רשתות מחשבים

פרק 5- Scapy

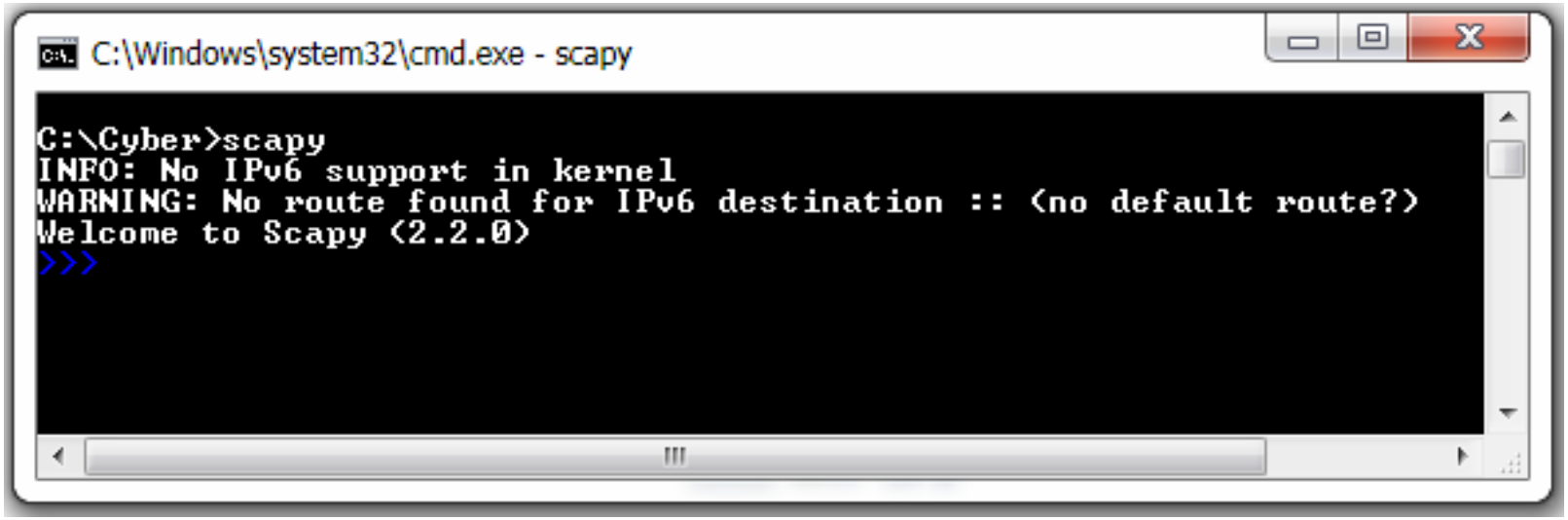
ברק גונן
מבוסס על ספר הלימוד "רשתות מחשבים" מאת
עומר רוזנבוים

מבוא לעבודה עם Scapy

- בשיעורים הקודמים למדנו כיצד אפשר לפתח אפליקציות באמצעות המודול socket בפייתון
- המודול socket אינו עונה על הצורך כשמדובר בשכבות יותר נמוכות
- Scapy- ספריה חיצונית לפייתון שמאפשרת מגוון פעולות:
  - יצירה של פקטות
  - שליחת פקטות
  - הסנפה
  - פילטור פקטות לפי שדות שונים

הפעלה

- בתוך cmd, כיתבו scapy
  - התעלמו מהאזהרה



הפונקציה sniff

- מאפשרת הסנפה של פקטות

```
>>> packets = sniff(count=2)
>>> packets
<Sniffed: TCP:1 UDP:1 ICMP:0 Other:0>
>>> packets.summary()
Ether / IPv6 / UDP fe80::d0d7:e117:159e:a608:59302 > ff02::c:ssdp / Raw
Ether / IP / TCP 10.0.0.2:61649 > 173.194.65.188:5228 A / Raw
>>>
```

- הפקטות נשמרות ברשימה
- ניתן לגשת לכל פקטה ופקטה כמו לאיברי רשימה בפייתון

```
>>> packets[1]
<Ether  dst=c4:12:f5:f8:ab:3e src=38:60:77:25:8e:19 type=0x800 |<IP  version=4L
ihl=5L tos=0x0 len=41 id=15749 flags=DF frag=0L ttl=128 proto=tcp chksum=0x0 src
=10.0.0.2 dst=173.194.65.188 options=[] |<TCP  sport=61649 dport=5228 seq=875105
099 ack=2985448462L dataofs=5L reserved=0L flags=A window=16412 chksum=0xf99b ur
gptr=0 options=[] |<Raw  load='\x00' |>>>>
>>>
```

פילטור פקטות מסוג DNS (תרגיל 5.1)

- כשביצענו הסנפות עם wireshark ראינו שכאשר אנחנו מבצעים nslookup, בשכבת האפליקציה עובר פרוטוקול DNS
  - Scapy מזהה את השדות שמופיעים בפקטות
- אנו יכולים לפלטר לפי שדה כלשהו, ע"י הפרמטר lfilter
- לדוגמה, פילטור פקטות DNS:

```
>>> def filter_dns(packet):
...     return DNS in packet
>>> packets = sniff(count=4, lfilter=filter_dns)
>>> packets.summary()
Ether / IP / UDP / DNS Qry "138.0.0.10.in-addr.arpa."
Ether / IP / UDP / DNS Ans "Broadcom.Home."
Ether / IP / UDP / DNS Qry "www.google.com.Home."
Ether / IP / UDP / DNS Ans
```

המתודה show()

```
>>> my_packet = packets[3]
>>> my_packet.show()
###[ Ethernet ]###
  dst= 38:60:77:25:8e:19
  src= c4:12:f5:f8:ab:3e
  type= 0x800
###[ IP ]###
     version= 4L
     ihl= 5L
     tos= 0x0
     len= 140
     id= 0
     flags= DF
     frag= 0L
     ttl= 64
     proto= udp
     chksum= 0x25d6
     src= 10.0.0.138
     dst= 10.0.0.2
     \options\
###[ UDP ]###
        sport= domain
        dport= 52940
        len= 120
        chksum= 0x324b
###[ DNS ]###
           id= 2
           qr= 1L
           opcode= QUERY
```

- לאחר שבחרנו פקטה מסויימת, ניתן להציג את התוכן שלה על ידי המתודה show()

המתודה show() - המשך

```
>>> my_packet[DNS].show()
###[ DNS ]###
  id= 4
  qr= 0L
  opcode= QUERY
  aa= 0L
  tc= 0L
  rd= 1L
  ra= 0L
  z= 0L
  rcode= ok
  qdcount= 1
  ancount= 0
  nscount= 0
  arcount= 0
  \qd\
   |###[ DNS Question Record ]###
   |  qname= 'www.google.com.'
   |  qtype= A
   |  qclass= IN
  an= None
  ns= None
  ar= None
>>>
```

- אם נרצה להתמקד בפרוטוקול ה-DNS, נבצע show רק לו
- ניתן לבדוק ערך של כל שדה. לדוגמה:

```
>>> my_packet[DNS].opcode
0L
```

התו L מציין שהמספר הוא integer בגודל Long

- מצאנו שכאשר שדה זה הוא 0, הכוונה ל-QUERY (שאילתא)

פילטור פקטות לפי שדות נוספים

```
>>> my_packet[DNS].show()
###[ DNS ]###
  id= 4
  qr= 0L
  opcode= QUERY
  aa= 0L
  tc= 0L
  rd= 1L
  ra= 0L
  z= 0L
  rcode= ok
  qdcount= 1
  ancount= 0
  nscount= 0
  arcount= 0
  \qd\
   |###[ DNS Question Record ]###
   |  qname= 'www.google.com.'
   |  qtype= A
   |  qclass= IN
  an= None
  ns= None
  ar= None
>>>
```

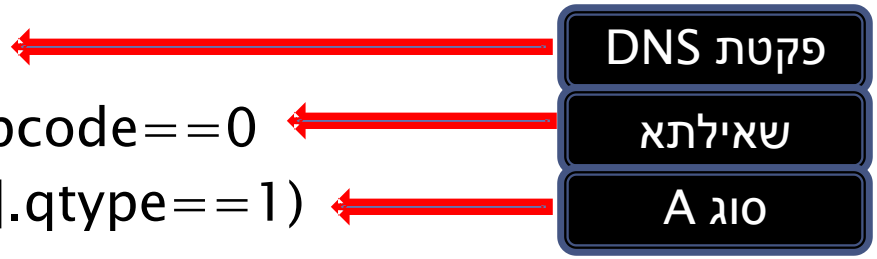
- נניח שאנחנו מעוניינים לפלטר פקטות שעונות לתנאים הבאים:
  - DNS- למדנו
  - שאילתא - למדנו
  - טיפוס A (לא PTR):

```
>>> my_packet[DNSQR].qtype
1
```

פילטור פקטות - המשך

- נשפר את הפילטר שלנו:

```
>>> def filter_dns(packet):
        return (DNS in packet
            and packet[DNS].opcode==0
            and packet[DNSQR].qtype==1)
```



הרצת פונקציה על פקטות לאחר פילטר

- כעת כשלמדנו ליצור פילטרים מרובי תנאים, נרצה לבצע פעולות על הפקטות שפילטרנו
- ניצור פונקציה שמדפיסה את ה-domain עליו מתבצע תשאול ה-DNS

```
>>> def print_query_name(dns_packet):
        print dns_packet[DNSQR].qname
```

- איך אפשר לגרום לכך שרק פקטות שנתפסו בפילטר שלנו יגיעו לפונקציה print_query_name?

הפרמטר PRN

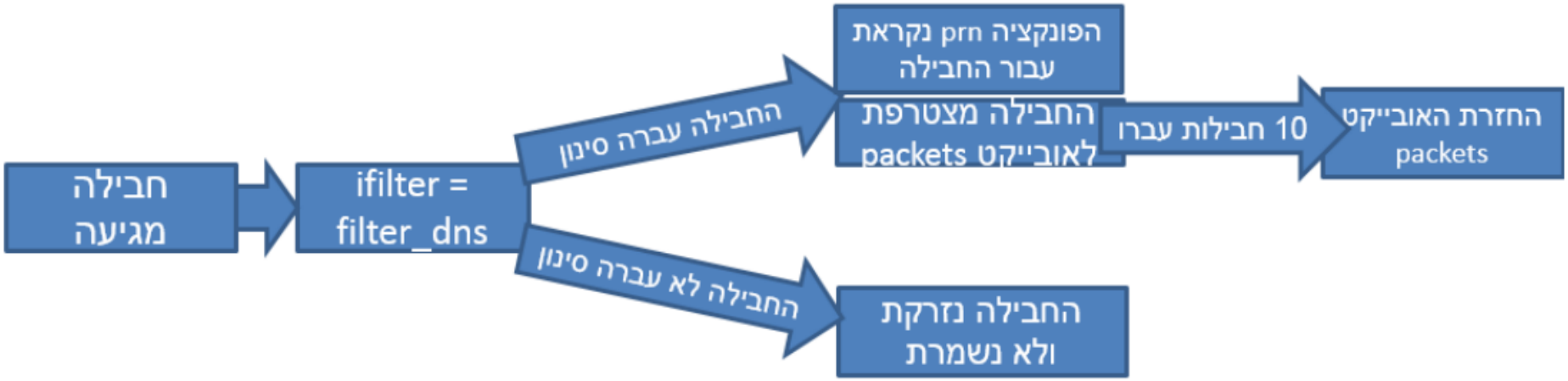
- נעביר את הפונקציה ל-sniff ע"י הפרמטר prn:

```
>>> sniff(count=10, lfilter=filter_dns, prn=print_query_name)
www.google.co.il.
www.google.co.il.
www.google.com.
www.google.com.
apis.google.com.
lh4.googleusercontent.com.
plus.google.com.
apis.google.com.
lh4.googleusercontent.com.
plus.google.com.
<Sniffed: TCP:0 UDP:10 ICMP:0 Other:0>
>>>
```

- הערה: במידה ולא התקבלו תוצאות, יתכן שזה עקב מנגנון ה-cache, יש לאפס אותו ע"י הפקודה:

ipconfig/flushdns

סיכום- שרשרת הפעולות המלאה



גבהים
תכנית הסייבר הלאומית

שימוש ב-scapy מתוך סקריפט פייתון

- כתיבת תוכניות פייתון שעושות שימוש ב-scapy יסייעו בדיבוג ובשמירת הקוד להמשך
- הקוד הבא מייבא את scapy ופותר באג בהדפסה

```
import sys
i, o, e = sys.stdin, sys.stdout, sys.stderr
from scapy.all import *
sys.stdin, sys.stdout, sys.stderr = i, o, e
```

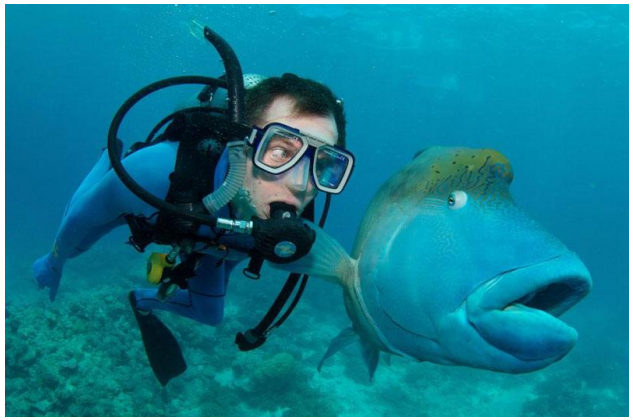
תרגיל- פילטור HTTP

- בצעו את תרגיל מודרך 5.2, הסנפה של HTTP

גבהים
תכנית הסייבר הלאומית

סיכום ביניים

- למדנו איך משתמשים ב-scapy להסנפה של פקטות
- ראינו שיש ביכולתנו ליצור פילטרים מסוגים שונים
  - ב-wireshark אנחנו מוגבלים על ידי פילטרים שמישהו אחר כתב
- למדנו לגרום להרצה של פונקציה על פקטות שעברו סינון
- ... מה לגבי יצירה של פקטות?

שימוש ב-Scapy ליצירת פקטות

- הזכרנו ש-scapy מאפשר יצירה של פקטות
- פעולה זו דורשת היכרות עם השכבות שמתחת לשכבת האפליקציה
- לכן נלמד אותן, ואז נחזור ל-scapy וניישם את מה שלמדנו ☺
- כעת אנחנו מוכנים לצלול לעומק השכבות הבאות...