הטכניון - מכון טכנולוגי לישראל

הפקולטה למדעי המחשב | סמסטר חורף תשפ״ד

הגנה ברשתות 02360350

תרגיל בית מס׳ 3

הגשה: עד יום ה׳, 07/03/2024, 23:59

הגשה ביחידים בלבד

חל איסור חמור על החזקת פתרונות של סטודנטים אחרים.
עליך לרשום את התשובות **עצמאית** ובמילים שלך.
נא להקליד את התשובות ולהגישן אלקטרונית בלבד.

בנוגע לשאלה 1 נא לפנות ל- דיאנה (diana.cohen@cs.technion.ac.il)
בנוגע לשאלה 2 נא לפנות ל- תום (tomazoulay@cs.technion.ac.il)

בתרגיל זה חלק רטוב בו תידרשו לכתוב קוד ולצורכי דיבאג תוכלו להיעזר בכלים שאיתם עבדתם בתרגיל הבית הראשון. ניתן להשתמש במכונה הווירטואלית (פרטים מטה), עליה מותקנות כל החבילות והכלים הדרושים לפתרון התרגיל.

בתוך התיקייה ~/cs_236350/dns_spoofing_and_mitm על ה-VM תמצאו את הקבצים הדרושים לשאלה זו. לצורכי פיתוח וכתיבת קוד, על המכונה מותקן VSCodium הזהה ל-VS Code, אך אין בו מעקב אוטומטי של מיקרוסופט. אם בחרת לא לעבוד עם המכונה, ניתן להוריד את ה-zip המופיע בצמוד לתרגיל הבית. ההמלצה שלנו לפתח על ה-VM.

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

1. את קובץ ה-PDF עם התשובות לשאלה 1 חלק 1 ושאלה 2.
2. את הקובץ attacker.py המלא (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).

על מנת להריץ את ה-VM:

1. התקינו VirtualBox – בלינק הבא מומלץ לבחור את תת-הגרסה העדכנית ביותר של גרסה יציבה, המתאימה למערכת ההפעלה שברשותך:
   https://www.virtualbox.org/wiki/Downloads
   אין להשתמש ב-EXTENSION PACK, זה מנוגד לרישוי הטכניוני!
   ניתן להשתמש בתוכנת וירטואליזציה אחרת, אך ההוראות כאן הן עבור VirtualBox.

2. הורידו את ה-image של ה-VM אל המחשב האישי (ההורדה עלולה לקחת כשעה כתלות ברשת): הקישור מופיע בצמוד לתרגיל הבית והוא פתוח לחשבונות טכניוניים בלבד.

3. פתחו את קובץ ה-ova ב-VirtualBox. הוראות לפתיחת image בעזרת virtual box נמצאות בלינק הבא (אין צורך לבצע שינויים בהגדרות ברירת המחדל במסך appliance settings):
   https://www.alphr.com/ova-virtualbox

4. לחצו על start להפעלת ה-VM. תידרשו להכניס סיסמה. הסיסמה זהה לשם המשתמש:
   cs236350
   שימו לב: ייתכן שתופיע הודעת שגיאה הקשורה ל-network interface. בחרו ב- Change Network Setting ואז OK במסך הבא. אם עדיין אין למכונה גישה לאינטרנט, אפשרו אותה ע״י מעקב אחר ההוראות בקישור הבא: https://linuxhint.com/enable-internet-virtualbox

5. על מנת לאפשר העתקת קבצים מה-VM ל-host ולהיפך:
   i. כבו את ה-VM.
   ii. צרו תיקייה ב-host בשם shared אותה תירצו למפות ל-VM.
   iii. בהגדרות ה-VM בחרו Settings -> Shared Folders ולחצו על הסימן '+'.
   הוסיפו את התיקייה shared וסמנו Auto-mount.
   iv. לאחר שתפעילו את ה-VM התיקייה המשותפת היא `/media/sf_shared`

שאלה 1 – התקפות על DNS ו-Man in the Middle

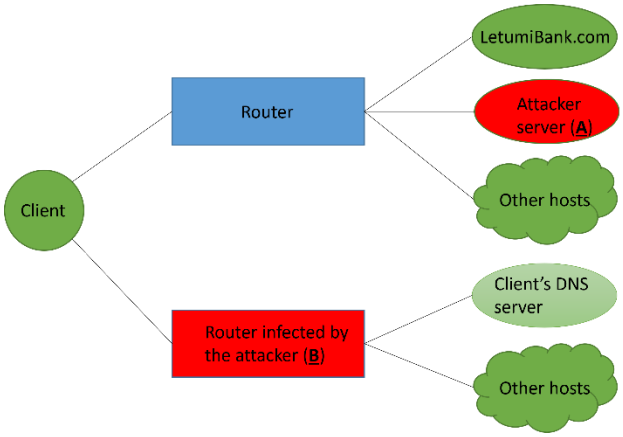
חלק 1:
בשנת 2008 הוצגה שיטת התקפה לביצוע DNS poisoning ע"י חוקר בשם דן קמינסקי ז"ל.
קראו את ההסבר על שיטה זו ב- http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

1) הסבירו את שיטת ההתקפה הזו ופרטו במה שונה ההתקפה בהשוואה להתקפה שתוארה בהרצאה (שקף מס' 227).
2) מה היתרונות של ההתקפה בהשוואה להתקפה שתוארה בהרצאה?
3) מה הגנה המרכזית שהוצעה כדי להתמודד עם ההתקפה?
הסבירו איך היא מונעת את הצלחת ההתקפה.

חלק 2:
בחלק זה תגלמו את תפקידו של תוקף רשת שמשכנע את הדפדפן של הקורבן לגלוש לשרת של התוקף במקום לאתר שאליו 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 האמיתית של LetumiBank.com. יש להשתמש בספריית Python הנקראת scapy על מנת לבצע מניפולציה על ה-headers של פקטות בצורה מהירה ואמינה. ברגע שתשובת ה-DNS תזויף בהצלחה, תקבלו קשר אל הקורבן ותעבירו את כל הבקשות שלו אל ומשרת הרשת האמתי LetumiBank.com.

לצורך כך, כתבנו תוכניות Python עבור הקורבן והתוקף. עליך 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 שעליך 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 באיור מטה. 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.



עליך לבצע את הפעולות הבאות:
4) לפתוח TCP socket המאזין לפורט 8888 (מוגדר ע"י המשתנה WEB_PORT) על מכונת התוקף (A) על מנת לקבל קשרים חדשים מהקורבן.

5) ב-(B) עליך לבצע sniffing לחבילות DNS העוברות ברשת. יש להשתמש בפונקציית sniff של scapy בשביל לאתחל פונקציית callback שתיקרא כאשר מזוהה חבילת DNS.
ה-callback הנ"ל אמורה להגיב לבקשות DNS עם תשובת DNS מזויפת אשר תנתב את הקורבן לכתובת ה-IP של A.

6) לאחר שליחת תשובת ה-DNS המזויפת, יש להמתין לקורבן שיתחבר לפורט 8888 של A ולקבל בקשות HTTP על ה-TCP socket שייווצר.
   i. 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 יש לחפש הודעות מסוג POST עם הפרמטרים username ו-password ולקרוא לפונקציה log_credentials הנתונה לך כדי לרשום את המידע החסוי לקובץ.

ii. לאחר מכן יש להעביר את התוכן המקורי של בקשת ה-HTTP לשרת האמיתי LetumiBank.com (שאת כתובת ה-IP שלו יש להשיג ע"י קריאה לפונקציה resolve_hostname הנתונה לך), ולהעביר את תשובת השרת בחזרה לקורבן.
אם ביצעת 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 שבשרת.

7) יש לסגור את הקשר ולצאת 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

- **סיפקנו לך מגוון קבצי Python שיש להריץ עבור הקורבן, השרת והתוקף.**
- **שינויי קוד נדרשים אך ורק בקובץ attacker/attacker.py, ויש להגיש רק אותו. אין לשנות קבצים אחרים ואין ליצור קבצי קוד נוספים!**
- **ספריות ה-Python הנחוצות מותקנות על ה-VM. אין להתקין 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.
לשים לב שהלקוח מתרגם את ה-hostname של שרת הרשת, ומתחבר אליו בצורה אוטומטית (ראו את client.py לפרטים נוספים).

- לצורך תרגיל זה הנחנו שהתוקף שולט בנתב בין הקורבן ושרת ה-DNS שלו. לכן, לא יהיה race condition בין תשובת התוקף לקורבן לבין תשובת שרת ה-DNS. במציאות (ללא השתלטות על נתב) לרוב זהו לא המצב.

- יש להשתמש ב-Wireshark כדי לדבג את ממשק הרשת הלוקלי 'lo'. התנסית מעט ב-Wireshark בתרגיל הראשון. היכולת לראות בדיוק אילו חבילות נשלחות מעל הרשת מקלה מאד על הדיבאג.

- פונקציית sniff של scapy יכולה לאתחל callback שנקרא כאשר פקטות מסוימות מזוהות על הרשת. כדי להעביר פרמטרים נוספים ל-callback יש 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. לאחר העברת שדות שונים עבור ה-headers, אזי scapy משלים את השדות שלא הועברו 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 סיפקנו לך כמה headers עם ערכים אפשריים. אין הכרח 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

- יש להשתמש בספריית socket של Python 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.

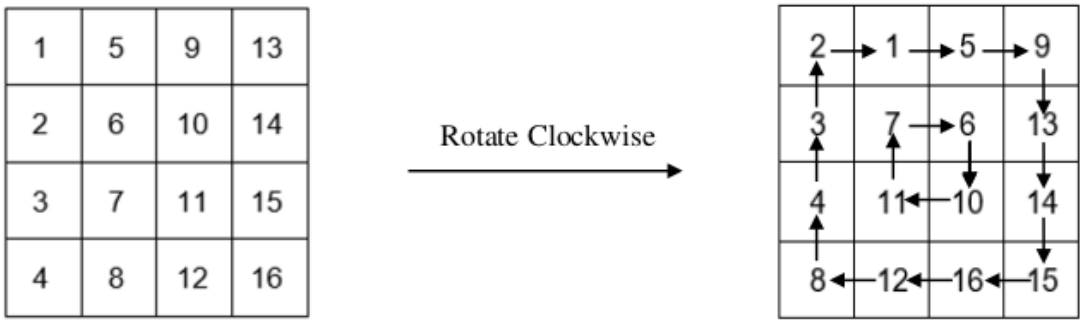
- לשים לב ש-HTTP 1.0 משתמש בקשר TCP חדש עבור כל בקשה; כדאי לוודא זאת ע"י Wireshark!

עבור חלק זה עליך להגיש את הקובץ attacker.py המלא 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:



השוו בין עמידות הצופן החדש לעמידות הצופן המקורי כנגד התקפת chosen-plaintext. היעזרו לתרגול 5.
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 את 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 את 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. מומלץ להיעזר בחוברת הקורס.

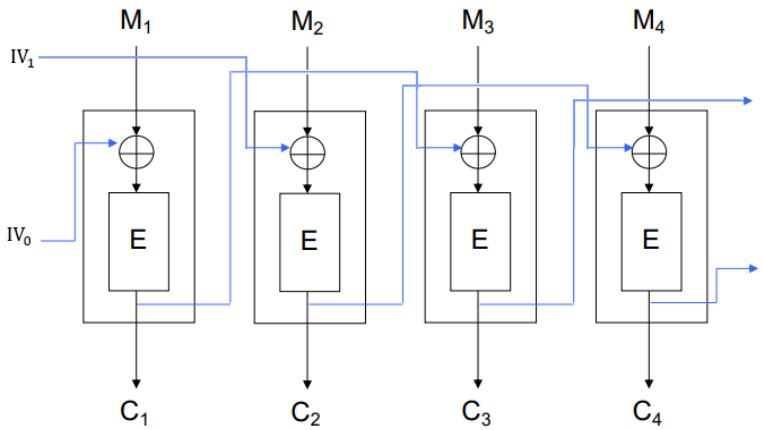
i. ציינו מהי התכונה של ההצפנה שהובילה לטענות החוקרים שהפגישות אינן מאובטחות. הסבירו את תשובתכם.

ii. 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.

i. ECB.

ii. 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-$IV_1 = IV_0 \oplus 27$.

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 האפשרויות "כן" או "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.