים מורכבים ישנם הרבה מצבי ביניים אפשריים ואנו צריכים לדעת באיזה מצב נמצא כל חיבור על מנת לאפשר חיבור תקין ומאובטח. תוכן שדה זה נתון לבחירתכם, כולל אם ואיך הוא ימומש: כמצביע לטבלה, כטבלה בעצמו, כמחרוזת, או כל דבר אחר העולה על דעתכם.

כאשר אנו רוצים להוסיף חיבור חדש עלינו לראות אם לחיצת היד המשולשת (3-way handshake) של TCP מסתיימת בהצלחה, לכן לכל חיבור חדש שברצוננו להוסיף נגדיר timeout של 25 שניות, (ברירת מחדל שרירותית, באופן כללי תלוי מערכת הפעלה), שבמהלכן עליו להשלים את תהליך ההתחברות. במידה שהתהליך השלים את ההתחברות בהצלחה, נרשום אותו בטבלת החיבורים. שימו לב שכל פקטת TCP שתגיע תבחן באופן הבא: אם ביט ה-ACK כבוי – נבדוק מול טבלת החוקים הסטטית, ואם ביט ה-ACK דולק, נבדוק בטבלת החיבורים הדינאמית.
כאשר קישור ה-TCP מסתיים, נמחק את הקישור מטבלת החיבורים.
עליכם לממש את טבלת החיברים כפי נכתב לעיל.
הערה: טבלת החיבורים שלנו תתמוך רק בפרוטוקול TCP. כדי לפשט את היישום, על פקטות מסוג UDP, OTHER ו-ICMP לעבור דרך טבלת החוקים הסטטית כפי שמימשנו תרגיל קודם.
הערה: לצורך פישוט הישום והקטנת מספר הבעיות שבהן אתם עלולים להתקל, מומלץ לממש חלק זה לפני שתעברו לחלק הבא של התרגיל.

חלק שני: תמיכה בפרוטוקולים מורכבים

מכיוון שכמות המשאבים בגרעין מערכת ההפעלה קטנה, את חלק זה נממש ב-userspace באמצעות "פרוקסי שקוף". כאשר יגיעו אלינו חבילות השייכות לפרוטוקולים אותם נאכוף, נעביר אותן לתור מיוחד בו הן ישלחו אל תכנת המשתמש שנכתוב ושם נעבד אותן, נבחן אותן ונחליט על גורלן בהתאם. צורת מימוש זאת נותנת לנו כח לאכוף חוקים שבקרנל לא נוכל עקב מגבלות מקום וזמן.

- שימו לב – בחלק זה אתם נדרשים לממש stateful inspection לשני סוגי פרוטוקולים, FTP ו-HTTP **ברמת המשתמש**. על מנת להצליח, תצטרכו להשתמש במבני נתונים שישמרו מידע על החיבורים הקיימים, פונקציות שמקשרות את החיבורים בין הגרעין למשתמש הנכם יכולים לממש זאת בכל דרך שתבחרו. **שימו לב:** כאשר תממשו את התרגיל הזה שימו לב שעבור המשתמש על החיבור להיות שקוף לחלוטין, משמע חומת האש שלכם לא תבצע שינויים **נראים לעין** עבור המשתמשים.
- לצורך שינוי ניווט הפקטות, יש לעשות זאת עבור פקטות נכנסות בPRE_ROUTING ועבור פקטות יוצאות בLOCAL_OUT. כמו כן, יש לתקן את checksum הן של IP Header והן של TCP Header (גם אם לא עושים שינוי בשכבת הTCP). מצורפת דוגמא של שינוי ניווט והתיקונים הנדרשים.
- את החלק בצד הuserspace אתם יכולים לממש בשפות: Python, C, C++, Ruby. אם ברצונכם לממש בשפה אחרת – עשו זאת בתיאום איתי. כמובן, יש לכתוב בתיעוד היבש הוראות הפעלה מפורטות.
- ***HTTP***
  - פרוטוקול ה-HTTP הוא פרוטוקול פופולארי שמשמש אפליקציות רבות, וכמובן משמש לגלישה באינטרנט. בתרגיל זה נרצה להגביל את גודל המידע המועבר אל הרשת הפנימית עבור קבצי Office. על מנת לאפשר זאת, עליכם לעקוב אחרי תקשורת HTTP שמתבצעת בפורט 80, ולשים לב לתחילת הקובץ (Magic) ול-HTTP headers. ממשו מנגנון שייבחן את המידע ויאפשר\יחסום בהתאם.
  - ***זיהוי גודל המידע המתקבל:*** על ידי סריקת הheader בשם Content-Length המופיע בresponse עבור הרשת הפנימית. אם הוא לא מופיע כלל – עלינו לחסום את העברת המידע. אם מופיע, והערך שלו גדול מ2000 בתים – עלינו לבדוק את סוג הקובץ.
  - ***זיהוי סוג הקובץ:*** אם הMagic מתאים לקובץ Office, עלינו לחסום את העברת המידע. שימו לב שמעניין אותנו לחפש את הMagic אך ורק בתחילת הקובץ.
- ***FTP:***
  - כפי שלמדתם, פרוטוקול FTP נחשב לפרוטוקול בעייתי, כך שחומות אש פרמיטיביות מסוג stateless packet filtering מסוגלות לתמוך בו אך ורק במחיר של פגיעה חמורה ברמת האבטחה, שכן כל מחשב יוכל לגשת לרשת דרך מס' פורטים רב והגישה תאושר. לעומת זאת, בחומת אש מסוג stateful inspection ביכולתינו לקרוא את שדות האפליקציה בפקטה, להבחין שמדובר ב-FTP ולאשר את החיבורים הצפויים מבעוד מועד. עליכם בתרגיל זה לספק תמיכה ל-active FTP בחומת האש שלכם.
  - *זיהוי:* לאחר לחיצת היד המשולשת שנעשת בפרוטוקול TCP עם השרת בפורט 21 (פורט שרת ה-FTP), החיבור מתבצע באופן



המתואר בתמונה:
על מנת לזהות חיבור FTP, עקבו אחר לחיצת היד המשולשת ושימו לב לפורט ולקוד המגיע מהשרת. במידה שהקוד תקין, נוצר חיבור FTP חדש. על מנת לתמוך ב-FTP אתם יכולים לראות את רשימת ה-status codes כאן:
http://en.wikipedia.org/wiki/List_of_FTP_server_return_codes. בנוסף, מצורף לכם באתר הקורס קובץ pcap של הקלטה של התקשרות עם שרת FTP. הורידו אותו ושימו לב לצורת התקשורת. למידע כיצד לעבוד עם פילטרים ב-wireshark, הנכם מוזמנים לקרוא כאן: http://wiki.wireshark.org/DisplayFilters
  - **הערה:** על מנת לבדוק את חומת האש שלכם, הורידו ל-host2 שרת FTP והתחברו אליו מ-host1, ולאחר מכן נסו לשלוח/לקבל קבצים דרך השרת.
  - *בונוס*, ברצונינו לא לאפשר קבלת קבצי הרצה (exe) משרת FTP. לצורך כך, עליכם לבדוק כל קובץ המועבר משרת FTP אל הרשת הפנימית. בדיקה זו נעשית ע"י בדיקת מחרוזת הmagic של קבצי הרצה (MZ).

- גישה לפרוטוקולים אחרים של TCP יעבדו כרגיל, משמע דרך טבלת החיבורים **בלבד**. כל פקטת TCP תעבור צעד ראשון דרך טבלת החיבורים, ואם הפקטה היא פקטה של **הפרוטוקולים הספציפיים** שנכתבו מעלה, הסטטוסים שתגדירו בטבלת החיבורים יפנו אותה אל ה-userspace לבחינה מעמיקה ע"י stateful inspection.

ממשק הניהול בקרנל

עליכם להוסיף לממשק הניהול שכתבתם בתרגיל 3 התקים חדשים בשם conn_tab וhosts. פונקציונליות ההתקנים יהיו כדלקמן:

name - all below have the perfix /sys/class/fw/	***Perm***	***Functionality***
`fw/conn_tab`	R	Reading from the file will return the table. We do not need it to be beautiful, we have user-space program for this, we just need it to transfer the data to the user interface.

ממשק הניהול בתוכנית משתמש

יש להרחיב את תוכנית הניהול ב-userspace כדלקמן.

- ברצוננו לראות את טבלת החיבורים הנוכחית בכל רגע נתון, ולכן עלינו להוסיף תמיכה לצפיה ב-connection table. בנוסף לפונקציות של תרגיל 3, הוסיפו לתוכנית את האפשרות לקרוא לפקודה `show_connection_table`, שתדפיס את טבלת החיבורים בפורמט הבא:

```
<Source_IP> <Source_port> <Dest_IP> <Dest_port> <protocol>
```

  - **הערה:** הנכם יכולים להוסיף עוד שדות בהמשך במידת הצורך. רק שימו לב שחמשת השדות הראשונים יהיו אלו.
  - **הערה:** מומלץ לממש חלק זה יחד עם החלק הראשון על מנת להקל עליכם בדיבוג התוכנית.

הגשה

הכינו קובץ zip שיכיל בדיוק שתי תיקיות וקובץ:

1. firewall – תיקייה עם כל הקבצים הנדרשים לקימפול המודול שלכם יחד עם קובץ Makefile.
2. interface – תיקייה עם התוכנה שדואגת לניהול חומת האש מצד המשתמש, יחד עם קובץ Makefile.
3. תיעוד יבש – מסמך PDF המסביר את הקוד שכתבתם ברמת המערכת הכללית. בנוסף, יש לתת הסבר קצר על כל אחת מהפונקציות שבחרתם לממש. חלק זה עומד בפני עצמו, ואמור להיות ברור לקורא גם ללא עיון בקוד.

```
<Submission_format>.zip -->
            firewall -->
                  fw.c fw.h … Makefile
            interface -->
                  main.c … Makefile
            hw4_dry.pdf
```

בהצלחה!