"הגנת סייבר במערכות מבוססות רשת" – תרגיל 1

תאריך הגשה: 06.12.2015

הגשה ביחידים או בזוגות בלבד

את התרגילים כולם יש לבצע במכונה וירטואלית ללא גישה לאינטרנט אלא במצב Internal Network בלבד

כלל שורות הקוד הנכתבות הפייתון צריכות לעמוד בתקן PEP8[1]

שאלה 1

טכניקה ידועה לזיהוי מערכת ההפעלה של רכיבים שונים ברשת בעזרת ניתוח תעבורת הרשת אותה הם שולחים נקראת "Passive Fingerprinting[2]". יש לכתוב תוכנית ב-Python אשר עוברת הודעה הודעה (Packets/Frames/Segments/Dagagrams בהתאם לשכבות השונות) ומנתחת אותה. הפרמטרים אשר בעזרתם תקבע מערכת ההפעלה הינם:

- TTL
- Window Size
- TCP Options

מערכת הפעלה	ערך התחלתי של TTL	ערך התחלתי של Window Size	TCP Options
Linux	64	-	מכיל מידע אודות ה-timestamp בתחילת תקשורת TCP
Windows	128	8192	-

- HTTP UserAgent (http://www.useragentstring.com/)
  - ניתן לתמוך רק בדפדפנים הבאים:
    - Chrome
    - Firefox

התוכנית צריכה לתמוך הינה בניתוח קבצי pcap והן בביצוע הסנפת תקשורת באופן עצמאי.

הערה: ישנם מאפיינים נוספים אשר בעזרתם ניתן לבצע Passive Fingerprinting אולם במסגרת התרגיל יש לממש רק את אלו המוזכרים

הערה: מומלץ להשתמש ב-Scapy לשם כתיבת התוכנית

---

[1] https://www.python.org/dev/peps/pep-0008/

[2] http://forensicswiki.org/wiki/OS_fingerprinting

Usage:

- עבור הרצת התוכנית על הקובץ לדוגמא **q1a.pcap**:

```
q1.py -f /tmp/q1a.pcap
```

- עבור הרצת התוכנית על תעבורת תקשורת חיה (דהיינו ביצוע הסנפת תקשורת):

```
q1.py -s
```

דוגמאות פלט:

- עבור הרצת התוכנית על הקובץ **q1a.pcap** יתקבל הפלט אשר יכלול לפחות את השורות הבאות:

```
IP (TTL), 10.0.2.15, Linux

TCP (Options), 10.0.2.15, Linux

HTTP (User-Agent), 10.0.2.15, Ubuntu Linux x86_64, Chromium/45.0.2454.101

HTTP (User-Agent), 10.0.2.15, Ubuntu Linux x86_64, Firefox/41.0
```

- עבור הרצת התוכנית על הקובץ **q1b.pcap** יתקבל הפלט אשר יכלול לפחות את השורות הבאות:

```
IP (TTL), 192.168.1.106, Windows

TCP (Window Size), 192.168.1.106, Windows

HTTP (User-Agent), 192.168.1.106, Windows 10 x64, Firefox/42.0

HTTP (User-Agent), 192.168.1.106, Windows 10 x64, Chrome/46.0.2490.80
```

שאלה 2

יש לכתוב תוכנית ב-Python אשר שולחת הודעות GET ב- HTTP עבור (יש להשתמש ב-Sockets בלבד ללא RAW Sockets/Scapy) אשר מתחזה לטלפון סלולרי מבוסס Android המריץ דפדפן Opera. ניתן להשתמש ב-User-Agent הבא:

Opera/12.02 (Android 4.1; Linux; Opera Mobi/ADR-1111101157; U; en-US) Presto/2.9.201 Version/12.02

על ההודעה להיות זהה להודעה אשר יוצאת מהדפדפן שלכם רק בשינוי ה-User-Agent.

(על מנת לבדוק זאת ניתן להשתמש בשרת Web מבוסס python באמצעות הרצת הפקודות הבאות python -m SimpleHTTPServer, יש לשים לב שכברירת מחדל השרת מאזין לפורט 8000).

Usage:

הפרמטרים הנדרשים הם כתובת ה-IP והפורט של שרת ה-Web אליו נשלחת בקשת ה-GET אשר שלחתם

```
q2.py 127.0.0.1 8000
```

יש לצרף הסנפת תקשורת של המידע ששלחתם לשרתם ולוודא כי אתה מקבלים תשובה תיקנת משרת ה-Web (Status Code 200 ולא שגיאה כלשהי).

שאלה 3

יש להסביר במילים כיצד ניתן לזהות את תהליך ההתחזות משאלה 2. ניתן להיעזר בתשובתכם לשאלה 1.

שאלה 4

יש לכתוב תוכנית ב-Python אשר מקבלת כפלט נתיב לקובץ pcap ומחזירה רשימה של כתובת ה-IP של כלל הרכיבים אשר זוהו כ-Default Gateways. יש לתעד את באופן ברור את האלגוריתם לשם זיהוי רכיבי ה-Default Gateway.

Usage:

- עבור הרצת התוכנית על הקובץ לדוגמא **q4.pcap**:

```
q4.py -f /tmp/q4.pcap
```

דוגמאות פלט:

עבור הרצת התוכנית על הקובץ **q4.pcap** יתקבל הפלט הבא:

```
Default Gateway was found on 192.168.1.1 (00:27:19:14:07:24)
```