

Cyber Security

Penetration Testing final project

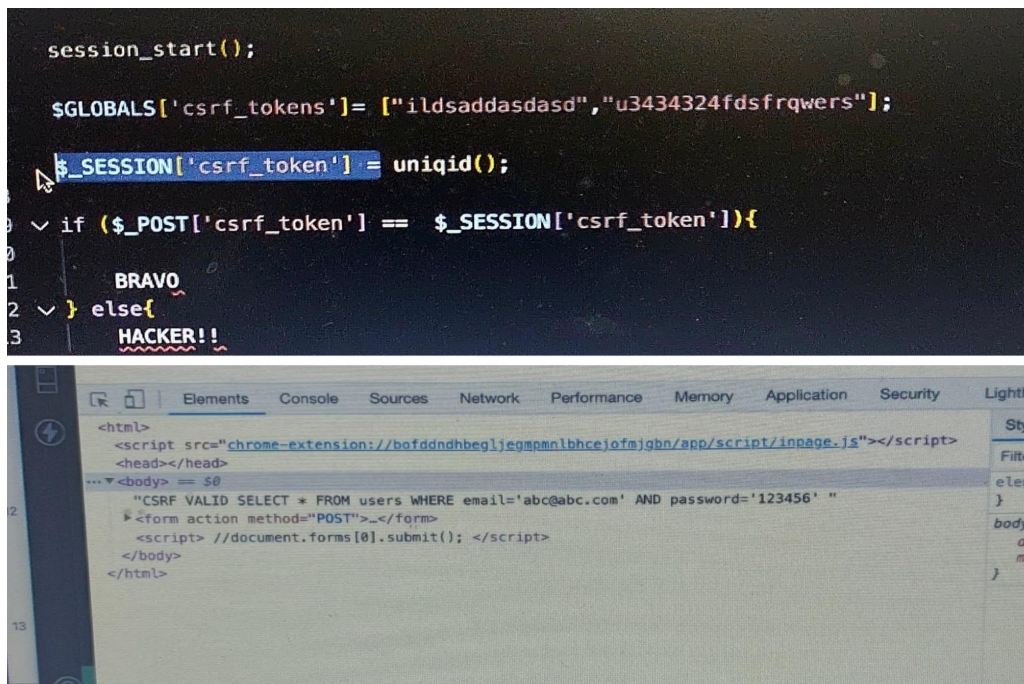
Maor & elhdad

Test Name- Cross Site Request Forgery CSRF attack

Description

CSRF

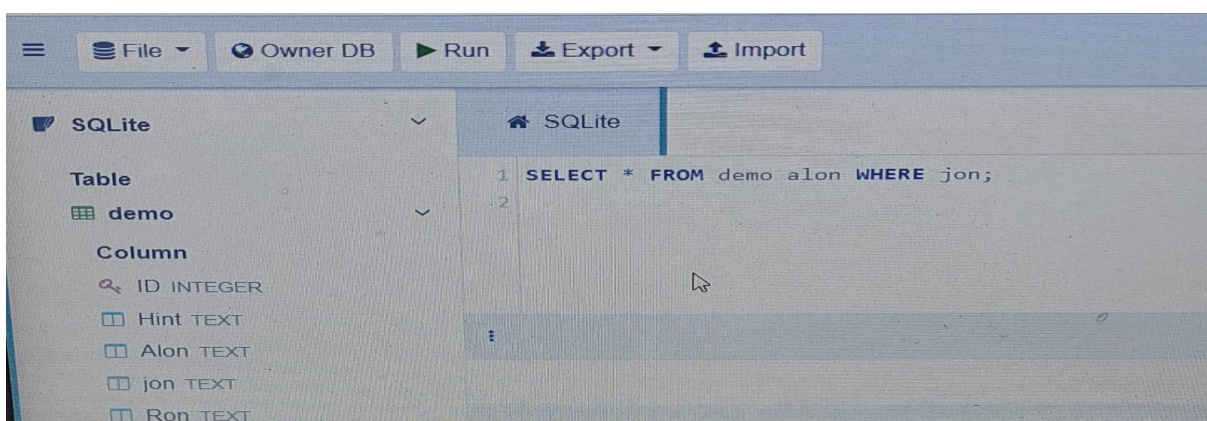
היא התקפה נגד גולש אינטרנט המאלצת את דפדפן האינטרנט של הגולש לבצע פעולות לא רצויות ביישומי אינטרנט בשמו של המשתמש בשירות באמצעות ניצול מגבלות בפרוטוקול HTTP.



Test Name-SQL injection

Description

הזרקת SQL היא שיטה לניצול פרצת אבטחה בתוכנית מחשב בעזרת פניה אל מסד הנתונים. השם נובע מכך שהמשתמש מכניס קוד SQL לשדה קלט אליו אמורים היו להיכנס נתונים תמימים. באופן זה יכול משתמש זדוני לחרוג לחלוטין מן התבנית המקורית של השאילתה, ולגרום לה לבצע פעולה שונה מזו שיועדה לה במקור



Test Name- Remote Code Execution attack

Description

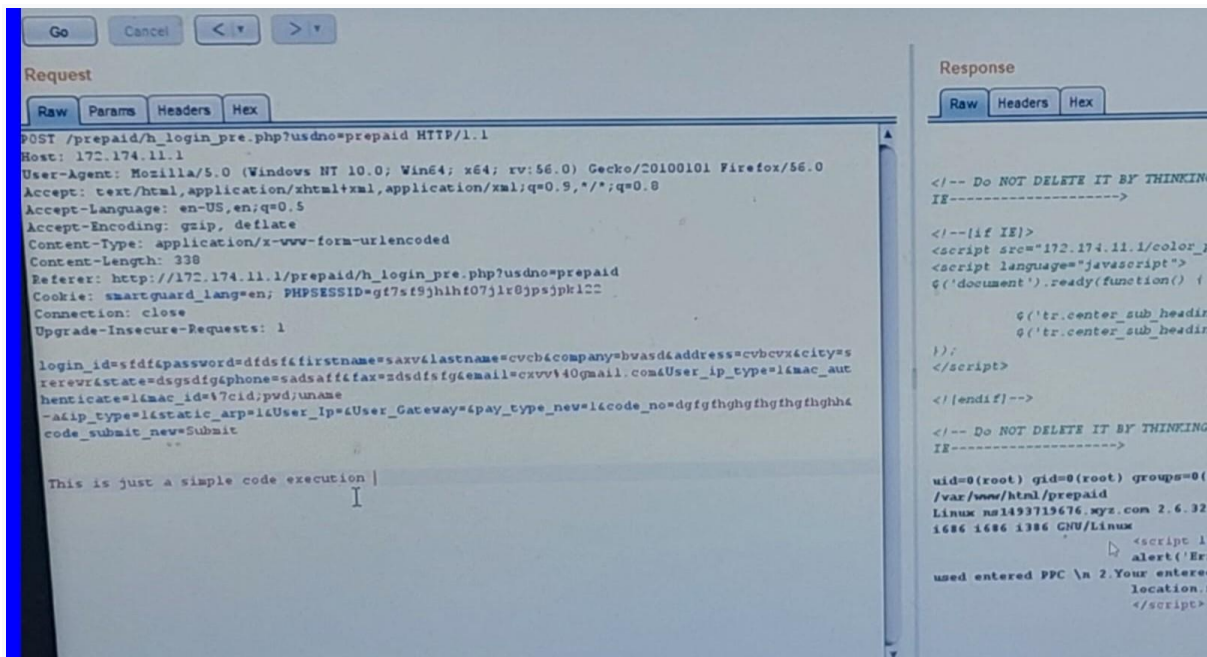
הוא סוג של פגמי אבטחת תוכנה/פגיעויות המאפשרות (RCE) ביצוע קוד מרוחק לתוקף לבצע כל קוד לבחירתו במחשב מרוחק דרך האינטרנט ללא גישה ישירה למעטפת אינטראקטיבית.

באפליקציית אינטרנט וכולן RCE ישנן מספר דרכים שבהן ניתן להשיג סלנצל לוגיקה/קוד לא מאובטח שהאפליקציה פועלת

RCE-שלושת הסוגים העיקריים של פגיעויות שיכולות להוביל ל

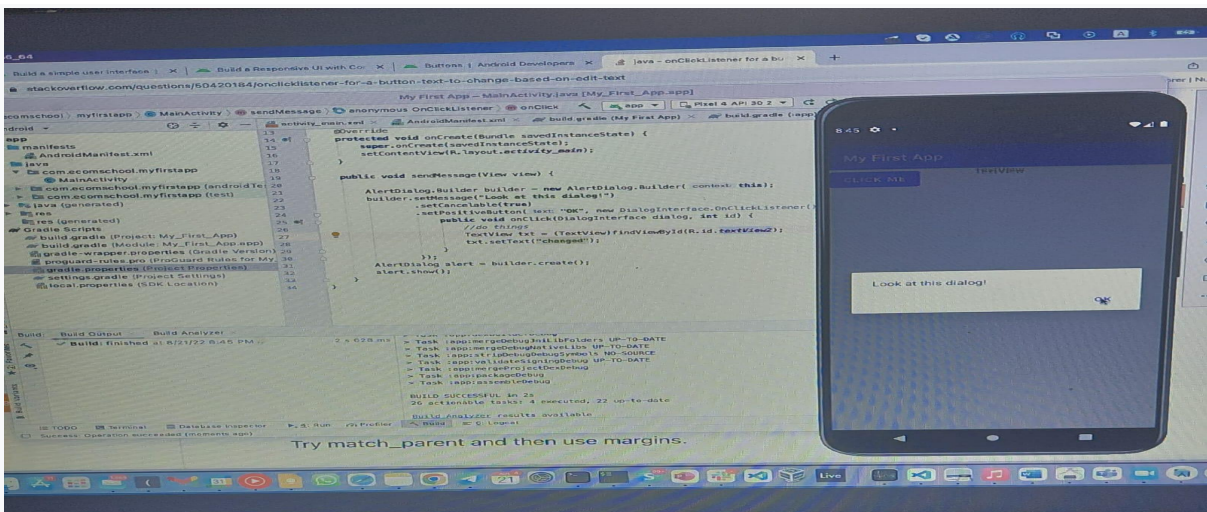
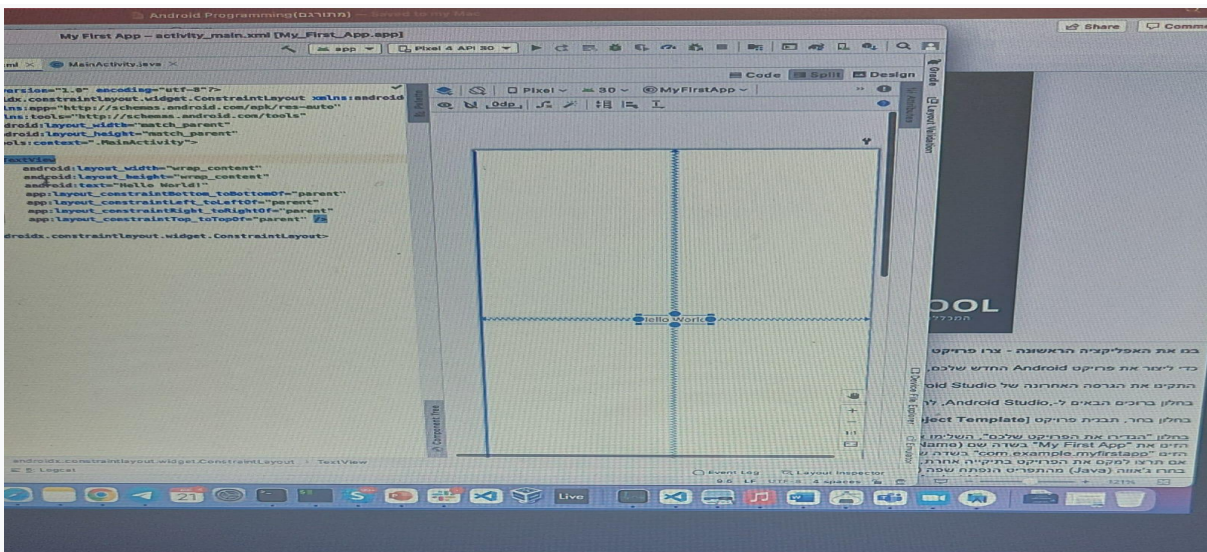
רוב התקפות RCE על אפליקציות אינטרנט מודרניות נגרמות על ידי אחת מהתקפות אלו.

1. העלאת קבצים שרירותית
2. הזרקת פקודה
3. הזרקת קוד



Test Name- Android Studio visual Description

אנדרואיד היא ערימת תוכנה בקוד פתוח
נוצר עבור מגוון רחב של מכשירים עם
גורמי צורה שונים. הראשי של אנדרואיד
המטרה היא ליצור תוכנה פתוחה
OEM פלטפורמה זמינה עבור ספקים, יצרני
מפתחים ליצור את הרעיונות החדשניים שלהם
מציאות ולהציג מוצלח
OEM(Original Equipment Manufacturer)
כלומר יצרן ציוד מקורי

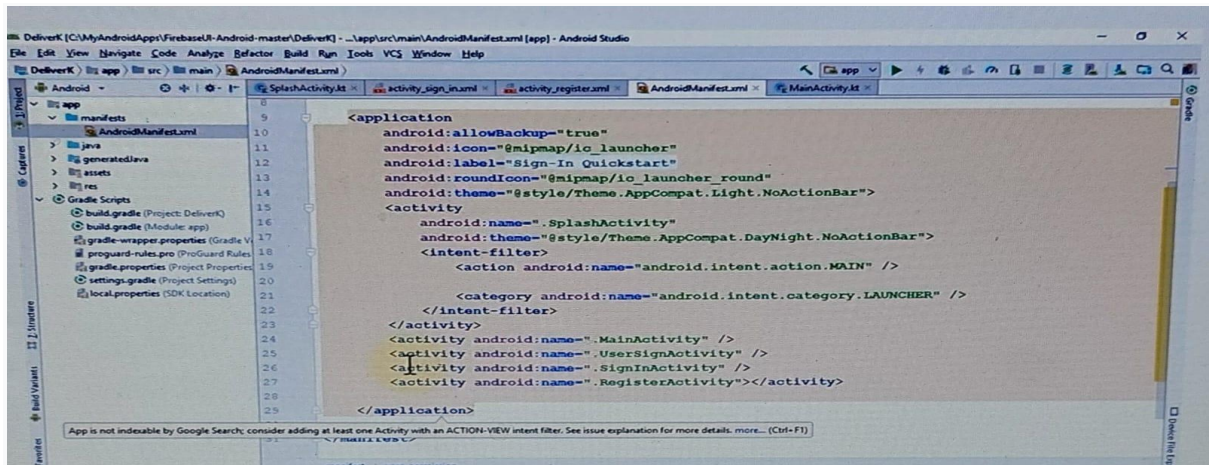


Test Name MANIFEST.MF

Description

manifest.mf

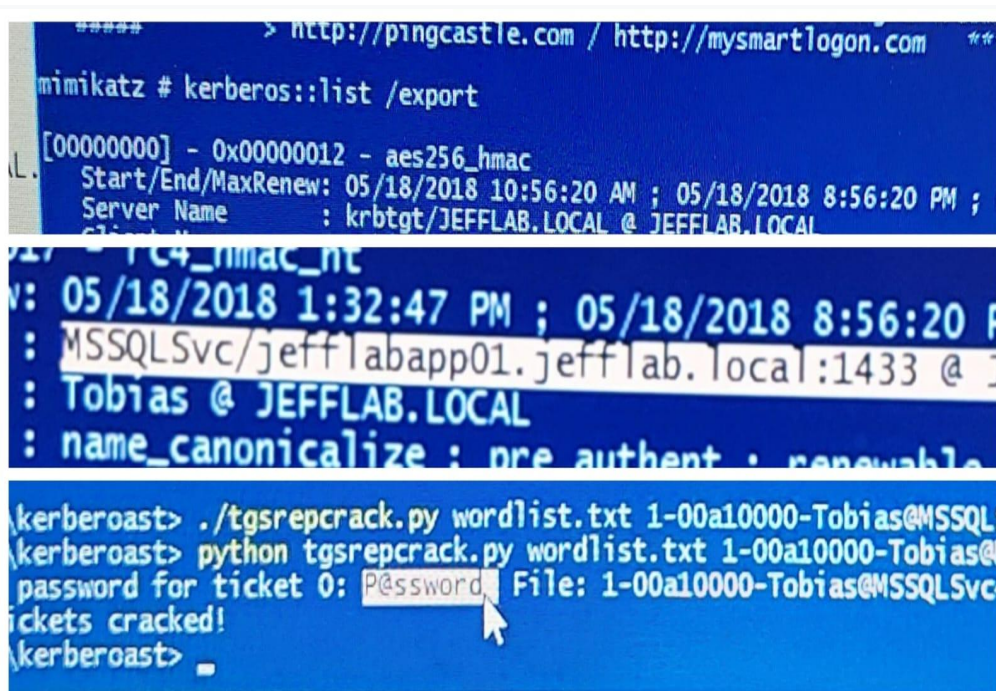
מכיל מידע שונה המשמש את סביבת זמן הריצה של JAVA כאשר
טעינת קובץ Jar שהיא המחלקה הראשית שיש להפעיל קובץ Jar
הגרסה של החבילה רשימת שמות קבצים בצננת יחד עם תקצירי SHA1 שלהם.



Test Name Kerberoasting

Description

היא שיטת התקפה המאפשרת לתוקף לפצח את הסיסמאות של חשבונות
שירות Active Directory במצב לא מקוון כדי למנוע זיהוי למידע נוסף
על התקפה זו וכיצד למתן ולזהות אותה.



Test Name 3-Way Handshake

Description

3-Way Handshake attack

תהליך לחיצת יד תלת כיוונית מתוכנן בצורה כזו ששני הקצוות עוזרים לך ליזום, לנהל משא ומתן ולהפריד חיבורי שקע TCP בו זמנית. זה מאפשר לך להעביר מספר חיבורי שקע TCP בשני הכיוונים בו זמנית אלו 3 השלבים:

משמש כדי ליזום וליצור חיבור עוזר לסנכרן מספרי רצף בין מכשירים-SYN

ACK עוזר לאשר לצד השני שהוא קיבל SYN

הודעת SYN מההתקן מקומי ו-ACK של החבילה הקודמת

משמש לניתוק חיבור-FIN

The screenshot displays a network traffic capture in Wireshark. The top section shows a list of packets, including Telnet data and TCP ACKs. The bottom section provides a detailed view of a TLSv1.3 Record Layer and Handshake Protocol: Server Hello. The handshake details include:

- Content Type: Handshake (22)
- Version: TLS 1.2 (0x0303)
- Length: 122
- Handshake Protocol: Server Hello
- Handshake Type: Server Hello (2)
- Length: 118
- Version: TLS 1.2 (0x0303)
- Random: 30707c9a4510fae9882f687227b4be070ca18c8c045701746

The packet list on the right shows the following details:

No.	Time	Source	Destination
3	0.051374	0.000128	192.168.0.46
4	0.051658	0.000284	192.168.0.46
5	0.129374	0.077716	192.168.0.46
6	0.141320	0.011946	192.168.0.46
7	0.141320	0.000000	192.168.0.46
8	0.141414	0.000094	192.168.0.46
9	1.922015	1.780601	192.168.0.46
10	1.922333	0.000318	192.168.0.46
11	1.922724	0.000391	192.168.0.46
12	1.947282	0.024558	192.168.0.46

Test Name Man in the middle (MITM)

Description

Man in the middle (MITM) attack

מתקפת אבטחת מידע המתקפה נועדה להתקפת האזנה (Sniff) מידע הקשור בין שתי ישויות "מחשב", טלפון סלולרי ומערכות מידע על ידי כך שהוא מתחזה ליישות האמצעית זאת שמעבירה את המידע בין הישויות כדי לצלוח בהתקפה זאת חייב התוקף להתחזות

למעביר ההודעה (Router, סוויץ) באופן ברירת מחדל כל המידע מוסט ממעביר הודעה אל אותו תוקף ורק אחרי שעבר אצל התוקף הוא מועבר אל מעביר ההודעה

בתרשים ניתן לראות את התוקף באדום עד כה העבירו מחשב 1 ומחשב 2 הודעות דרך ראוטר בתרשים אבל ברגע שהתוקף התחבר למערכת דרך Wifi או חיבור פיזי או דרך התחברות מרחוק אל המחשב הוא גרם לשתי מחשבים לאחד מהם לחשוב שהוא הראוטר (Default gateway) וכך כל המידע מוזרם דרך התוקף ורק אחרי זה עובר אל הראוטר כך צולח התוקף לייטר את כל המידע עוד לפני שיצא או חזר מהאינטרנט בסיום קליטת המידע הוא מבצע העברה אל הראוטר והמותקף לא יודע כלל שהוא השתנה

The diagram illustrates a Man-in-the-Middle (MITM) attack. It shows three devices: PC2 on the left, PC1 on the right, and a Laptop at the bottom. A central Router is labeled with the MAC Address CF:C1:F4:A2:EC:3D. Red arrows indicate the flow of traffic: PC2 and PC1 send traffic to the Router, and the Router sends traffic to the Laptop, which then forwards it to the Router and back to PC2 and PC1. This setup allows the Laptop to intercept and potentially alter the communication between the two PCs.

The screenshot below shows a network tool interface, likely Wireshark, displaying a list of captured packets. The interface includes a packet list pane on the left, a packet details pane in the middle, and a packet bytes pane on the right. The packet list pane shows a table of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data of the selected packet in hexadecimal and ASCII.

NBT-NS, LLMNR & MDNS Responder 3.0.7.0
Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

[+] Poisoners:
LLMNR [ON]
NBT-NS [ON]
DNS/MDNS [ON]
DHCP [OFF]

Test Name Python

Description Scapy Attack Chain

משמש בודקי חדירה והאקרים לרשת Scapy בשביל

לשלוח ולנתח מנות

רחרח תעבורת רשת

סריקת רשתות

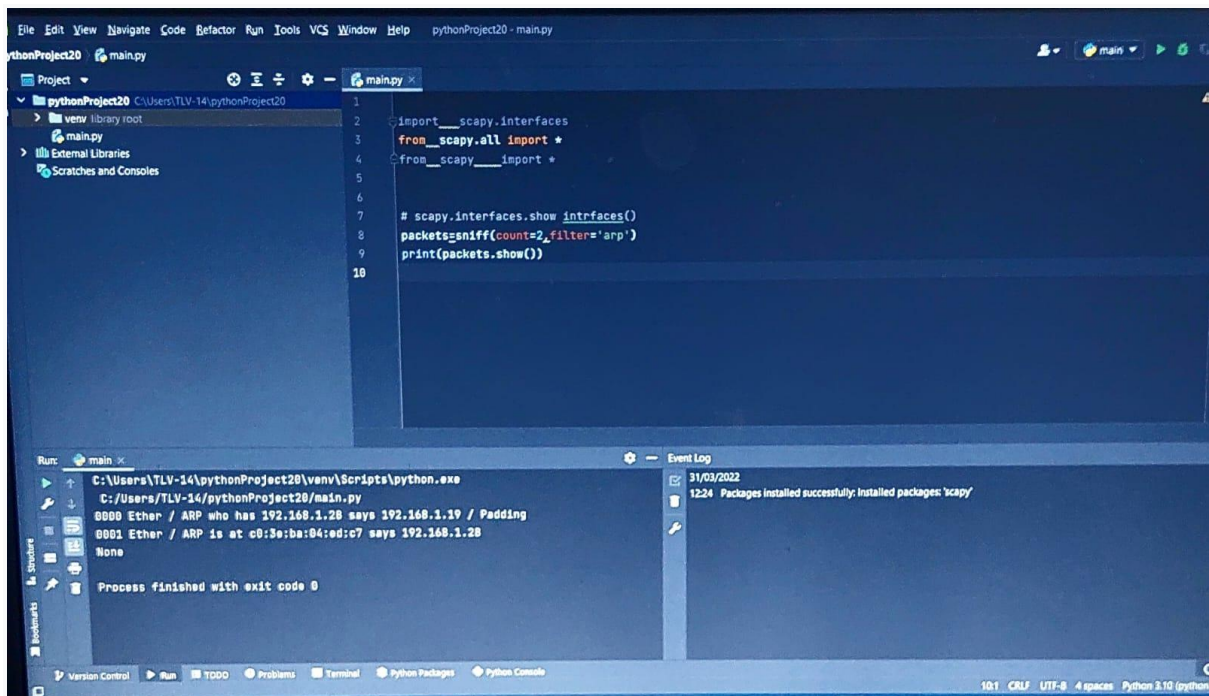
תקשורת מזויפת

מניפולציות זיוף מנות

נשתמש בפונקציה sniff ל- filter

(ARP)

Address Resolution Protocol

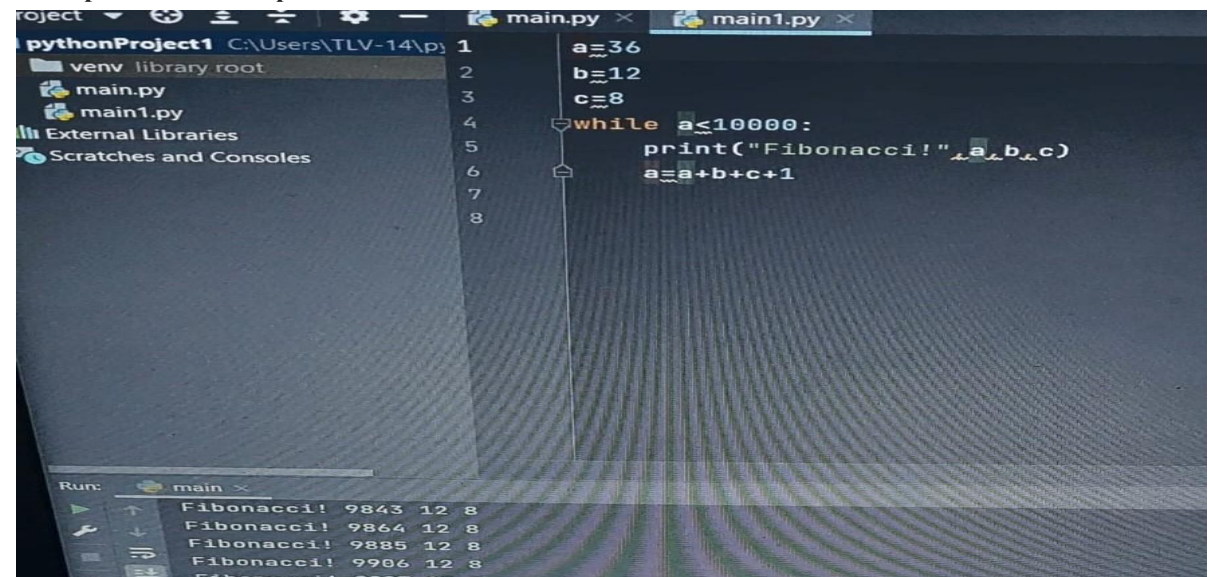


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject20 - main.py
pythonProject20 main.py
Project
pythonProject20 C:\Users\TLV-14\pythonProject20
venv library root
main.py
External Libraries
Scratches and Consoles
1
2 import __scapy.interfaces
3 from __scapy.all import *
4 from __scapy__ import *
5
6
7 # scapy.interfaces.show_interfaces()
8 packets=sniff(count=2,filter='arp')
9 print(packets.show())
10

Run: main.exe
C:\Users\TLV-14\pythonProject20\venv\Scripts\python.exe
C:/Users/TLV-14/pythonProject20/main.py
0000 Ether / ARP who has 192.168.1.28 says 192.168.1.19 / Padding
0001 Ether / ARP is at c0:3e:ba:04:ed:c7 says 192.168.1.28
None
Process finished with exit code 0
Event Log
31/03/2022
12:24 Packages installed successfully: Installed packages: 'scapy'
```

Test Name Python

Description While Loop Fibonacci Attack



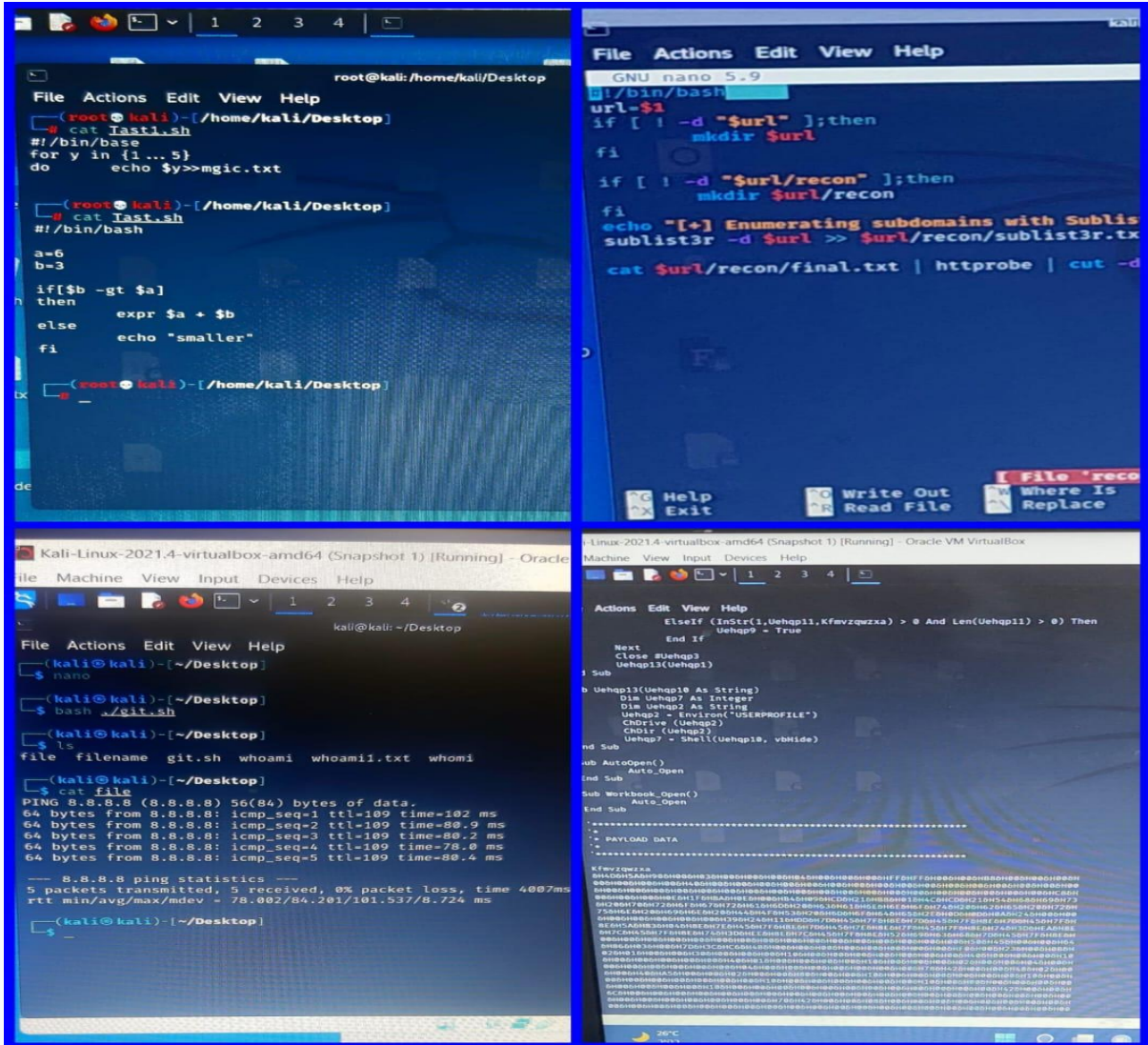
```
pythonProject1 C:\Users\TLV-14\p
1 a=36
2 b=12
3 c=8
4 while a<10000:
5     print("Fibonacci!" ,a,b,c)
6     a=a+b+c+1
7
8

Run: main.exe
Fibonacci! 9843 12 8
Fibonacci! 9864 12 8
Fibonacci! 9885 12 8
Fibonacci! 9906 12 8
Fibonacci! 9927 12 8
```

Test Name kali linux

Description Script Bin base

hash and ping 5 packets



Bonus for finishing a report PT

ניתן לראות פונקציה של 3 דפים סינגל אנימציה קוד HTML

```
    }
    h1{
    position: relative;
    animateion-nsme:move;
    animateion-duration:2s;
    animateion-fill-mode:forwards;
    animateion-iteration-count:infinite;
    }
  </style>
  <script>
  document.addEventListener('DOMContentLoaded',function(){
  document.querySelector('button').onclick=>{
  const h1=document.querySelector('h1');
  if (h1.style.animateionPlayState=== 'paused');){
  h1.style.animateionPlayState='runnig';
  }else{
  h1.style.animateionPlayState='paused';
  }
  }
  }
```

```
<html lang="en">
<head>
<title>Animate</title>
<style>

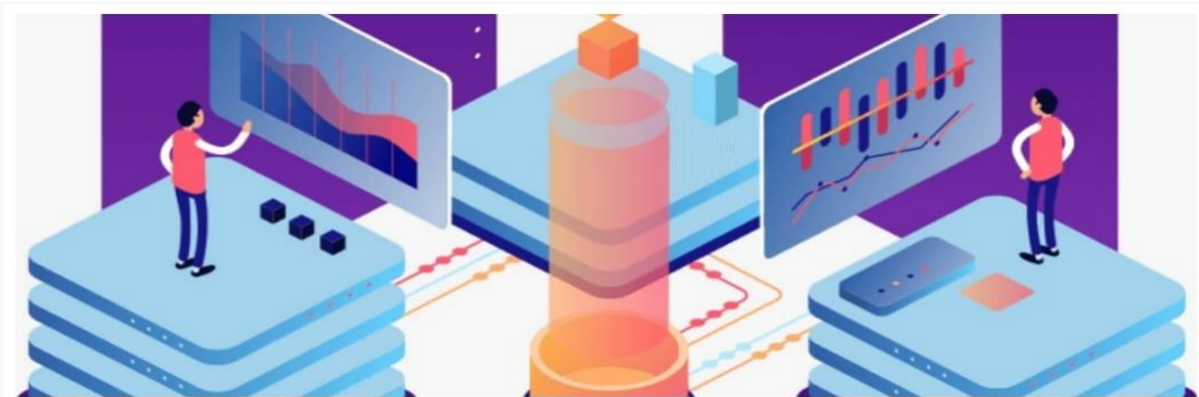
  @Keyframes move {
    from{
      font-size: 20px;
    }
    to{
      font-size: 100px;
      left: 0%;
    }
    to{
      left: 50%;
    }
  }

  h1{
  position: relative;
  animateion-nsme:move;
  animateion-duration:2s;
  animateion-fill-mode:forwards;
  animateion-iteration-count:infinite;
  }
```

```
code Refactor Run Tools VCS Window Help ce me - C:\Users\TLV-14\Documents\Maor
MaorSinglepage.txt
MaorSinglepage.txt
1 singlepage.html<html>body>div#page3>h1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5 <title>Single page</title>
6 <style>
7   div {
8     display: none;
9   }
10 </style>
11 <script>
12   function showPage(page){
13
14     document.gurySelectorAll('div').forEa
15   })
16   document.gurySelector("#${page}").style.
17   }
18   document.addEventListener('DOMContentLoaded
19   document.gurySelectorAll('button').forEach(bu
20   button.onclick=function(){
21     showPage(this.dataset.page);
22   }
23   });
24 </script>
25 </head>
26 <body>
27 <button data-page="page1"</button>
28 <button data-page="page2"</button>
29 <button data-page="page3"</button>
30
31
```

Attempts to break into the project website

PT



```
File Actions Edit View Help
(root@kali)-[~/home/kali]
# sudo nmap -sVC -T4 --top-ports 75 -vv {ip}

Starting Nmap 7.92 ( https://nmap.org ) at 2022-08-31 14:52 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 14:52
Completed NSE at 14:52, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 14:52
Completed NSE at 14:52, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
```

```
[~] Failed to load module: port
msf6 > use tcp http://138.197.64.192

Matching Modules
-----
#  Name                                                                                               Disclosure Da
-  -                                                                                               -
0  auxiliary/dos/scada/igss9_dataserver                       2011-12-20
1  payload/aix/ppc/shell_bind_tcp
2  payload/aix/ppc/shell_reverse_tcp
3  payload/android/meterpreter_reverse_tcp
4  payload/android/meterpreter/reverse_tcp
5  auxiliary/gather/zookeeper_info_disclosure                 2020-10-14
6  auxiliary/dos/http/apache_mod_isapi                       2010-03-05
7  payload/osx/armle/shell_bind_tcp
8  payload/osx/armle/shell_reverse_tcp
```

```
(root@kali)-[~/home/kali]
# rcrack http://138.197.64.192//
RainbowCrack 1.8
Copyright 2020 RainbowCrack Project. All rights reserved.
http://project-rainbowcrack.com/

usage: ./rcrack path [path] [...] -h hash
       ./rcrack path [path] [...] -l hash_list_file
       ./rcrack path [path] [...] -lm pwdump_file
       ./rcrack path [path] [...] -ntlm pwdump_file
path:  directory where rainbow tables (*.rt, *.rtc) are stored
-h hash: load single hash
-l hash_list_file: load hashes from a file, each hash in a line
-lm pwdump_file: load lm hashes from pwdump file
-ntlm pwdump_file: load ntlm hashes from pwdump file
```