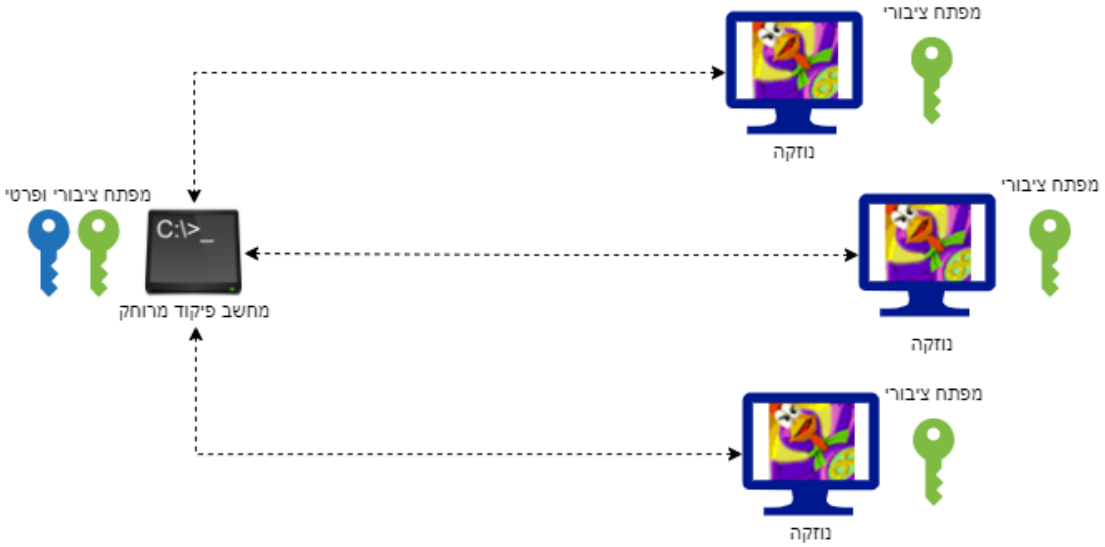
טכנו"צ סייבר – תרגיל 3 – "B4ckd00r"

הרצת פיקודים מרחוק

מטרת העל

בתרגילים הקודמים בנינו את התשתית לנוזקה שלנו כך שהיא תוכל להיות מוצפנת ומותממת. כעת, הגיע הזמן להשמיש אותה ולדאוג שאכן נוכל לשלוט בה מרחוק. נרצה להיות מסוגלים להריץ פקודות זדוניות במחשב הפגוע. בתרגיל זה נתמקד ביכולת להריץ קוד גנרי – **ללא ההצפנה וללא פעולות מוגדרות**.



***את ההצפנה בתקשורת נממש בתרגיל הבא**

השיטה

במשורטט באיור לעיל, הנוזקה מופצת במחשבים מסוימים ומצפה לקבל פיקודים מרוחקים ממחשב שולט כלשהו. עם זאת, נרצה להגן על עצמנו ולדאוג שהתהליך יהיה חשאי וללא יכולת להתערבות עוינת, כלומר לא נרצה שאדם זדוני אחר יוכל לשלוח פיקודים בעצמו. כיצד נעשה זאת? נשתמש בשיטה שנקראת Port Knocking.

מה הוא Port Knocking?

ישנן דרכים רבות לבצע את השיטה הזאת – אנחנו נבחר בדרך אחת מהן. במחשב הנתקף נפתחים מספרים פורטים מוסכמים מראש. במקרה שלנו נניח ש-3 כאלה (וספציפית 1234,5678,1278) . שלושת הפורטים האלה לא משמשים את הנוזקה ישירות, כלומר היא לא תקבל פיקודים מהפורטים האלה ותתעלם מתוכן כל ההודעות הנשלחות אליהן. עם זאת, הנוזקה תאזין ברשת ותחכה לקבל רצף של פקטות בסדר מסויים ב-3 הפורטים הללו. ברגע שרצף נכון של פקטות מגיע, הנוזקה תפתח פורט נוסף – פורט השליטה עליה (לבחירתכם איזה פורט זה). בפורט זה יוכל התוקף לשלוח פקודה כלשהי ואותה הנוזקה תריץ. לאחר שליחת הפקודה פורט השליטה יסגר וכדי שהתוקף יוכל לשלוח פקודה נוספת הוא יצטרך לחזור על התהליך שתארנו. שימו לב, כל

טכנו"צ סייבר – תרגיל 3 – "B4ckd00r"

החלק של ה-Port Knocking ***קורה ב-UDP.***
בתרגיל זה נרצה שהרצף יהיה 1234, לאחר מכן 5678 ולבסוף 1278.

אז איך התהליך נראה?

1) הנוזקה מורצת במחשב, זאת בעזרת ההתממה שכתבתם בתרגיל 1.
2) יפתחו שלושת הפורטים הרלוונטים ל-Port Knocking.
3) הנוזקה תאזין ברשת ותצפה לרצף מסוים של פקטות.
4) אם רצף כזה מגיע, הנוזקה תפתח פורט נוסף ותצפה לקבל בו פיקוד.
5) הנוזקה תריץ את הפיקוד על המחשב
6) הנוזקה תחזיר את הפלט חזרה למחשב ששלח את הפיקוד
7) הנוזקה תסגור את פורט השליטה ותחזור לשלב 3

<u>שימו לב</u>: בתרגיל זה נממש רק את שלב 2 והלאה, כלומר אין צורך לחבר לנוזקה שכתבנו בתרגיל 1.

כעת מהו הפיקוד עליו אנו מדברים?
במקרה שלנו נרצה להיות מסוגלים להריץ כל קוד פייתון כרצוננו. בתרגיל מס' 4 נכתוב מספר פיקודים מעניינים מאוד – אבל כעת רק נרצה שתממשו פיקוד אחד שמדגים את הקונספט. לכן, נרצה שתכתבו פיקוד אשר מריץ את הפקודה:

```
print('Hello World')
```

על המחשב עם הנוזקה ומחזיר את הפלט (Hello World במקרה הזה) הביתה.

נשמע פשוט?
העניינים מסתבכים, הפונקציה בה נרצה שתשתמשו היא exec() – פונקציה זו אכן מקבלת קוד פייתון כפרמטר ומריצה אותו. עם זאת, היא **אינה** מחזירה את הפלט, לכן נצטרך למצוא דרך לקבל ממנה את הפלט (כדי להחזיר אותו). כיצד נעשה זאת?
נעטוף את הקוד שלנו שמריץ את ה exec בקטע קוד אשר יוצר קובץ על המכונה הנגועה וכותב בו את הפלט של קטע הקוד שלנו. כעת, יכולה המכונה הנגועה לפתוח את הקובץ ולשלוח אותו חזרה למחשב הבית. לבסוף תמחק המכונה הנגועה את הקובץ כדי לטשטש ראיות.

<u>שימו לב:</u> שלבים 4 עד 7 יכולים לקרות ב TCP

חומרי עזר

יסופקו לכם קבצי שלד לתרגיל, עליכם לממש את הפונקציות בהן. תוכלו לכתוב פונקציות עזר נוספות משלכם אם תרצו. בנוסף אנחנו ממליצים לקרוא באינטרנט על socket-ים בפייטון וכיצד להשתמש בהם.

טכנו"צ סייבר - תרגיל 3 - "B4ckd00r"

קבצי השלד

ישנן 4 פונקציות משמעותיות אותן תצטרכו לממש (בנוסף לפונקציות עזר כרצונכם במידת הצורך):

עבור הנוזקה:
בקובץ ה-main, פונקצית **handle_recive** אשר מקבלת פקודה בקלט ומריצה אותה במחשב הנגוע כמו שהסברנו בפסקת ההסבר על הפיקוד.
בקובץ ה-listener, פונקציית **listen_for_commands** אשר מקבלת כקלט את הפונקציה שכתבתם בקובץ ה-main. פונקציה זאת תטפל בכל תהליך האזנה כמו שפרטנו לעיל (כולל ה-Port Knocking עליו הסברנו קודם). ברגע שתגיע הנקישה הנכונה, הפונקציה תפתח את פורט השליטה (שנתון לבחירתכם) ותהיה מוכנה לקבלת הפקודה ממחשב הבית. לאחר מכן היא תריץ את הפקודה בעזרת הפונקציה שקיבלתם כקלט ותחזיר את הפלט חזרה למחשב הבית.

עבור מחשב הבית:
בקובץ ה-main, פונקציית ה-**main** שמטפלת בממשק מחשב הבית, במקרה שלנו כעת היא פשוט תשלח לפונקציה עליה נסביר מיד את הפיקוד שכתבתם (כלומר את קטע הקוד הזדוני).
בקובץ ה-sender, פונקצית **send_payload** אשר מטפלת בכל תהליך התקשורת כמו שפרטנו לעיל (רק כעת מצד הבית) ולאחר שיפתח הפורט הרלוונטי תשלח את הפקודה הזדונית שקיבלה כקלט. לבסוף נרצה שתדפיס את התשובה אשר מגיעה מהמחשב הנגוע.

עבור שני המחשבים:
קיים קובץ settings.py, השתמשו בו עבור הפורטים והכתובת IP של מחשב היעד. במידה ועשיתם לו שינויים משמעותיים (יותר מלקבוע פורט פיקוד), הוסיפו ל README.

בונוס מעשי

ברגע הנוזקה שלנו יוצרת קובץ כל פעם שאנחנו מריצים פיקוד, דבר זה יחסית חשוד. קראו על StringIO והחזירו את הפלט בעזרתו.

בונוס תיאורטי

התכונה שכתבנו לא תעבוד כנראה על מחשבים רגילים באינטרנט בלי הכנות מקדימות.

1. קראו על המנגנונים NAT ועל Firewall והסבירו בקצרה (משפט שתיים) על שניהם.
2. הסבירו כיצד המנגנונים עליהם הסברתם יכולים למנוע מהנוזקה שלנו לעבוד כמו שצריך ולקבל מידע.
3. הציעו דרך להתמודד עם ה-NAT.

את התשובות לשאלות כתבו בקובץ ה README.

טכנו"צ סייבר – תרגיל 3 – "B4ckd00r"

דגשים חשובים לתרגיל

שימו לב שהקוד שלכם **אמור לתמוך** בדברים הבאים:

1. פיקודים בכל אורך שנרצה
2. החזרת פלט עבור כל אורך פלט שנרצה
3. החזרת שגיאות במידה וקרתה שגיאה בזמן הריצה

ספרייה רלוונטית לתרגיל

במהלך התרגיל עליכם להשתמש בספרייה PyDivert – בעזרת ספרייה זו תוכלו להאזין לתעבורה ברשת וגם להתערב בה במידת הצורך (שליפת פקטות ומניעת הגעתן אל היעד) – רלוונטי למחשב הנגוע. שימו לב, **ספרייה זו עובדת רק כשרצים כמנהל, אל תפספסו את זה.**

הגשה

הגישו קובץ zip בשם **ex3_[FullName].zip** שעוטף את כל תיקיית התרגיל שלכם (התיקיות commander, malware, global וקבצי main) כולל קובץ README במידת הצורך.

בהצלחה!