הטכניון - מכון טכנולוגי לישראל

הפקולטה למדעי המחשב סמסטר אביב תשפ״ב

הגנה ברשתות 236350

תרגיל בית מס׳ 3

הגשה: עד יום ד׳, 18/05/2022, 23:59

הגשה ביחידים

חל איסור חמור על החזקת פתרונות של סטודנטים אחרים. על כל סטודנט לרשום את תשובותיו **עצמאית** ובמילותיו שלו.

נא להגיש את התרגילים אלקטרונית בלבד

בנוגע לשאלה 1 נא לפנות להדר (hadarsivan@cs)

בנוגע לשאלה 2 נא לפנות לטל (talneoran@cs)

בתרגיל זה חלק רטוב בו תידרשו לכתוב קוד ולצורכי דיבאג תוכלו להיעזר בכלים איתם עבדתם בתרגיל הבית הקודם. עבור החלק הזה תשתמשו במכונה הווירטואלית cs_236350.ova איתה עבדתם בתרגיל הקודם.
הוראות ההתקנה של המכונה הוירטואלית מופיעות בתרגיל הבית הקודם.
על ה-VM מותקנות כל החבילות והכלים הדרושים לפתרון התרגיל. בתוך התיקייה ~/cs_236350/dns_spoofing_and_mitm/ על ה-VM תמצאו את הקבצים הדרושים לחלק זה. אנו ממליצים לפתח על ה-VM. אתם יכולים להתקין על המכונה PyCharm לצורכי פיתוח וכתיבת קוד:

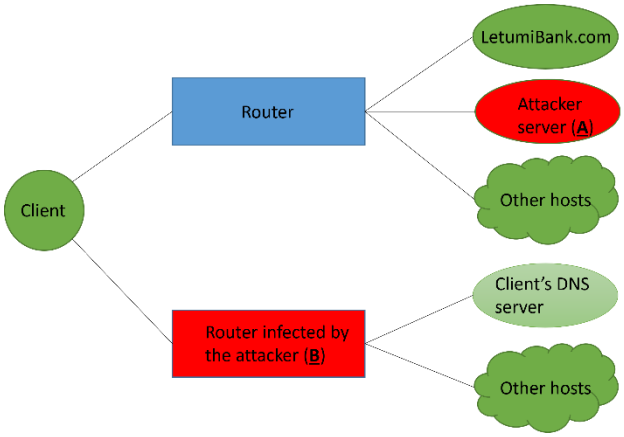
```
sudo snap install pycharm-community --classic
```

שאלה 1 – DNS Spoofing ו-Man in the Middle

בתרגיל זה אתם תגלמו את תפקידו של תוקף רשת המשכנע דפדפן רשת של קורבן לגלוש לשרת של התוקף במקום לאתר אליו רצה לגלוש. התוקף למעשה מבצע מתקפת Man in the Middle בשביל להעביר את הודעות הקורבן אל ומהאתר מבלי להתגלות, בעודו גונב מידע סודי. תצטרכו לזייף הודעת תשובה משרת ה-DNS על מנת לשכנע את הקורבן לגלוש לאתר LetumiBank.com עם כתובת ה-IP של התוקף, במקום עם כתובת ה-IP האמתית של LetumiBank.com. אתם תשתמשו בספריית Python הנקראת scapy על מנת לבצע מניפולציה של ה-headers של פקטות בצורה מהירה ואמינה. ברגע שתשובת ה-DNS תזויף בהצלחה, אתם תקבלו קשר אל הקורבן ותעבירו את כל הבקשות שלו אל ומשרת הרשת האמתי LetumiBank.com.

לצורך כך כתבנו תוכניות Python עבור הקורבן והתוקף. עליכם לשנות את תוכניות התוקף כדי לבצע בהן את ההתקפה. לצורך נוחות איחדנו את שתי תוכניות התוקף לתוכנית אחת שבה ייעשו כל השינויים. תוכנית התוקף כוללת חלקי קוד שעליכם להשלים. תוכנית הקורבן כוללת בתחילתה תרגום של כתובת אתר הבנק לכתובת IP ע"י פניה לשרת DNS ולאחר מכן תקשורת עם אתר הבנק.

טופולוגיית המערכת מופיעה למטה. עבור מטלה זו, נניח שהתוקף הצליח להדביק נתב רשת שנמצא בין הקורבן ובין שרת ה-DNS בו משתמש הקורבן, ולכן התוקף יכול לבצע רחרוח (sniffing) על פקטות הנשלחות מהקורבן לשרת ה-DNS ולמנוע משרת ה-DNS לשלוח תשובות בחזרה לקורבן.



עליכם לבצע את הפעולות הבאות:

1. לפתוח TCP socket המאזין לפורט 8000 (מוגדר ע"י המשתנה WEB_PORT) על מכונת התוקף (A) על מנת לקבל קשרים חדשים מהקורבן.
2. ב-B עליכם לבצע sniffing לחבילות DNS העוברות ברשת. השתמשו בפונקציית sniff() של scapy בשביל לאתחל פונקציית callback שתיקרא כאשר מזוהה חבילת DNS. ה-callback הנ"ל אמור להגיב לבקשות DNS עם תשובת DNS מזויפת אשר תנתב את הקורבן לכתובת ה-IP של A.
3. לאחר שליחת תשובת ה-DNS המזויפת, חכו לקורבן שיתחבר לפורט 8000 של A וקבלו בקשות HTTP על ה-TCP socket שיווצר.
   a. על התוקף (A) לקרוא את תוכנה של כל הודעה, לבדוק האם ההודעה מכילה מידע חסוי, ולתעד מידע חסוי זה ע"י כתיבתו לקובץ. לצורך כך תחפשו הודעות מסוג POST עם פרמטרים username ו-password ותקראו לפונקציה log_credentials() הנתונה לכם בכדי לרשום את המידע החסוי בקובץ.
   b. לאחר מכן אתם תעבירו את התוכן המקורי של בקשת ה-HTTP לשרת האמתי LetumiBank.com (שאת כתובת ה-IP שלו יש להשיג ע"י קריאה לפונקציה resolve_hostname() הנתונה לכם), ותעבירו את תשובת השרת בחזרה לקורבן. במידה שביצעתם את העברת ההודעות כראוי בין השרת והקורבן, תוכן הקובץ שהוריד הקורבן מהשרת, שהקורבן שומר אצלו ב-client/lib/downloadedPage.txt, יהיה תואם לתוכן הקובץ המקורי httpServer/lib/fileToDownload.txt בשרת.
4. סגרו את הקשר וצאו מהתוכנית ברגע שהשרת LetumiBank.com והקורבן סיימו את התקשורת ביניהם. הקורבן מודיע על סיום ה-session שלו ע"י בקשת POST ל-post_logout/.

מידע נוסף

- **אנו מספקים לכם מגוון קבצי Python שיש להריץ עבור הקורבן, השרת והתוקף.**

- **שינויי קוד נדרשים אך ורק בקובץ attacker/attacker.py, והגישו רק אותו. אין לשנות קבצים אחרים ואין ליצור קבצי קוד נוספים!**
- **ספריות ה-Python אותן אתם צריכים כבר מותקנות על ה-VM. אל תתקינו ספריות נוספות.**

- לכל השרתים בתרגיל יהיו כתובות IP מקומיות, והם יתקשרו לוקלית מעל ממשק הרשת הלוקלי הנקרא loopback (או 'lo'). הרצת run.sh/. תאתחל את טופולוגיית המערכת, תקצה כתובות IP לקורבן, שרת ה-DNS, שרת הווב והתוקף, ותריץ את התהליכים שלהם. שימו לב: הלקוח מתרגם את ה-hostname של שרת הרשת, ומתחבר אליו בצורה אוטומטית (ראו את client.py לפרטים נוספים).

- לצורך תרגיל זה אנו מניחים שהתוקף שולט בנתב בין הקורבן ושרת ה-DNS שלו. לכן לא יהיה race condition בין תשובת התוקף לקורבן לתשובת שרת ה-DNS. שימו לב שבמציאות (ללא השתלטות על נתב) לרוב זהו לא המצב.

- השתמשו ב-Wireshark בכדי לדבג את ממשק הרשת הלוקלי 'lo'. התנסיתם מעט ב-Wireshark בתרגיל הקודם. היכולת לראות בדיוק אילו חבילות נשלחות מעל הרשת מקלה מאד על הדיבוג.

- פונקציית sniff() של scapy יכולה לאתחל callback שנקרא כאשר פקטות מסוימות מזוהות על הרשת. כדי להעביר פרמטרים נוספים ל-callback אתם יכולים להשתמש בפונקציית lambda.
לדוגמה: cb = lambda org_arg: callback(org_arg,(extra_arg1,extra_arg2))

- scapy בונה פקטות ע"י בניה של כותרות כל שכבת רשת בנפרד, ואז מחברת אותן יחד. כאשר אתם מעבירים שדות שונים עבור ה-headers אז scapy משלים את השדות שלא העברתם ע"י בחירת ערכי ברירת מחדל עבורם. ניתן למצוא את הדוקומנטציה של scapy בקישור https://scapy.readthedocs.io/en/latest, אבל היא לא מספקת הרבה אינפורמציה על פרמטרים ספציפיים, ולכן אנו מספקים לכם כמה headers עם ערכים אפשריים. אתם לא בהכרח חייבים להשתמש בכל ה-headers או בכל רשימת הפרמטרים.

```
IP(src=source_ip, dst=dest_ip, proto=protocol, ttl=TTL)
TCP(sport=source_port, dport=dest_port, flags=TCP_flags(A=ack, etc.))
UDP(sport=source_port, dport=dest_port)
DNS(id=id, qd=DNSQR(query), an=DNSRR(answer), qr=0/1, aa=0/1, ...)
DNSRR(rdata=IP_address of host, rrname=host.com, ttl=TTL)
```

פרמטרים נוספים של DNS header ומה מייצגים הפרמטרים ניתן למצוא בקישורים הבאים:
https://en.wikipedia.org/wiki/Domain_Name_System,
http://www.networksorcery.com/enp/protocol/dns.htm

- השתמשו בספריית socket בה השתמשתם בתרגיל בית 1 כדי ליצור את שני הסוקטים – אחד עבור הקשר בין הקורבן והתוקף והשני עבור הקשר בין התוקף ושרת הבנק:
https://docs.python.org/3/library/socket.html.

- שימו לב ש-HTTP 1.0 משתמש בקשר TCP חדש עבור כל בקשה; מוזמנים לוודא זאת ע"י Wireshark!

עליכם להגיש את הקובץ attacker.py המלא שלכם ואת הקובץ הנוצר בתיקיית attacker/lib.

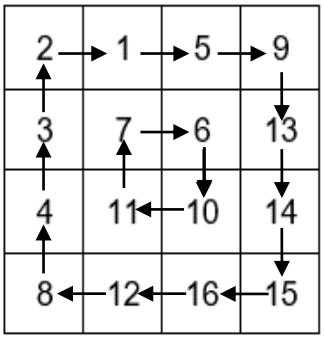
שאלה 2 – AES, Modes of Operation

שאלה זו עוסקת בגרסה של AES שנלמדה בתרגול עם מפתח בגודל 128 ביט.

1. סטודנט בקורס 'מבוא לקריפטוגרפיה' רצה להרשים את המרצה ולשפר את צופן AES. הסטודנט הציע את השינוי הבא:
במקום לבצע את פעולת ה-**Mix Columns**, נבצע פעולה חדשה שנקראת **Rotate Clockwise**.
הפעולה הזו מבצעת הזזה של כל תא בטבלה פעם אחת עם כיוון השעון. להלן תיאור של הפעולה.
החיצים שעל הטבלה מציינים את כיוון ההזזה:

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Rotate Clockwise →



השוו בין עמידות הצופן החדש לעמידות הצופן המקורי כנגד התקפת chosen-plaintext. אם הצופן החדש פחות בטוח, תארו התקפת chosen-plaintext כנגדו, ציינו מהו מספר הזוגות (P, C) הדרוש עבור ההתקפה, ציינו כיצד נבחרים הזוגות בהתקפה, וציינו מה סיבוכיות המקום והזמן של ההתקפה. אם הבטיחות כנגד ההתקפה לא השתנתה הסבירו מדוע.

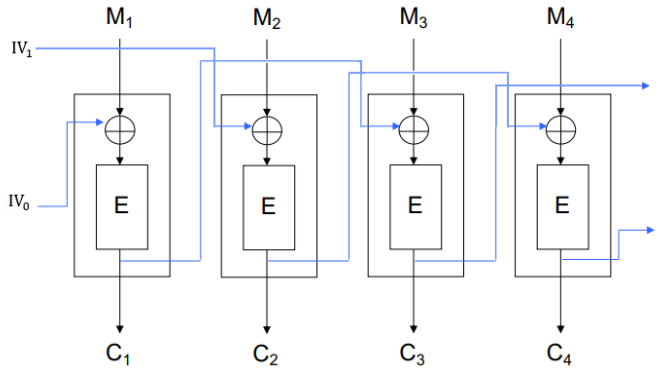
2. המרצה הגיב באדישות כשהציג הסטודנט את ההצעה בפניו, ולכן במאמץ נוסף להרשים אותו, הציע הסטודנט שינוי **אחר** (לא בנוסף לשינוי הקודם): כל הפעולות יהיו זהות לאלגוריתם המקורי, פרט לפעולת ה-**Byte Substitution**, אותה נחליף בפעולה **Rotate Clockwise**. הסטודנט טען שהצופן שלו טוב יותר מ-AES המקורי כי כאן אין טבלת החלפה (S-box) שכולם צריכים להכיר ולשמור. חוו דעתכם על בטיחות הצופן החדש לעומת הצופן המקורי. בתשובתכם עליכם לציין האם הצופן חלש ביחס ל-AES המקורי או האם הוא חזק לפחות כמו AES המקורי. נמקו.

3. המרצה חזר למשרדו כדי לקבל קצת שקט. במשרד הוא נזכר בפעולת ה-**Rotate Clockwise** שהרשימה אותו ורצה בכל זאת להכניס אותה איכשהו למנגנון AES. המרצה חשב על הרעיון הבא: נשתמש בגרסה המקורית של AES אך לפני כל ביצוע של פעולת **Key Mixing**, נבצע את פעולת ה-**Rotate Clockwise** על תת המפתח שהיא מקבלת כקלט. חוו דעתכם על בטיחות הצופן החדש לעומת הצופן המקורי. בתשובתכם עליכם לציין האם הצופן חלש ביחס ל-AES המקורי או האם הוא חזק לפחות כמו AES המקורי. נמקו.

הסעיפים הבאים אינם קשורים לסעיפים הקודמים.

4. באביב 2020 התפרסמו דוחות אודות שימוש בהצפנה לא מאובטחת על ידי אפליקציית זום. לאחר חקירת התוכנה, התגלה שמתבצע שימוש באלגוריתם AES-128 באופן תפעול ECB להצפנת תוכן אודיו ווידאו בפגישות זום.
   a. ציינו מהי התכונה של ההצפנה שהובילה לטענות החוקרים שהפגישות אינן מאובטחות. הסבירו את תשובתכם.
   b. הציעו שינוי שחברת זום יכלה לבצע על מנת לפתור את בעיית האבטחה.

5. בכל אחד משני תתי-הסעיפים הבאים עליכם לבחור בקפידה שתי הודעות, באורך שני בלוקים של AES כל אחת, ולתאר כיצד בהינתן ההצפנה של אחת מהן תוכלו לדעת איזו אחת היא זו שהוצפנה. ההצפנה מתבצעת עם אלגוריתם AES עם מפתח שאינו ידוע לכם, ובאופן התפעול שמוצג בתת-הסעיף.
   a. ECB.
   b. Interleaved-CBC המוגדר באופן הבא:



אתחול: $C_{-1} = IV_0,\ C_0 = IV_1$

הצפנה: $C_i = E_k(M_i \oplus C_{i-2})$

פענוח: $M_i = D_k(C_i) \oplus C_{i-2}$

כאשר IV_0 נבחר באופן אקראי ו- $IV_1 = IV_0 \oplus 24$.

6. לרועי ויורי יש מערכת המצפינה הודעות שהיא מקבלת בעזרת אלגוריתם ההצפנה AES, באופן תפעול CBC תחת מפתח הידוע לשניהם בלבד. לאחר שידור ההודעה המוצפנת בצירוף ערך ה-IV, המערכת מגרילה את ערך ה-IV בו תשתמש להצפנה הבאה, ומשדרת אותו בגלוי. רועי השתמש במערכת בשביל לשלוח ליורי הודעה בגודל בלוק המכילה את המילה "כן" או את המילה "לא" (מדופן עם אפסים לגודל בלוק) ושלח את התוצאה ליורי. עומר מאזין לתקשורת בין רועי ליורי ויודע שההודעה שהוצפנה היא אחת מבין "כן" ו"לא". נניח שעומר יכול לגרום כעת למערכת להצפין הודעה כרצונו במקום ההודעה הבאה של רועי, מוצפנת תחת המפתח אשר עומר כאמור אינו יודע וה-IV הבא שכבר פורסם. איזו הודעה יכול עומר לבקש מהמערכת לשלוח על מנת לדעת איזו הודעה רועי הצפין ("כן" או "לא")? נמקו.

הוראות הגשה: עליכם להגיש קובץ zip יחיד המכיל שלושה קבצים:

1) את קובץ ה-PDF עם התשובה שלכם לשאלה 2.
2) את הקובץ attacker.py המלא שלכם (נא לא לשנות את שם הקובץ משום שתתבצע בדיקה אוטומטית).
3) את הקובץ הנוצר בתיקיית attacker/lib.

וודאו שהקבצים הנ"ל נמצאים ב-root של ה-zipped folder (ולא בתוך תיקייה פנימית).