טכנו"צ סייבר - תרגיל 7 - "WORM"

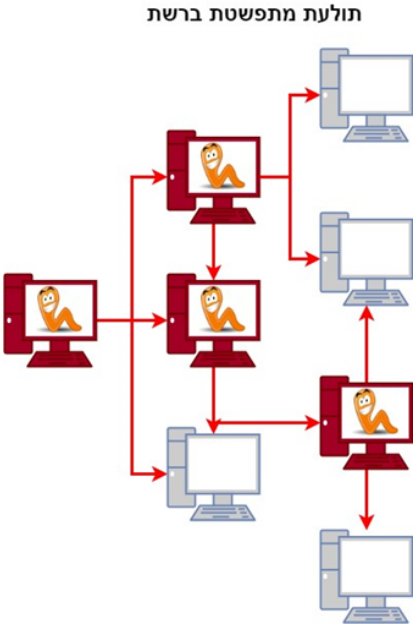
ניצול חולשות ברשת

מטרת העל

בתרגילים הקודמים הוספנו לנוזקה שלנו יכולת להשתמש בחולשות רשת. בתרגיל זה נרצה לנצל חולשה באחת התוכנות ברשת כדי להפיץ את הנוזקה שלנו ברשת ולהגיע למחשב היעד העיקרי שלנו.

רקע

לאחר שגנבו לדופנשמירץ את הפרטים לאימייל הצלחנו להוריד לו את הנוזקה למחשב. להפתעתנו גילינו שדופנשמירץ הוא למעשה רק בורג קטן במערכת, דרג זוטר. ברצוננו לחקור את הרשת ולהגיע למחשב של מנב"ל הרשת.



מתוך מחקר מעמיק על הרשת הצלחנו לגלות על המחשב של דופנשמירץ קליינט אימייל ישן שרץ כל כמה שעות ופונה לשרת בכדי לשלוף מיילים. לאחר חיפוש ברחבי הרשת הצלחנו להבין כיצד עובד קליינט האימייל הזה וללמוד כי השימוש בו היה אמור להיפסק עקב חולשה שהתגלתה בו. אם זאת נראה שרוב המחשבים ברשת עדיין מריצים את הקליינט באופן אוטומטי אחת לכמה שעות.

טכנו"צ סייבר - תרגיל 7 - "W0RM"

השיטה

במהלך המחקר של קליינט האיימיל הצלחנו לאפיין את פרוטוקול התקשורת של הקליינט. הפרוטוקול מבוסס json ומסתמך על פרטי הזדהות ששמורים ליד הקליינט בקובץ settings.dat.

הפרוטוקול מורכב משלוש פאקטות שונות:

- get_contacts -> {'ID': 0, 'user': username, 'pass': password}

  פאקטה זו משמשת כדי לגלות לאילו משתמשים המשתמש user יכול לשלוח מיילים

- write_mail -> {'ID': 1, 'user': username, 'pass': password, 'to': dest, 'subject': subject, 'message': message}

פאקטה זו משמשת כדי לכתוב מיילים למשתמשים אחרים ברשימת אנשי הקשר. הפאקטה תחזיר אם היא הצליחה לשלוח את המייל או לא.

- get_mail-> {'ID': 2, 'user': username, 'pass': password}

  פאקטה זו משמשת כדי לקבל חזרה את רשימת המיילים **החדשים** שהמשתמש קיבל.

שימו לב שכל פעם שמבקשים את רשימת המיילים מהשרת הוא ישלח רק את אלו שהמשתמש לא קיבל עדיין.

בנוסף גילינו שמבנה קובץ ה־settings.dat גם הוא json והוא בנוי בצורה הבאה:

{"user": "doofen", "password": "perry", "server": "localhost", "port": "25565"}

לבסוף הצלחנו להשיג את קוד המקור של הקליינט שכתוב ב־cpp והבנו שהפונקציה החשודה (**והיחידה שאתם צריכים להסתכל עליה**) היא הפונקציה:

save_mail(const char* from, const char* subject, const char* data)

החשד הנוכחי הוא שהפונקציה מכילה חולשת shell injection.

טכנו"צ סייבר - תרגיל 7 - "W0RM"

שימו לב: יש בונוס מעשי שמהווה תחליף מלא לתרגיל ומזכה בנקודה לציון הסופי, במידה והצלחתם אותו. אם אתם מעוניינים בכך רדו עכשיו לתחתית התרגיל ותקראו עליו קצת לפני הגרסה הרגילה של התרגיל.

שלבי התרגיל

שלב ראשון - הכנת הקרקע

הורידו את הקבצים מהמודל והריצו (לחצו פעמיים) את הקובץ install library.bat. קובץ זה אמור להתקין לכם את הספריות - במקרה הזה ספרייה - הדרושות לתרגיל. כעת הריצו את הקובץ start server.bat וודאו שהוא עובד. הוא אמור להציג לכם:

```
Starting server !
Server Online ;)
```

שלב שני - פונקציות הבסיס

כעת נתחיל לכתוב את הפתרון שלנו, פתחו הקובץ solution_skeleton.py. עברו עליו בזריזות וודאו שאתם מצליחים להבין מה כל פונקציה עושה (עד כדי פונקציית ה־main) מהתיעוד של הפונקציות.

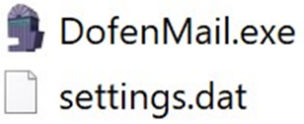
כעת מלאו את שלושת הפונקציות:

- craft_user_list_packet
- craft_mail_write_packet
- craft_mail_get_packet

כעת שחקו קצת עם הקוד בפונקציית ה־main וודאו שאתם יודעים איך לקבל את ה־contacts שלכם, לכתוב מייל לאדם חדש או לקבל את רשימת המיילים הנוכחית (אמורה להיות ריקה).

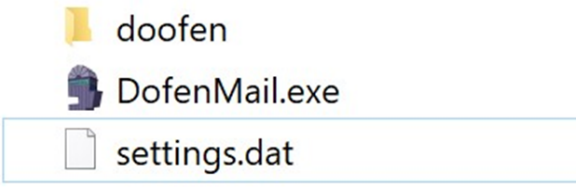
שימו לב: כדי שתוכלו לקרוא לפונקציות האלו, השרת צריך להיות דלוק!

במידה ואתם רוצים לבדוק שאתם מסוגלים לשלוח מיילים, אתם מוזמנים לבדוק את המיילים של dora; היכנסו לתיקיית users, משם לתיקייה של dora, והפעילו את התוכנה DofenMail.exe.



במידה והצלחתם לשלוח לדורה מייל התיקייה אמורה להיראות בצורה הבאה:

טכנו"צ סייבר - תרגיל 7 - "W0RM"



ובתוך התיקייה doofen אמור להיות קובץ שתוכנו הוא המייל שלכם.

שלב שלישי - מציאת החולשה

בשלב זה נחפש את החולשה שאותה נשמיש במהלך התרגיל. פתחו את הקובץ DofenMail.cpp ורדו לפונקציה save_mail. קראו את התיעוד המסופק אליה ואת הקוד ונסו לחשוב מה החולשה בה וכיצד ניתן לנצל אותה (רמז: קראו על **shell injection**).

```
void save_mail(const char* from, const char* subject, const char* data)
{
    char buffer[MAX_BUFF] = { 0 };
    sprintf(buffer, "if not exist \"%s\" mkdir \"%s\"", from, from);
    system(buffer);
    sprintf(buffer, "echo %s > \"%s\"/\"%s\"", data, from, subject);//Quick trick to quickly save files ;)
    //printf("%s\n", buffer);
    system(buffer);
}
```

לאחר שנראה שהצלחתם להבין את החולשה השתמשו ב־CMD כדי לוודא שאתם מצליחים להבין איך לנצל אותה (ממש תחליפו את המחרוזות כפי ש־sprintf היה עושה). נסו לחשוב כיצד ניתן לגרום לקטע הקוד הזה להדפיס "hello world" למסך בצורה זדונית.

שימו לב שבאמצעות && ניתן לחבר פקודות לדוגמא:

echo hello World && echo hello line

יפעיל את שתי הפקודות אחת אחרי השניה.

שלב רביעי - POC - Proof Of Concept

מהשלב הזה נתחיל לנסות לנצל את החולשה, שלחו לדורה payload שגורם לתוכנה שלה להדפיס "Hello world" למסך. אמור להיראות משהו בסגנון הזה:

```
C:\Users\NacOJ\Desktop\Academy\Cyber\Ex7\users\dora>DofenMail.exe
{"ID": 2, "mail_count": 1, "mails": [{"from": "doofen", "subject":  [redacted] ta": "There once was a donkey who loved to run"}]}
{"data":"There once was a donkey who loved to run","from":"doofen","subject":[redacted]"}
Hello World!"
```

שימו לב, הרצתי את התוכנה באמצעות CMD כדי לראות את כל הפלט שלה. בנוסף בתמונה עצמה השחרתי חלקים מהמייל (ספציפית שדה ה־subject) כדי לא לעשות ספוילרים לתרגיל; החלקים החשובים מסומנים בחצים אדומים.

שלב חמישי - הפצה נרחבת

בשלב זה נרצה להתאים את קובץ הפייתון שלנו כדי שיתאים לתולעת; נרצה שהקובץ שלנו יוכל בעצמו להבין מי האנשים שהוא יכול לשלוח להם מייל ומי המשתמש שהוא רץ דרכו באותו הרגע.

טכנו"צ סייבר - תרגיל 7 - "WORM"

שנו את הקוד כך שידפיס תחילה עבור איזה משתמש הוא רץ, ואז ישלח לכל האנשים ברשימת הקשר של המשתמש את ה־payload משלב 4.
הקובץ simulate_users.bat מדמה גישה למייל של כל המשתמשים בשרת; כאשר נריץ אותו נצפה לראות שגם עבור dora וגם עבור swiper מודפס "Hello World".

שלב שישי - מניעת הדבקה חוזרת

הקובץ simulate_users.bat מדמה גישה למיילים 3 פעמים, ולכן אם כל מחשב יפיץ את הוירוס נאיבית זה ייצור תקשורת הלוך ושוב, בכמות שגדלה מעריכית ותחשוף את המבצע. לכן, כאשר נתקדם בפרצה שלנו נרצה לוודא שמחשב שכבר הודבק לא יודבק שוב - בשביל לעשות זאת נשתמש בקובץ זיהוי שמטרתו לסמן האם כבר הדבקנו את המחשב או לא.
עבור הוירוס שלנו נשתמש בקובץ בשם touched (הסימולציה יודעת למחוק אותו בתחילת כל ריצה).
קראו באינטרנט כיצד ניתן להריץ פקודות CMD רק במידה וקובץ מסויים לא קיים והריצו את הפקודות הבאות:

1) הפקודה הראשונה תיצור את הקובץ touched כדי למנוע הדבקה חוזרת
2) הפקודה השנייה תדפיס למסך "Hello World"

כעת נרצה לוודא שהקוד שלנו רץ ועובד כמו שצריך, הפעילו את ה־solution שלכם ואז את simulate_users.bat. נצפה לראות בסבב הראשון שוב ש־dora ו־swiper מדפיסים "Hello World". כעת בסוף הסבב הראשון (אבל לפני הסבב השלישי), הריצו שוב את ה־solution. אם הכל עבד כמו שצריך תראו בסבב שלוש ש־dora ו־swiper קיבלו את המיילים שלהם אבל הם לא ידפיסו "Hello World".

שלב שביעי - הסתרה

שלב זה יחסית מהיר למימוש; הוסיפו ל־payload (פקודת ה־CMD) קוד שידאג "להעלים" את touched אחרי שהוא נוצר, כלומר תהפכו אותו ל־hidden-file (חפשו בגוגל איך).

טכנו"צ סייבר - תרגיל 7 - "W0RM"

שלב שמיני - שכפול הנוזקה

בשלב הזה נתחיל באמת להפיץ את הנוזקה לשאר המחשבים. נפעל מנקודת הנחה מקלה שיש לכל היעדים שלנו פייתון על המחשב. נשתמש ב־python -c כדי להריץ קוד מה־CMD וכך נוכל לגרום ל־payload שלנו להתחיל להריץ קוד פייתון:

```
C:\Users\NacOJ>python -c "print('Hello Cyder'); print('This is another test')"
Hello Cyder
This is another test
```

כעת נשתמש בקידוד base64 ובפייתון כדי להדפיס למסך את התוכן של הקובץ שלנו (קיימת לכם כבר פונקצית עזר שמחזירה לכם את הקוד).

כעת:

1. שמרו את הפלט הזה לקובץ באמצעות הפניית הפלט של CMD
2. הריצו את קובץ הפייתון שיצרתם
3. מחקו את הקובץ לאחר הריצה

כעת הפעילו את השרת, הריצו את ה־solution והפעילו את ה־simulator. אם הכל עובד אתם אמורים לראות יותר ויותר אנשים שמריצים קוד מפעם לפעם.

שלב תשע - זיהוי היעד

כעת הוסיפו לתולעת שלכם קטע קוד שיודע לזהות האם אנחנו נמצאים במחשב של ה־CEO, ובמידה וכן התולעת תיצור קובץ בשם HelloWorld.

כעת הריצו שוב את הקוד ומיד לאחריו את ה־simulator, במידה והצלחתם תקבלו את הפלט הבא:

```
You are indeed a pro hacker cracker
Removed the file so you can try again

If you wish to retry the test just re-run the program
Press any key to continue . . .
```

שלב עשר - מחשבות להמשך

חשבו על השאלות הבאות וכתבו את התשובות בקובץ ה־README:

1) חשבו כיצד יכולתם להדליף את הסיסמא של המחשב של ה־CEO.
2) חשבו כיצד הייתם יכולים לנצל את החולשה במידה והייתה לכם שליטה על האינטרנט של החברה של דופנשמירץ (על מה שנכנס ויוצא מהרשת שלה), אך ללא שליטה ויכולת להריץ קוד על המחשב שלו.
3) נסו לזהות עוד פרצות בקליינט האימייל.

טכנו"צ סייבר - תרגיל 7 - "W0RM"

הגשת התרגיל

עליכם להגיש את קובץ ה solution ואת קובץ ה-README בתוך זיפ בשם ex7_[FullName].zip

בהצלחה!

טכנו"צ סייבר - תרגיל 7 - "W0RM"

בונוס - מעשי - קשה

כיאה לתרגיל אחרון, הבונוס המעשי לתרגיל הזה יהיה קשה ויזכה את העושים אותו בנקודת בונוס לציון הסופי. **הבונוס הינו גרסה שונה לתרגיל, כך שמי שיעשה אותו פטור מהגרסה הרגילה.** בגרסה זו של התרגיל, במקום להשמיש shell injection נשמיש את החולשה המפורסמת, שהתגלתה השנה, שהייתה קיימת בספרייה log4j; מלבד זאת כלל השלבים זהים, ובפרט עליכם להגיע גם פה למסך הסיום:

```
You are indeed a pro hacker cracker
Removed the file so you can try again

If you wish to retry the test just re-run the program
Press any key to continue . . .
```

בנוסף, עליכם להשמיש אותה ללא הרבה הדרכה; נהיה נחמדים ונגיד שהחולשה היא בקטע הקוד הבא:

```
Client c = new Client("settings.dat");
JSONObject get_mail = new JSONObject();
get_mail.put("ID", 2);
c.connect();

JSONObject response = c.sendPacket(get_mail);
c.disconnect();
JSONArray array = (JSONArray)response.get("mails");
for(int i = 0; i < array.size(); ++i)
{
    JSONObject mail = (JSONObject) array.get(i);
    String from = (String) mail.get("from");
    String subject = (String) mail.get("subject");
    String data = (String) mail.get("data");

    logger.error("Received a new message from " + from + " with title " + subject);

    from = from.split("\\\\")[0];
    subject = subject.split("\\\\")[0];
    from = from.split("/")[0];
    subject = subject.split("/")[0];//No path traversal on me

    File fromF = new File(from);//mmm might be a security threat as well
    if(!fromF.exists())
        fromF.mkdir();
    File path = new File(from, subject);
    FileWriter subjectF = new FileWriter(path.getPath());
    subjectF.write(data);
    subjectF.close();
}
```

טכנו"צ סייבר - תרגיל 7 - "W0RM"

נספר גם שאפשר להשתמש בסדנה הבאה, שאחד.ת מהמתרגלים כתב.ה. שימו לב כי יש סט קבצים **אחר במודל עבור הבונוס, השתמשו בו.**

פרטים חשובים עבור הבונוס

כבו את כל ה־firewallים שיש לכם; קיים סיכוי שתצטרכו גם להוסיף exception בשביל התרגיל, כי הרבה מהם אגרסיביים לגבי מניעת ההשמשה של log4j (שזה סימן טוב בסך הכל, כמובן). מומלץ לעבוד לפי השלבים שמתוארים בתרגיל עצמו, הם נוחים ומקלים מאוד על העבודה. מצד אחד ההשמשה הראשונית יותר קשה, ומצד שני מהרגע שאתם מסיימים אותה אתם כותבים את רוב הקוד ב־Java וזה יותר נחמד.

הגשת הבונוס

הגישו את כל הקבצים בתוך קובץ זיפ בשם ex7Bonus_[FullName].zip. **הוסיפו לתוך הזיפ קובץ וידאו שמסביר מה עשיתם, איך הדברים עובדים והדגימו ריצה של הבונוס.**