"הגנת סייבר במערכות מבוססות רשת" – תרגיל 2

תאריך הגשה: 17.01.2016

הגשה ביחידים או בזוגות בלבד

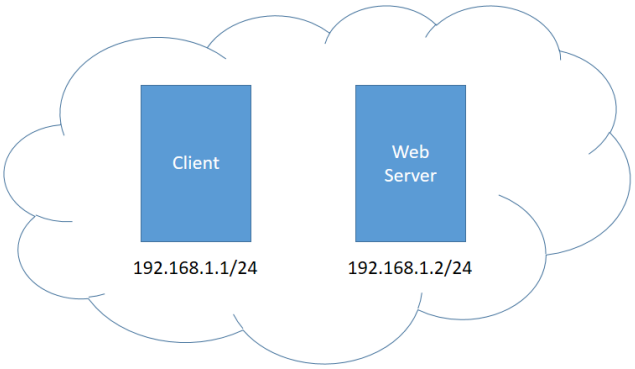
את התרגילים כולם יש לבצע במכונה וירטואלית ללא גישה לאינטרנט אלא במצב Internal Network בלבד

כלל שורות הקוד הנכתבות ב-Python צריכות לעמוד בתקן PEP8[1]

מומלץ לבצע את התרגיל תוך שימוש בכלי Source Control כלשהו (למשל git או bitbucket)

שאלה 1

נתונה הרשת הבאה:



ה-Web Server מקבל הודעת HTTP ומדפיס למסך הודעה שנשלחה וכתובת ה-IP שממנה הגיע.

יש לכתוב תוכנית ב-Python אשר שולחת הודעה HTTP תקינה מה-Client לשרת ה-Web אולם שולחת את ההודעה תחת כתובת IP שאינה הכתובת שלה (דהיינו ביצוע IP Spoofing), למשל 192.168.1.17.

הערה: ה-Web Server מצורף לתרגיל זה (http_server.py)

הערה: מומלץ להשתמש ב-Scapy לשם כתיבת התוכנית

Usage:

- עבור הרצת התוכנית הבאה:

```
q1.py -src ="192.168.1.17" -dst="192.168.1.2" -msg="AAA"
```

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

דוגמאות פלט (ב-Standard Output של שרת ה-Web, השורה הבאה צריכה להיות מוכלת בפלט):

```
Connected From: 192.168.1.17
```

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

שאלה 2

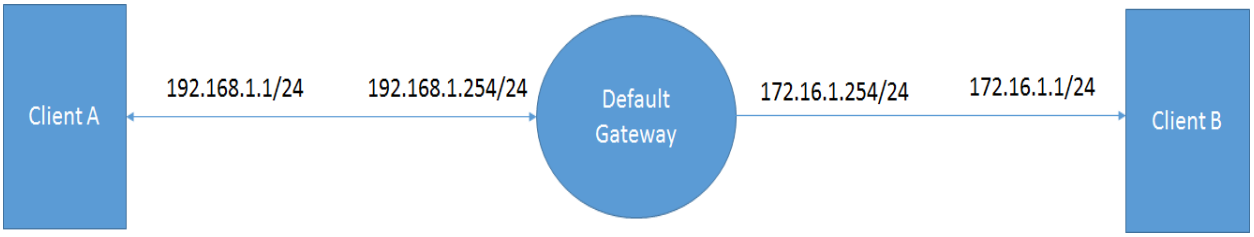
יש לשנות את הקוד של שרת ה-Web כך שבמידה ומבוצע IP Spoofing (כלומר לוודא את אמינות כתובת ה-IP) יזהה זאת השרת ויחזיר הודעת שגיאה מתאימה.

יש לצרף את הקוד של שרת ה-Web לאחר השינוי.

יש לצרף הסנפת תקשורת של שני המצבים אחד בהם מבוצע Spoofing (ניתן להשתמש בקוד משאלה 1) והשני כאשר המידע נשלח מתחנה לגיטימית (כאשר לא מבוצע Spoofing).

שאלה 3

על מנת לבצע את תרגיל זה נדרש להקים את הסביבה הבאה:



Client A: מחשב וירטואלי אשר מריץ מערכת הפעלה לינוקס (למשל Ubuntu). יש לבצע הגדרה סטטית של כתובת הרשת וה-DGW בהתאם למופיע בתרשים לעיל.

Default Gateway: מחשב וירטואלי אשר מריץ מערכת הפעלה לינוקס (למשל Ubuntu). יש לדאוג כי למחשב זה מחוברים שני כרטיסי רשת שכתובות ה-IP שלהן מוגדרות כמופיע בתרשים לעיל. על מנת שהמערכת ההפעלה תנתב הודעות בין כרטיסי הרשת השונים ניתן להיעזר בלינק הבא: http://www.ducea.com/2006/08/01/how-to-enable-ip-forwarding-in-linux/

Client B: מחשב וירטואלי אשר מריץ מערכת הפעלה לינוקס (למשל Ubuntu). יש לבצע הגדרה סטטית של כתובת הרשת וה-DGW בהתאם למופיע בתרשים לעיל.

יש לכתוב תוכנית ב-Python אשר כוללת את היכולות הבאות:

- **HTTP**[2]:
  - חסימת הורדה של קבצים אשר מכילים בשם הקובץ אחת מהסיומות הבאות:
    - xls
    - xlsx
    - pdf
    - doc
    - docx
    - exe
    - bat

כלל ההגדרות צריכות להופיע בקובץ הגדרות נפרד ולא בקוד עצמו, כמו כן יש להוסיף אפשרות בה במידה של חסימת ההורדה תוצג הודעה למשתמש אודות חסימת המידע או חסימה שקטה ללא הצגת הודעה למשתמש

אין צורך לתמוך ב-Chunked HTTP
יש להתמודד עם אפשרות של חלוקת ההודעה למספר Fragments (ניתן להניח כי לא יהיו יותר מ-4)

- **SSH**:
  - על מנת למנוע מגורמים לא מורשים להתחבר ב-SSH לרכיבים ברשת 172.16.1.0 יש לממש ב-DGW מנגנון דמוי "TCP Port Knocking"[3] אשר מתמודד עם התקפות Spoofing ו-Replay (לכן פתרון לדוגמא של שליחת הודעות אלו על ידי קיבוע ה-TTL לערך קבוע אינו פתרון מספק). דהיינו יש להוסיף מנגנון טרם קבלת ה-Prompt של SSH. לשם התקנת שרת SSH ב- Client B יש לעקוב אחר ההוראות הבאות: https://help.ubuntu.com/lts/serverguide/openssh-server.html. יש לצרף קבצי pcap מתאימים (אחד לפחות של המנגנון מאשר חיבור לפורט 22 ואחד לפחות שלא).

יש לתאר את האלגוריתם שהוצע תוך הסבר למה הוא מטפל ב-Spoofing ו- Replay Attacks. את התיאור אפשר לצרף כקובץ נפרד או באמצעות docstrings בקוד.

יש להשתמש ב-NetfilterQueue על מנת להעביר את הודעות התקשורת מה-Kernel Mode ל-User Mode.
לשם התקנה של מודל זה ניתן להיעזר בלינק הבא: https://pypi.python.org/pypi/NetfilterQueue/0.3.
כמו כן, ניתן להיעזר בדוגמאות המופיעות בלינק הנ"ל ו/או בקובץ הדוגמא[4] המצורף (nfqueue_example.py)

---

[2] ניתן להיעזר ב-SimpleHTTPServer כפי שנעשה בתרגיל הראשון.

[3] https://en.wikipedia.org/wiki/Port_knocking

[4] חשוב לציין כי הקוד בקובץ הדוגמא אינו עומד בסטנדרטי הפיתוח המצופים בקורס אלא מהווה דוגמא כללית בלבד.