טכנו"צ סייבר - תרגיל 3 - "B4ckd00r"

הרצת פיקודים מרחוק

מטרת העל

בתרגילים הקודמים בנינו את התשתית לנוזקה שלנו כך שהיא תוכל להיות מוצפנת ומוטממת. כעת, הגיע הזמן להשמיש אותה ולדאוג שאכן נוכל לשלוט בה מרחוק. נרצה להיות מסוגלים להריץ פקודות זדוניות במחשב הפגוע. יתר על כן, נרצה לוודא שאם התקשורת שלנו עם הנוזקה מזוהה בדרך כלשהי היא לא מתגלית כנוזקתית, ובוודאי שלא מפוענחת ומובנת ע"י חוקרי ההגנה.

התרגיל יתחלק לשני חלקים גדולים:

1. יצירת תשתית תקשורת מוצפנת עם הנוזקה
2. כתיבת מנגנון הרצת הפיקודים

שלב 1 – תשתית תקשורת

השיטה

כמשורטט 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.

טכנו"צ סייבר - תרגיל 3 - "B4ckd00r"

מה הוא *Port Knocking*?

ישנן דרכים רבות לבצע את השיטה הזאת - אנחנו נבחר בדרך אחת מהן. במחשב הנתקף הנוזקה מאזינה למספר 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 תתחיל להאזין לפורט נוסף - פורט השליטה עליה (לבחירתכם איזה פורט זה, אבל גם הוא יצטרך להיות קבוע מראש). בפורט זה יוכל התוקף לשלוח פקודה כלשהי ואותה הנוזקה תריץ. לאחר שליחת הפקודה פורט השליטה יסגר (=הנוזקה תפסיק להאזין עליו) וכדי שהתוקף יוכל לשלוח פקודה נוספת הוא יצטרך לחזור על התהליך שתארנו. שימו לב, כל 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) תתחיל האזנה לשלושת הפורטים הרלוונטים ל- Port Knocking.
3) הנוזקה תאזין ברשת ותצפה לרצף מסוים של פקטות.
4) אם רצף כזה מגיע, הנוזקה תתחיל להאזין לפורט נוסף ותצפה לקבל בו פיקוד מוצפן.
5) הנוזקה תפענח את הפיקוד, ואז תריץ את הפיקוד על המחשב.
6) הנוזקה תצפין את הפלט ותחזיר אותו חזרה למחשב ששלח את הפיקוד.
7) הנוזקה תפסיק להאזין לפורט השליטה ותחזור לשלב 3.

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

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

<u>שימו לב:</u> חשבו על הפקטות שמגיעות כחלק מתהליך ה Port Knocking (ולמעשה באופן כללי על התקשורת של הנוזקה שלנו) – האם הן מיועדות לתוכנה כלשהי על מחשב היעד מלבד הנוזקה? מה יקרה אם תוכנה אחרת על מחשב היעד תוכל לקלוט את הפקטות האלה? האם ניתן למנוע זאת?

כעת מהו הפיקוד עליו אנו מדברים?

שלב 2 – כתיבת מנגנון הרצת הפיקודים

במקרה שלנו נרצה להיות מסוגלים להריץ כל קוד פייתון כרצוננו. בתרגילים הבאים נכתוב מספר פיקודים מעניינים מאוד - אבל כעת רק נרצה שתממשו פיקוד אחד שמדגים את הקונספט. לכן, נרצה שתכתבו פיקוד אשר מריץ קוד פייתון, למשל:

טכנו"צ סייבר - תרגיל 3 - "B4ckd00r"

print('Hello World')

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

שימו לב שקוד הפייתון שרץ ניתן לנוזקה בקלט – כלומר הוא מגיע אל הנוזקה משרת הפיקוד על גבי פורט השליטה, ואינו קבוע בנוזקה. יכולת זו תאפשר לנו גמישות גבוהה מאוד ביכולת לפעול על גבי מחשב היעד, וזו יכולת שננצל בתרגילים הבאים היטב.

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 להעלים ראיות.

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 בפייתון וכיצד להשתמש בהם.

טכנו"צ סייבר - תרגיל 3 - "B4ckd00r"

קבצי השלד

ישנן 3 פונקציות משמעותיות אותן תצטרכו לממש (בנוסף לפונקציות עזר כרצונכם במידת הצורך):

עבור הנוזקה:

בקובץ ה-main, פונקציית **handle_receive** אשר מקבלת פקודה כקלט, מריצה אותה במחשב הנגוע כמו שהסברנו בפסקת ההסבר על הפיקוד ומחזירה כ string את כל מה שהפקודה מדפיסה למסך.

בקובץ ה-listener, פונקציית **listen_for_encrypted_commands** אשר מקבלת כקלט callback לטיפול בפקודות (on_recv). 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 תפענח אותה בעזרת decrypt_and_validate_data שכבר מימשתם, ולאחר מכן היא תריץ את הפקודה בעזרת on_recv שהתקבלה כקלט, תצפין את הפלט באמצעות encrypt_data ותחזיר את הפלט חזרה למחשב הבית.

הערה: כפי שתוכלו לראות בקבצי השלד, on_recv שתתקבל היא למעשה handle_receive שתכתבו בעצמכם!

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 **encrypted_send_payload** אשר מטפלת בכל תהליך התקשורת, ההצפנה והפענוח 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.

גם על המפתח ההצפנה להיות בקובץ ה settings של הפרויקט, תחת השם RSA_KEY (במחשב הפיקוד יהיה המפתח הפרטי, אצל הנוזקות יהיה המפתח הציבורי). בקובץ הsettings שלכם השתמשו בצמד המפתחות:

תוקף (פרטי) - (d, n) = (90687, 57101017)

נוזקה (פומבי) -(e, n) = (10013807, 57101017)

טכנו"צ סייבר - תרגיל 3 - "B4ckd00r"

בונוס מעשי I

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

בונוס מעשי II

זה חלק שהוא באמת לחלוטין סתם לכיף, שלבו את תרגיל 1 ואת תרגיל 4. בשלב זה יש לכם ווירוס שעובד. הרימו VM והדגימו שימוש של הווירוס שלכם על ה VM הזה (צלמו סרטון וצרפו להגשה).

בונוס מעשי III

הנוזקה שלנו מאפשרת למעשה הרצת קוד פייתון שרירותי על המחשב הנגוע, בהרשאות Admin!

מה כבר אפשר לעשות עם זה?

חשבו על פיקודים נוספים שניתן לממש וממשו אותם. שימו לב לשנות בתהליך **רק את הקוד בתיקיית commander** - הנוזקה כבר על מחשב היעד וכל מה שאפשר לעשות זה לנצל את היכולת הקיימת של הרצת קוד פייתון שרירותי.

הוסיפו כל פיקוד שאתם מממשים כפונקציות בקובץ main.py בתיקיית commander, ותארו אותו בקובץ ה README.

רעיונות:

- Shell: יצירת terminal לתוקף שמריץ פקודות על המחשב הנתקף
- Proxy: מאפשר לכם לגשת לאתר אינטרנט באמצעות המחשב הנתקף (כלומר התעבורה תעבור דרכו)
- Persistence: הווירוס שלכם ידלק כל פעם שהמחשב הנתקף ידלק
- Update: "עדכון גרסה" של הקוד שהנוזקה מריצה, למשל ע"מ לשנות את אלגוריתם ההצפנה או למטרה אחרת

בונוס תיאורטי

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

1. קראו על המנגנונים NAT ועל Firewall והסבירו בקצרה (משפט או שניים) 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. הרצת קוד פייתון בכל אורך שנרצה
2. החזרת פלט עבור כל אורך פלט שנרצה
3. החזרת שגיאות במידה וקרתה שגיאה בזמן הריצה
4. עבודה נקייה ככל האפשר – ללא "שריטות" והשארת רמזים לחוקרי הגנה

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, general וקבצי המain) כולל קובץ README במידת הצורך.

בהצלחה!