ARP spoof warning

הקדמה – איך מזהים ARP Spoof

בשיעורי הבית הקודמים התבקשנו לפתח קוד שמבצע ARP Spoof Attack. השבוע התבקשנו לבנות כלי שמזהה ומרתיע שהמכונה שלך מותקפת.
אנחנו חשבנו על שלוש שיטות שאיתן ניתן לזהות תקיפה זו:

1. ע"י PING למחשב שחושדים בו - ע"י שימוש באחת מן האפשרויות בפרוטוקול ICMP - echo request-replay
2. ע"י שליחת בקשת ARP לגיטימית למחשב החשוד.
3. ע"י הקמת שיחת TCP עם המחשב החשוד.[1]

נקודות חשובות לפני שמתחילים

- תחילה נציין שכאשר אנו אומרים "המחשב החשוד" אנו מתכוונים לכך שאנו חושדים שכתובת ה-MAC שבתשובת ה-ARP שהגיעה לקורבן אינה כתובת ה-MAC של המחשב האמיתי שכתובת ה-IP שלו מופיע בתשובה.
- הגשנו שני קבצים:
  - הקובץ ששמו $arpSpoofPrevention$ הינו הקובץ שירוץ בקורבן שיזהה ויתגבר על ההתקפה.
  - הקובץ ששמו $arpSpooferOvercomeDetection$ הינו הקובץ שירוץ אצל התוקף שאמור להתגבר על אמצעי הזיהוי של הקורבן
  - ביצענו את הבונוס הראשון בצורה מושלמת – הקובץ הראשון שציינו לא רק מזהה את המתקפה אלא גם מתגבר עליה, ואת הבונוס השני בצורה חלקית – הצלחנו לזהות את האמצעים שבהם הקורבן מנסה להתגבר על המתקפה שלנו – אך לא הצלחנו להתגבר עליהם.

אופן הפעולה

בקובץ $arpSpoofPrevention$:
בכללי - הפונק' `thwart_arp_spoof`:[2] בשני תרדים נפרדים הרצנו את כל אחת מן האפשרויות שמזהה את המחשבים החשודים. הבדיקה והפעולה הבאה מתבצעות בלולאה אין סופית כל 5 שניות: אם ישנו מחשב חשוד ששתי השיטות סימנו אותו כחשוד – כלומר הוא נמצא ברשימת החשודים של שניהם – אזי אנחנו שולחים פקודה אשר חוסמת רשומה זו בטבלת ה-ARP של הקורבן ומודיעה לו על כך.

ועכשיו בפירוט – איך עשינו את זה בפועל:

זיהוי ראשון – עם ARP

הסנפנו את כל התעבורה המגיעה למחשב הקורבן[3] ופילטרנו רק תשובות ARP replay.[4]
למעשה בפונק' process(pkt)[5] קיבלנו פאקטה שאומרת, לדוגמא, שהמחשב שכתובת ה-IP שלו היא X נמצא בכתובת Y MAC.
בתרגיל זה לקחנו את התשובה הזו בעירבון מוגבל ולכן שלחנו בעצמנו בקשת ARP כדי לבדוק אם הזוג הזה אמיתי או מתחזה.
לכן שלחנו בקשת ARP כדי לקבל את כתובת ה-MAC של המחשב שכתובת ה-IP שלו היא X, ובדקנו האם כתובת ה-MAC שקיבלנו היא באמת Y. אם לא – סימן שהתשובה שהסנפנו היא של מתחזה ולכן נוסיף את הזוג המתחזה לרשימת החשודים, במידה והוא לא כבר שם.

זיהוי שני – עם ICMP

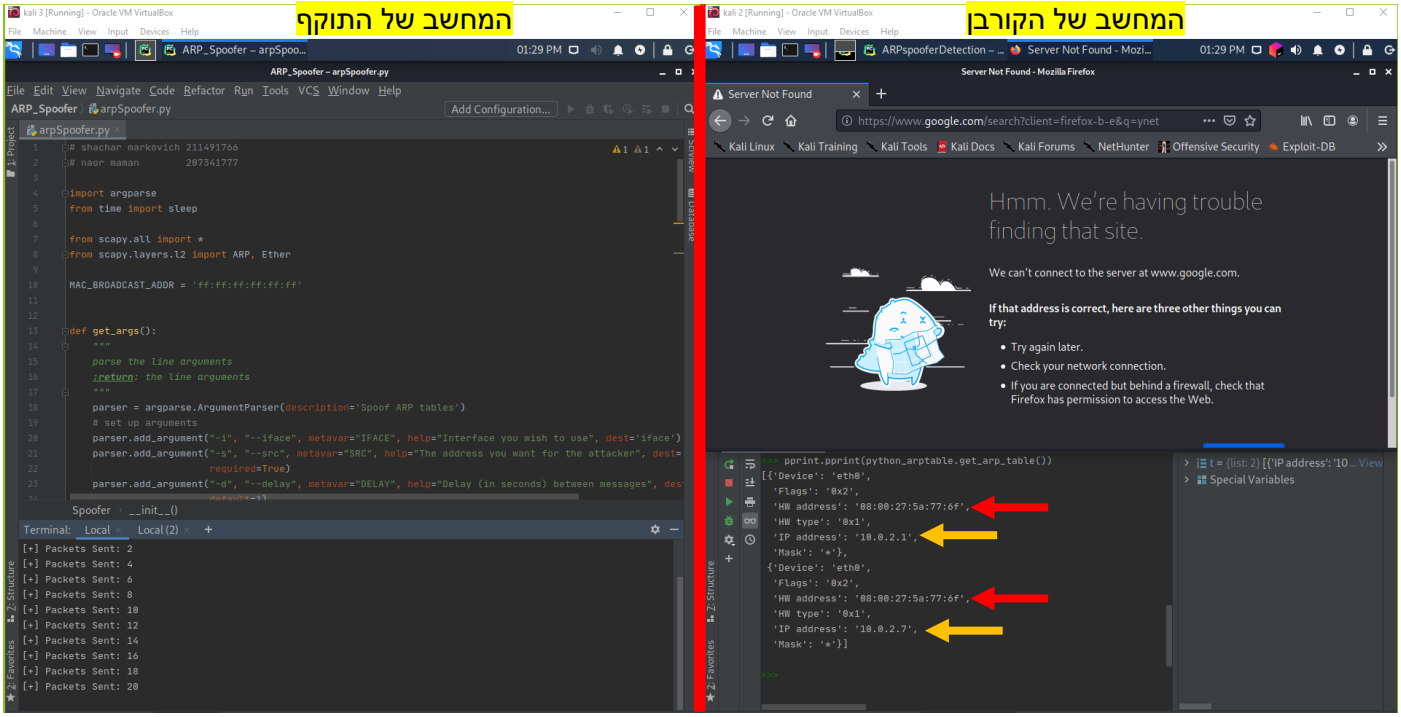
בלולאה אין סופית, כל חמש שניות הפעלנו את הפונק' check_arp_table[6] ששולחת ICMP echo-request לכל כתובות ה-IP שמופיעות בטבלת ה-ARP של הקורבן. אם עבור אחת מהן לא הגיע תשובה בתוך 5 שניות – זה אומר, לדוגמא, שהמחשב שכתובת ה-IP שלו היא X כתובת ה-mac שלו היא **לא** Y.

---

[1] לא מימשנו את האפשרות הזו כי הדרישה הייתה לזהות באמצעות שני כלים ובחרנו את השניים הראשונים.
[2] שורות 129-146
[3] נעשה בפונק' detection_option1, שורות 73-79
[4] נעשה באמצעות הפונק' is_arp_replay, שורות 61-70
[5] שורות 38-58
[6] שורות 86-99

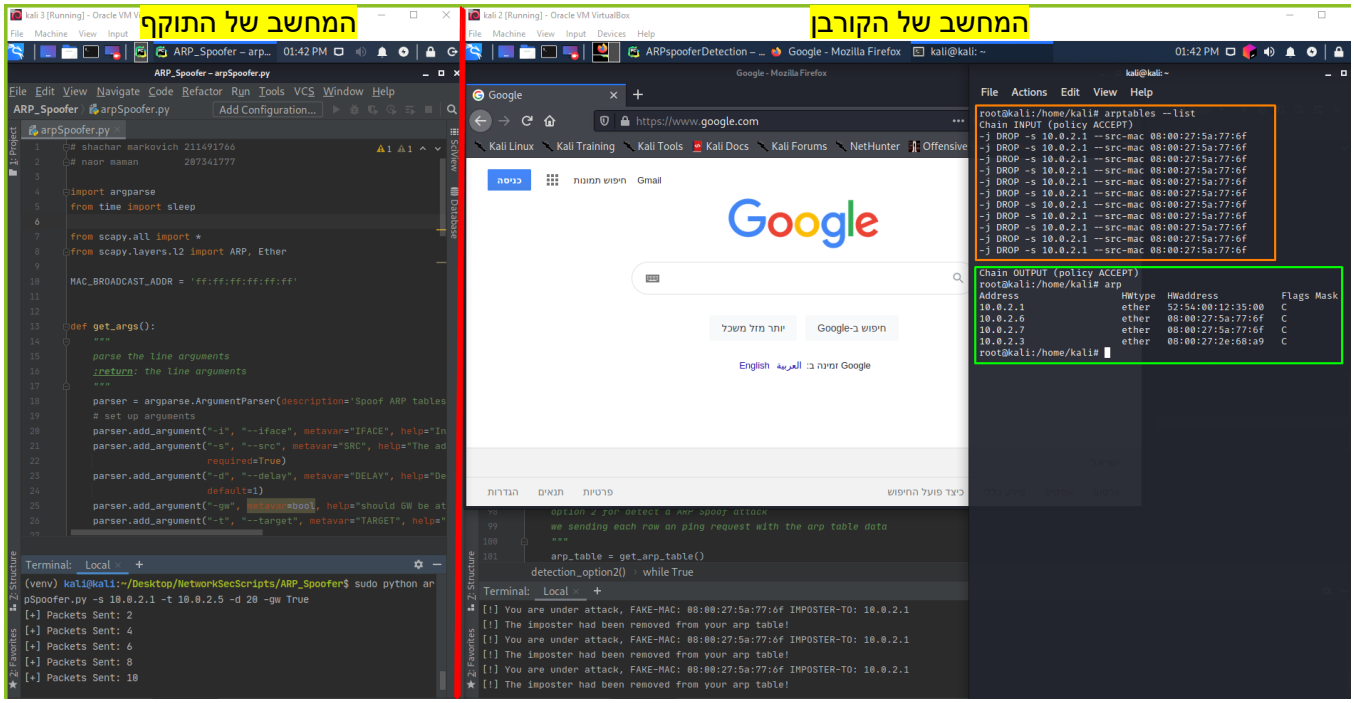
צילומי מסך

ראשית – הוכחה שהמתקפה עובדת ולקורבן אין אינטרנט ואכן בטבלת ה-ARP של הקורבן ישנם שני כתובות IP שלהן משויכת אותה כתובת mac:



מן הצד הימיני של הקו האדום ניתן לראות את המכונה הוירטואלית המותקפת: בחלק העליון של המסך - שאין לה אינטרנט, והבחלק התחתון – שישנם שני כתובות IP בטבלת ה-ARP של מכונה המותקפת שמשויכת להן אותה כתובת mac.

שנית, ברגע שהקורבן הפעיל את הקוד שאמור להזהיר ולמנוע את ההתקפה – הוא אכן עושה זאת!



מצד ימין של הקו האדום ניתן לראות את המכונה הוירטואלית המותקפת ולראות שלמרות שהתוקף מפעיל כרגע את המתקפה – עדיין יש לה אינטרנט והיא גולשת חופשי בגוגל. כמו כן ניתן לראות בחלק התחתון של המסך שהיא מודיע למשתמש שהוא מותקף, וניתן לראות את המדיניות החדשה (מוקף בכתום) שמונעת מן המחשב התוקף להשפיע על טבלת ה-ARP של הקורבן, ומתחתיה (מוקף בירוק) את טבלת ה-ARP של הקורבן ללא שיוך בין כתובת ה-mac של התוקף לכתובת ה-IP אליה הוא רוצה להתחזות.