

WEB SECURITY

Project report submitted in partial fulfilment of requirement for the degree of

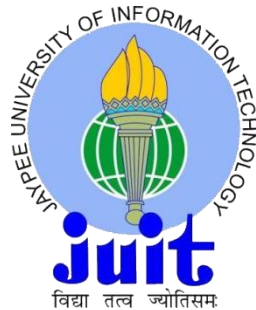
BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING

By

Ramit Bawa (171042)

UNDER THE GUIDANCE OF

Dr. Naveen Jaglan



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

DECEMBER 2020

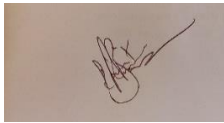
TABLE OF CONTENTS

CAPTION	PAGE NO.
DECLARATION	i.
ACKNOWLEDGEMENT	ii.
LIST OF ABBREVIATIONS	iii.
LIST OF FIGURES	iv.
ABSTRACT	v.
CHAPTER 1: INTRODUCTION	11
1.1.General Background	11
1.2.Problem statement	12
1.3.Objective	13
1.4.Scope of the project	14
CHAPTER 2: LITERATURE SURVEY	16
CHAPTER 3: WEB SECURITY	18
3.1. Web threats and their sources	18
3.1.1. Threats	18
3.1.2. Sources	18
3.2. Evolution of cyber threats	18
3.3. Steps taken for web security	19
3.3.1. Network security	19
3.3.2. Protection from malware	19
3.3.3. Monitoring	20
3.3.4. Incident management	20
3.3.5. Knowledge-full and aware users	20
3.3.6. Mobile work and home work	20

3.3.7. Encrypt config.	20
3.3.8. Controls for removable media contents	20
3.3.9. Maintaining privileges for users	20
3.3.10. Info risk management	21
3.4. Cyber security	21
3.5 ARP spoofing	23
CHAPTER 4: RESULTS AND DISCUSSION	24
CONCLUSION AND FUTURE WORK	34
REFERENCES	35
APPENDIX	36

DECLARATION

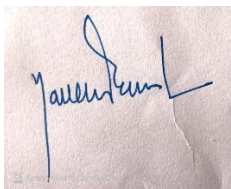
I hereby declare that the work reported in the B.Tech Project Report in the final year entitled “Web Security” submitted at **Jaypee University of Information Technology, Wagnaghat, Solan, India** is an authentic record of our work carried out under the supervision of **Dr. Naveen Jaglan**. I have not submitted this work elsewhere for any other degree, diploma or course activity.



Ramit Bawa

171042

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.



Dr. Naveen Jaglan

Date:

Head of the Department/Project supervisor

i

ACKNOWLEDGEMENT

I would like to thank God for guiding me throughout my academic journey and to acknowledge my project supervisor, Dr Naveen Jaglan, for his undying support, priceless motivation and guidance throughout the project duration. Moreover, I extend my sincere gratitude to all the lecturers and non-teaching staff of the Department of Electronics and Communication Engineering for their contribution towards the success of this work.

My friends also played a very important role therefore they cannot go unmentioned. I am deeply honoured and indebted to you all.

To my family, I know this quest of my academic journey has not been easy but I have always received your support and corporation to the entire process.

Thank you.

LIST OF ABBREVIATIONS

IT	information technology
&	and
Etc	ex-cetra
Co	company
Org	organisation
CS	computer science
Govt	government
Apps	applications
DNS	domain name system
BGP	border gateway protocol
ICT	information and communication technology
MOU	memorandum of understanding
HTML	hypertext mark-up language
LAN	Local Area Network
ARP	Address Resolution Protocol
MitM	Man in the Middle

LIST OF FIGURES

Figure no.	page no.
Figure 1: Evolution of Cyber Security Problems and Solutions	9
Figure 2: Some common victims attacked ad its percentage	10
Figure 3: Tactics used by hackers against victims	11
Figure 4: Who is behind the beaches	13
Figure 5: Vacancy of jobs for web security	14
Figure 6: Web Security Potential growth	15
Figure 7: Evolution of Cyber threats	19
Figure 8(a): Block diagram for connected devices	21
Figure 8(b): Block diagram for changing mac address	22
Figure 8(c): Block diagram for API and backend	22
Figure 9: Asking connected devices for their IP addresses	24
Figure 10: Changing the mac address (using python code)	24
Figure 11: Figure 9, result continued	25
Figure 12: command created for terminal	25
Figure 13: Figure 9, continued result of code	26
Figure 14: sending ARP packets	26
Figure 15: Results of ARP packets sent	27
Figure 16: Retrieval of list of connected devices	27
Figure 17: List of IP and MAC address of connected devices	28
Figure 18: IP address and MAC address retrieval	28
Figure 19: Running a Django project	29
Figure 20: Field in API to input name	29
Figure 21: Retrieval of MAC address using API	30
Figure 22: Field in API to fill with retrieved MAC address	30
Figure 23: Changing MAC address using API	31

Figure 24: Field to input retrieved IP address in API	31
Figure 25: Retrieval of IP address of connected devices	32
Figure 26: Checking for ARP attacks against the user	32
Figure 27: Stop check for ARP attacks	33

ABSTRACT

This project deals with the call of the hour web security/cybersecurity. As we know with the increase in technology and more frequent use of the internet there is always a threat to our information {personal, organizational, governmental} and then there comes the need for cybersecurity.

To make the information secure and encrypted we need to guard it and that is done by using secure gateways and fixing loopholes in the system or applications we are using.

In this project, we tried to create a secure network so that the information remains unharmed from threats by the intruders and the exchange of information goes on smoothly with anyone peeking into it.

We will be using different techniques in this project to overcome different vulnerabilities and fix loopholes.

v

CHAPTER 1

INTRODUCTION

1.1. General Background

The need of cyber or web security in today's world is at an enlarging pace. One needs to have a secure, protected and encrypt network for which different firms are contributing every possible thing they can do. They are encrypting their servers, creating new walls so that anyone can't penetrate them and are defining new boundaries by make their server more secure.

In this world information warfare is one such threat to every nation and every user. A nation can create a scenario in a country in which the people of that country starts devastating their own people. It can create a condition of cold war.

Other such stuff is hacking, in which any malicious personage can get access to every personal information you have in your hand{your mobile phone}, he can access every joyful moment or any sorriest moment you post online. There are certain threats to each and every one of us and it avoid such thing we need a secure and reliable web network.

In every corner of the globe we found new possibilities for new innovation and technologies and internet is reaching everywhere and with a reach of it reaches the hand of unwanted elements which are anonymous and can take away anything with being caught. To make web more secure from these people we need to make it more secure and less vulnerable.

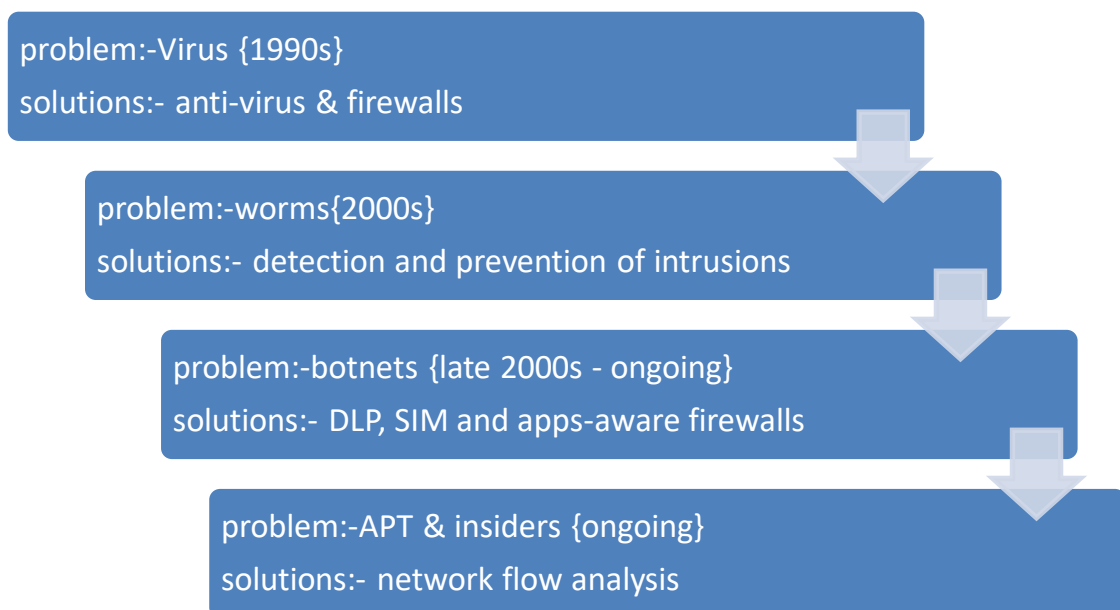


Figure 1:- evolution of cyber security {problems and solutions}

1.2. Problem Statement

The need to develop a secure web to surf is the need of the hour. Through innovative technologies and different virtual equipment just made for these purposes are very helpful for an encrypted system.

The system ought to be well enabled to detect any malicious movement in the system so that the personal stuff and even the crucial information may remain secure. These enacted tools which detect the malicious user help to intercept every request so that no one surpasses the system security and reaches at the depth of it.

There is much need to have such system and this need makes us more ambitious to create such cyber security network.

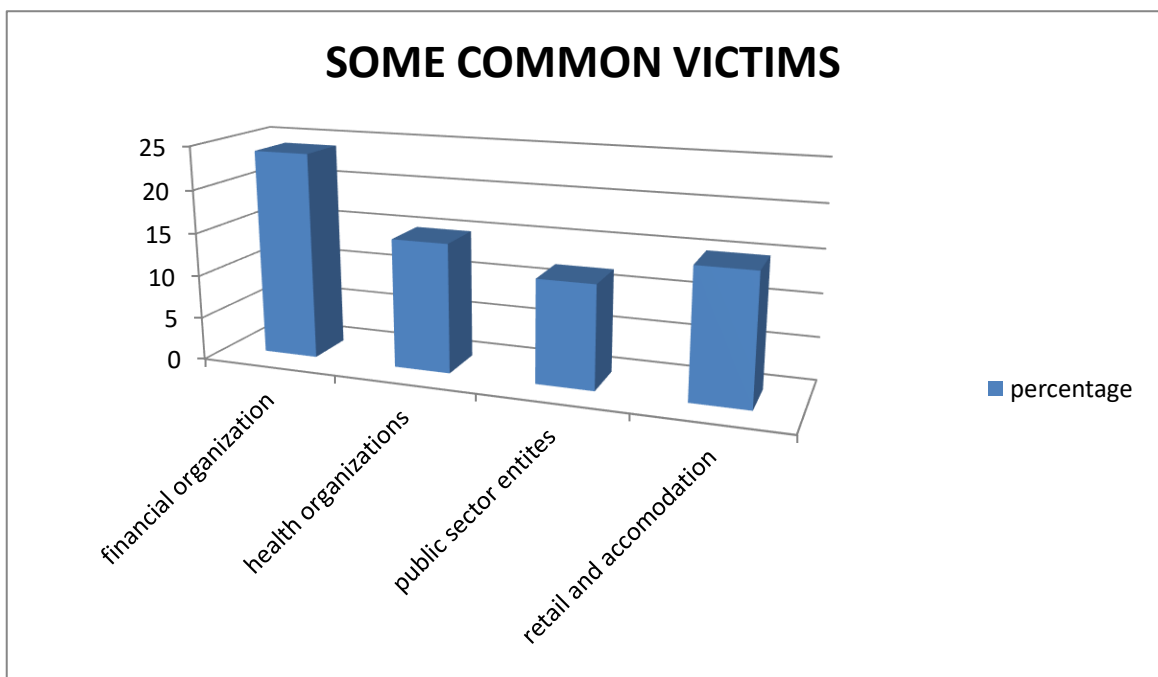


Figure 2:- some commons victims attacked and its percentage.

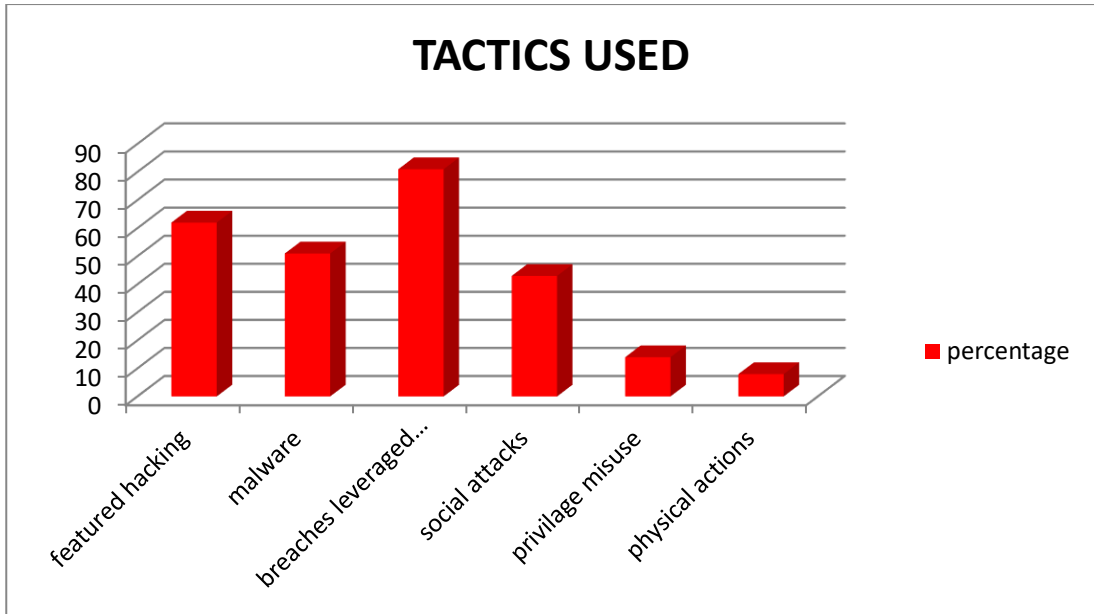


Figure 3:- tactics used by hackers against victims

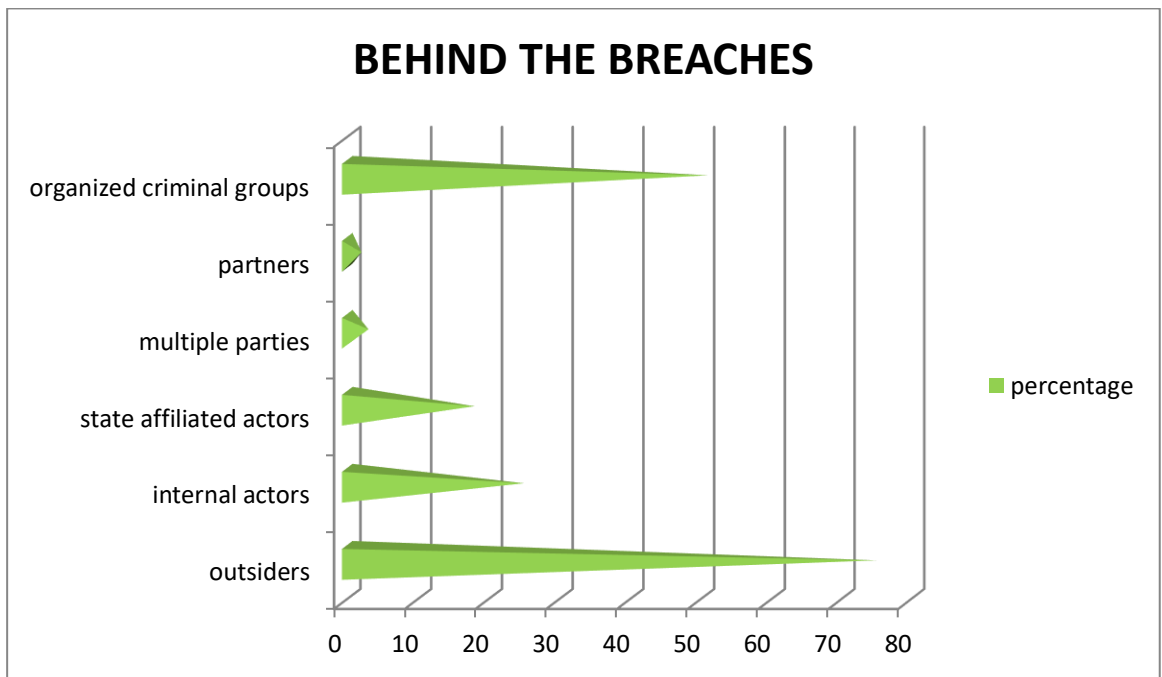


Figure 4:- who is behind the breaches

1.3.Objective

The main aim of this project is to focus on security and to design and develop a security system that includes features such as changing mac address, intercepting requests, etc.

This project also includes fixing of loopholes and vulnerabilities present in the system. People does not know what weak points are present in their system and this system will help them

recognise them and upgrade it eventually to surpass threats. These certain goals to achieve makes this project unique and motivates us to do so.

1.4.Scope of the project

Web Security can be referred to as the security of the information technology, protection of computers, networks and data from any unauthorised access change or destruction.

Now a days it can be seen that a significant threat to our privacies and personal data is increasing due to which a focus of attention towards protection of the data is quiet important.

A number of developed nations came to the conclusion that cyber-attacks, digital finger printing poses a greater threat to the nation’s security than terrorism itself.

Subsequently, there is a huge scope for cyber security professionals in different companies, nations, etc. Therefore, it can be said that national security and businesses in today’s world totally depends upon cyber security.

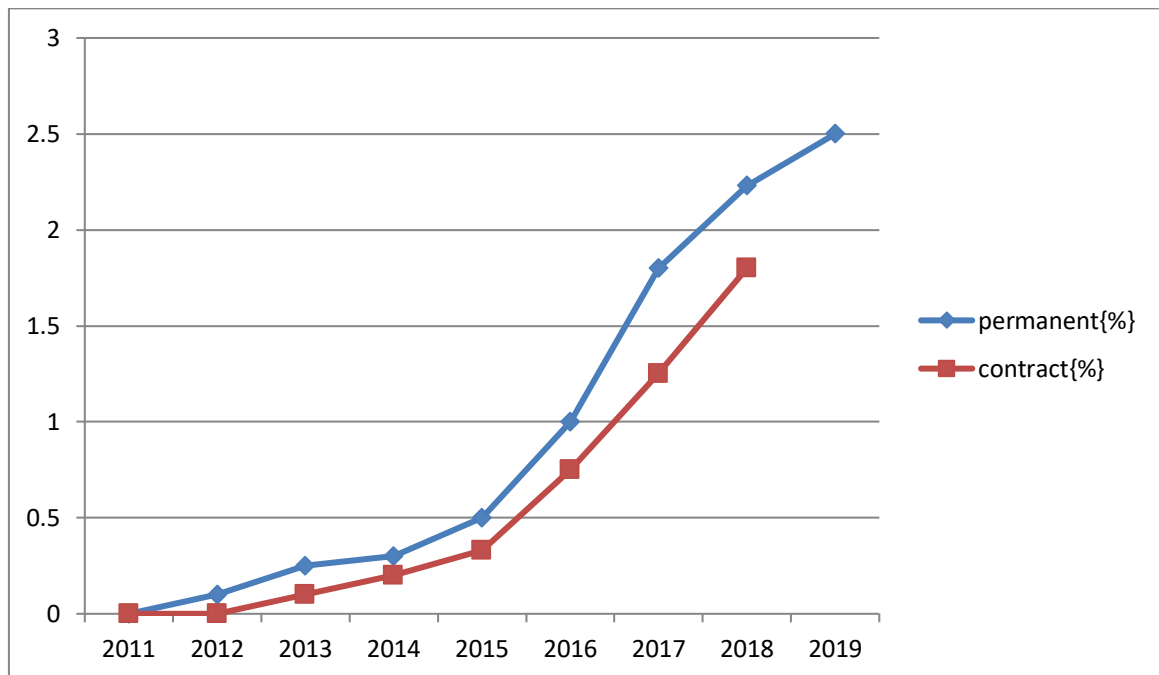


Figure 5:- vacancy of jobs for web security.

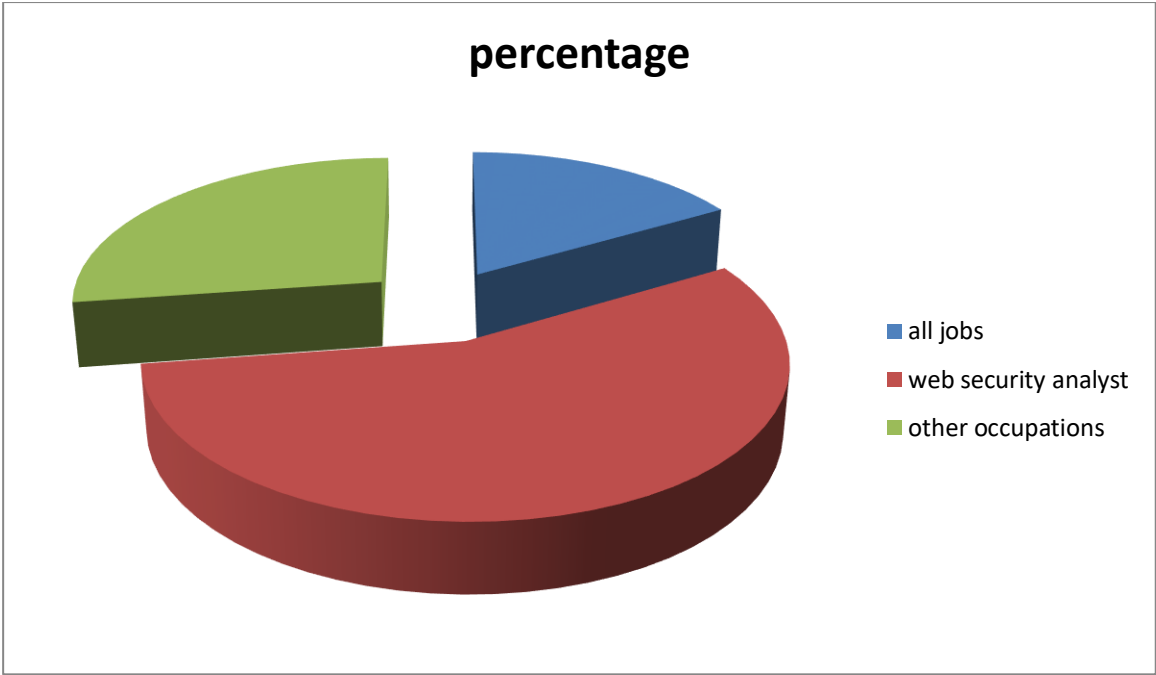


Figure 6:- web security potential growth {2012-2022}

CHAPTER 2

LITERATURE SURVEY

Web security as we know is not very latest to us rather its been rapidly evolved since past 50 years. As we know in the year 1968 a German spy in the Co., IBM got arrested by west German police for a case related to cyber espionage. Similarly, taking inspiration from a sci-fi movie named wargames some high school kids in 1983 got inside an unclassified military server naming themselves 414s. Nearly about 10 yrs ago, there was 1st real cyberwar which attacked Estonia & created a situational threat to national security. In today's world, every day we can find cases related to web security. Starting from spams, scams, frauds & even identity theft, to certain reports related to cyber warfare, cyber defence, cyber burglary, cyber espionage, etc. Now all this brings us the issue regarding how important cyber security is in this modern world. And eventually it becomes a very important and difficult task to coup up with cyber threats and increase web security as it connects every single individual and even country in certain situations. In the scenario where we have to define cyber security it will not only be done by us on the basics of our day to day life but also by govt., & by other prominent players. Due to the presence of politics in some kind of cyber threats is one of the reason why its been so unapproachable to issues related to web security. In this we are going to describe interrelated areas of web security which are web security and web securitization.

Dunn-Cavelty describes web security via a couple of ways i.e., regarding insecurities created by web & also by tech & non-tech practices to make web more encrypted. The above definition given by Dunn-Cavelty explains that web security is not just a tech issue related with CS, cryptography, IT, etc., but an entire large complex matter.

Dewar defines that the aim of web security is related to make use of tissue risk free from any type of harm. He also tells us about how nations tackle cyber threat and looks ahead for security of the web network and makes such reliable strategies and policies to eliminate malicious elements on web. He also describes 3 concepts of web security defence, that are as follows:-

- 2.1. Active Cyber Defence:- aims at elimination of web threats and agents which possess them in and out of the network.
- 2.2. Fortified Cyber Defence:- establishes a secure & encrypted environment.
- 2.3. Resilience Cyber Defence:- aims at providing continuity for the system.

Web security is such a matter which needs an extensive literature area to discuss the connection of web security and its development practices. The concept of cyber security discussed in the above survey describes how web security is looked as a national security issue.

Web security is a crucial concept & has a vital presence in the field of IT. Encryption & security of data and information have become real issue these days. There is an image which comes to our mind whenever we think about web security and that is cyber-crimes. In this modern world it is required by our govt., & even on individual level to fight these rapidly increasing criminal activities over internet. Apart from this it's a huge concern of web security to tackle these problems. The above literature survey aims at the elimination of problems related to web security on latest techs..

CHAPTER 3

WEB SECURITY

3.1. Web threats and their Sources

3.1.1. Threats:-

- Malware, viruses, worms, etc.
- Thieving crucial data, info and property.
- Apps {esp., 3rd party} which are associated with web attacks
- *Social engineering*- Provoke users to get access to malicious links and websites.
- *Hactivism*- web protests which are promoted by social and political sources.
- *Spear phishing*- spam mails, malicious texts and tweets.
- *Router security*- BGP hijacking
- *Denial of service*- restricting access to web pages and servers.
- DNS threats and attacks.
- Others.

3.1.2. Sources:-

- Hackers.
- Web Criminal orgs.
- Terrorists, criminals, etc.
- Countries.
- Others.

3.2. Evolution of web/cyber threat

There is more and more danger increasing on a daily basis regarding cyber security and these threats are on another pace and evaluates extensively. Web exploitation and maliciously targeted activities have become more sophisticated and more serious. There is a real time need to accelerate the counter attack on these cyber weapons which are the most important in this modern world.

Below is a figure showing the evolution of these threats since late 1970s:-



Figure 7:- Evolution of cyber threats

The upper 3 in the above figure are basic weapons and the lower 4 are advance weapons.

3.3. Steps taken for web security

3.3.1. Network Security

Make the network more secure to fight against threats and attacks possess by both external and internal sources. Maintain network premises. Separate unauthenticated access and suspicious contents. Watch out and run test on security controls.

3.3.2. Protection from Malwares

Provide required and efficient policies and build anti-malware defences which are relevantly applicable to all business fields. Scan each part of organisation for malware.

3.3.3. Monitoring

Built and maintain a monitoring system including strategies to detect several malicious content present & provide support policy. Make more frequent watches over all ICT systems & networks. Check out activity logs more often so that if any attack is there then you are ready before hand.

3.3.4. Incident management

Built incident response & manage recover capabilities in time severe diasaster. Provide and allow certain test for incident management plans. Make this management team as such that they can take control over the situation when attack over and finally report these criminal attacks to law enforcements.

3.3.5. knowledge-full & aware users

Provide such knowledge to the users & reliable policies for the use of org's system. Build such a trained staff for making the users learn new essential stuffs. Manage awareness programme to tackle web threats.

3.3.6. Mobile work & home work

Make such mobile manageable policies which in staff training & even helps users. Make an encrypted base line for each every device. Secure information in every aspect.

3.3.7. Encrypt config

Create patches for loopholes & apply & make sure it secure all config in information and communication technology system. Make inventory for system and build a baseline for information & communication system.

3.3.8. Controls for removable media content

Make MOUs for access control over removable media contents. Make a restrict media type & its use. Make a scanning system for media to detect malware before its import on Co. network.

3.3.9. Maintaining Privileges for users

Build a system to process and manage accounts and special privileged ones. Restrict privileges of user & watch their activities.

3.3.10. Info risk management

Built a governed system to maintain & determine risk factors. Manage & monitor the engagement of board members with web threats. Provide help policies for info risk management.

3.4. Cyber Security

Cyber security which is also known as web security makes an image in our mind that it has something to do with internet. And as we know it's been so crucially important to understand the concept of it and how to tackle the attacks it possess.

Here in this project we have tried something similar. We have firstly tried to take over the mac address of the system and then create a frame for take and broadcasting of devices present in the system. Then, we asked every device connected in the system to give a particular input and collect those input plus the mac addresses from the devices they are coming from. Then, we created a list for every answered and unanswered calls and we displayed the unwanted connections. We have used soft wares such as kali and pycharm.

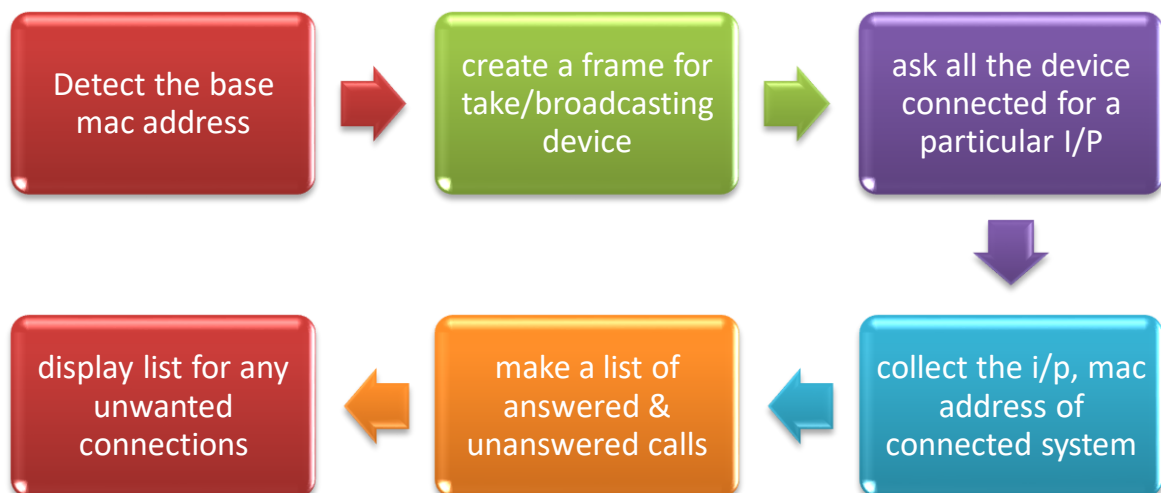


Figure 8(a): Block diagram for connected devices

Here, as per the below block diagram shows we have tried to take input for grabbing the mac address and then stored those values and made them visible. Then, tried to pull down the devices and passed the values and turned the device on.

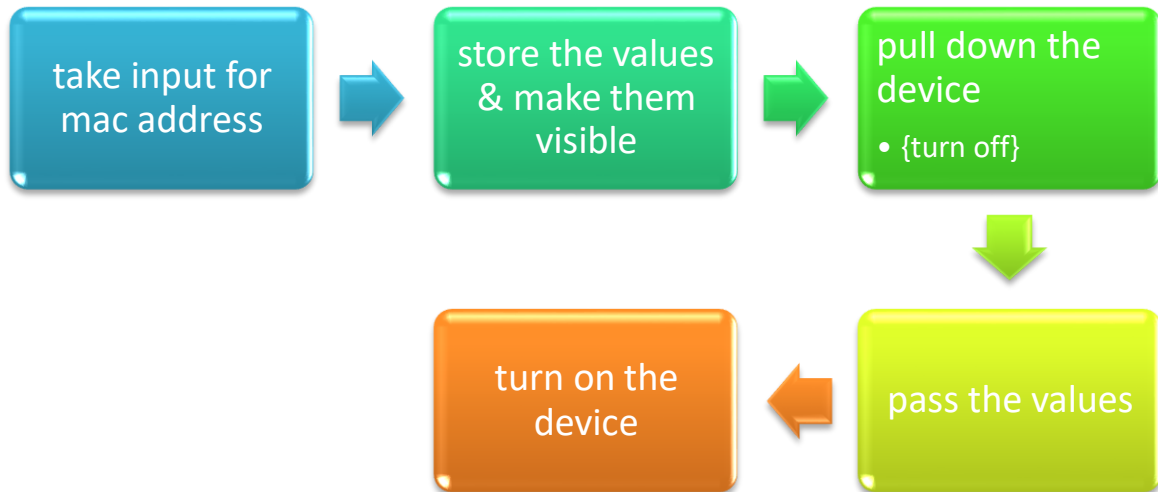


Figure 8(b):- Block diagram for changing mac address

Here, we have tried to create a frond end code using HTML and applied certain meta tags and anchors in its body and provided a proper path to all the tags applied. Then, created a script code and merges into a single file and ensured that the path always remains correct and sync.

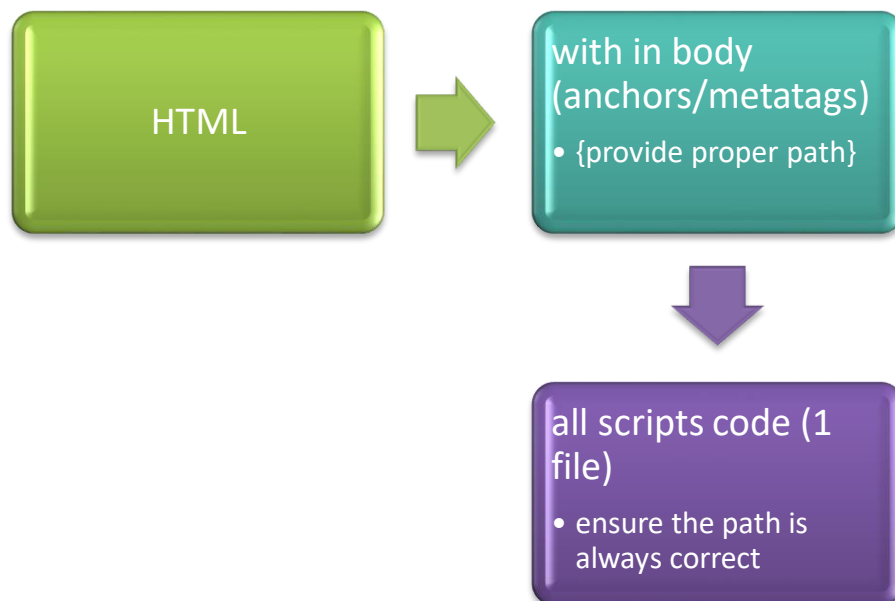


Figure 8©:- Block diagram for API and backend.

3.5. ARP Spoofing

It is certain kind of malicious injection that is sent to the user by some dangerous actor in which there is a falsified ARP over a LAN. It allows the hacker to link his MAC address with the IP address of the host network. If somehow the attacker gets this done then he will be able to watch over the host data being received on the IP address. This Technique of attack can allow the hacker to intercept, upgrade/ stop the data in-transit.

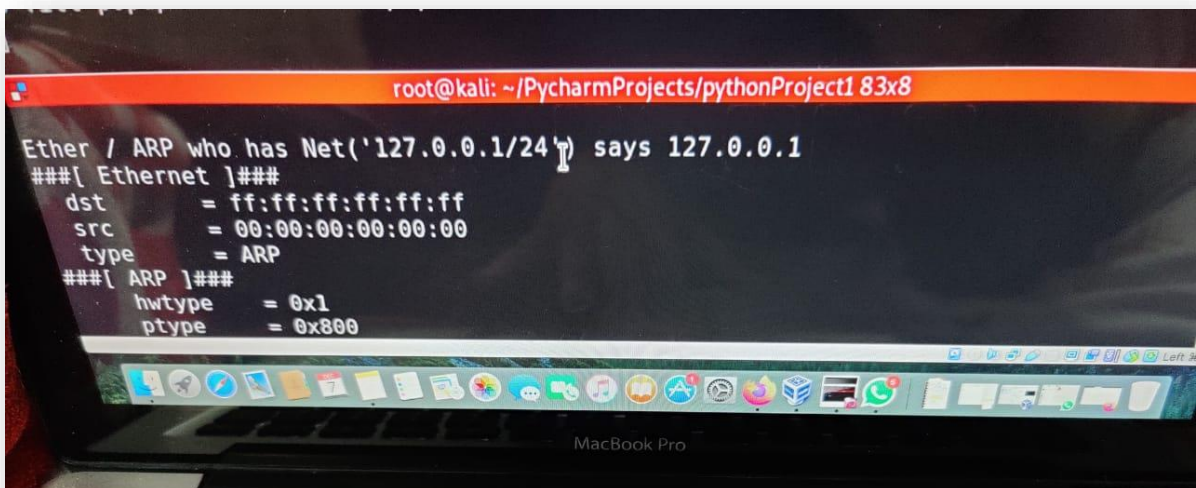
This malicious activity only occurs on LAN which uses ARP and is used to steal certain sensitive information that is crucial for any enterprise. It is a kind of MitM attack.

ARP spoofing is also known as ARP poisoning.

CHAPTER 4

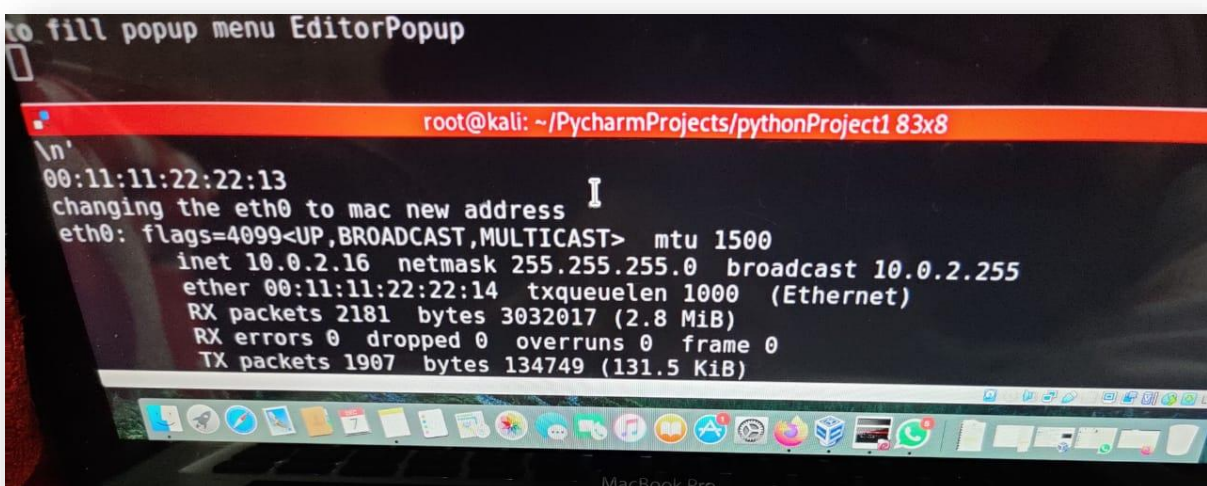
RESULT AND DISCUSSION

Here, in this project we have started create front end using HTML and also making backend and will upgrade it as per the upcoming events. Here, in this project you will find obtaining mac addresses and altering them. Here you will also find extensions and their uses given to the users to obtain certain protecting tools which users sometimes don't know about. Here, the user will able to find all the systems connected in the network and will able to know about their mac addresses.



```
root@kali: ~/PycharmProjects/pythonProject1 83x8
Ether / ARP who has Net('127.0.0.1/24') says 127.0.0.1
###[ Ethernet ]###
dst      = ff:ff:ff:ff:ff:ff
src      = 00:00:00:00:00:00
type     = ARP
###[ ARP ]###
hwtype   = 0x1
ptype    = 0x800
```

Figure 9:- Asking connected devices for their IP addresses



```
to fill popup menu EditorPopup
root@kali: ~/PycharmProjects/pythonProject1 83x8
\n'
00:11:11:22:22:13
changing the eth0 to mac new address
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
inet 10.0.2.16 netmask 255.255.255.0 broadcast 10.0.2.255
ether 00:11:11:22:22:14 txqueuelen 1000 (Ethernet)
RX packets 2181 bytes 3032017 (2.8 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1907 bytes 134749 (131.5 KiB)
```

Figure 10:- Changing the mac address (using python code)

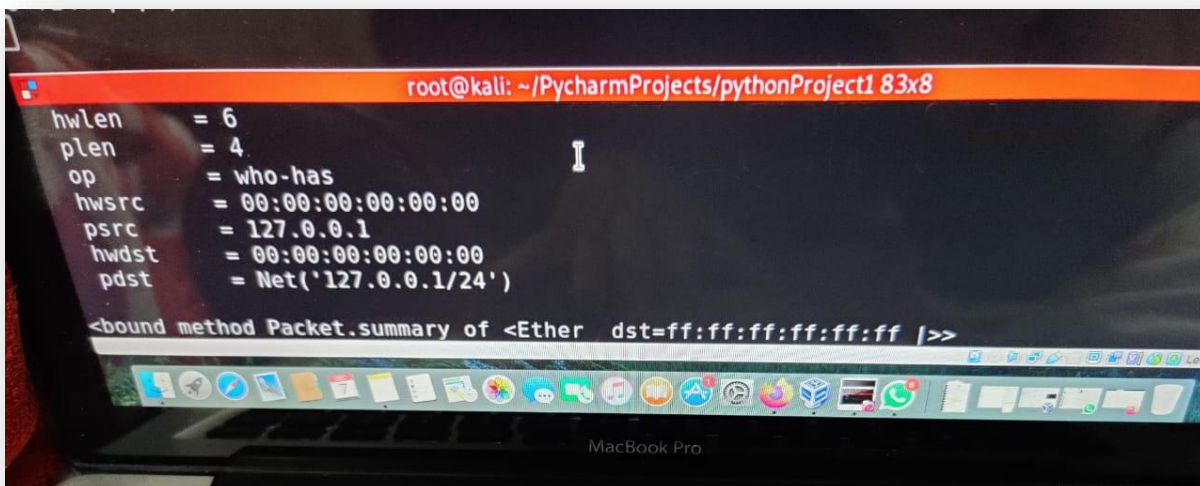


Figure 11:- Figure 9 ,result continued

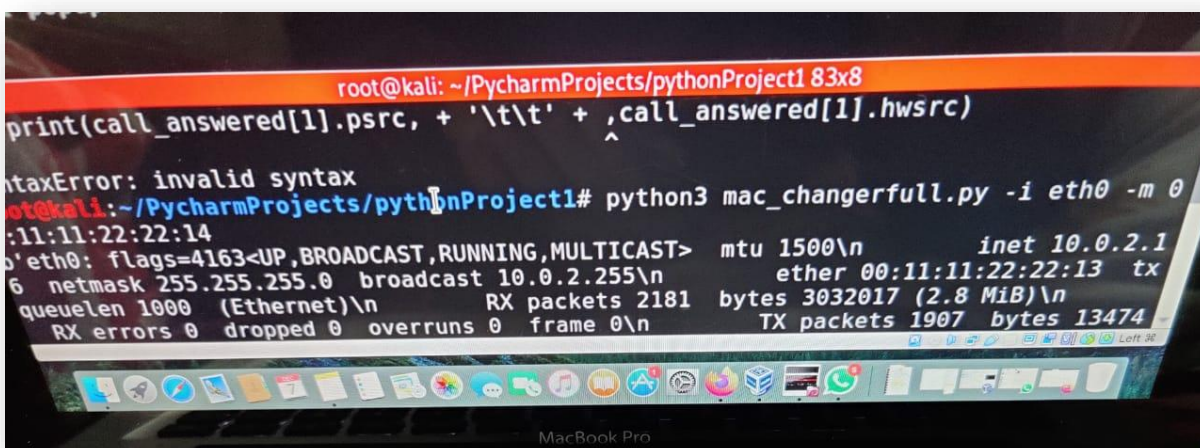


Figure 12:- command created for terminal

```
root@kali: ~/PycharmProjects/pythonProject1 83x8
pdst = Net('127.0.0.1/24')
<bound method Packet.summary of <Ether dst=ff:ff:ff:ff:ff:ff |>>
###[ Ethernet ]###
dst = ff:ff:ff:ff:ff:ff
src = 00:11:11:22:22:14
type = 0x9000
Ether / ARP who has Net('127.0.0.1/24') says 127.0.0.1
```

Figure 13:- Figure 9,continued result of code

```
root@kali: ~/PycharmProjects/pythonProject1 83x8
###[ ARP ]###
hwtype = 0x1
ptype = 0x800
hwlen = 6
plen = 4
op = who-has
hwsrc = 00:00:00:00:00:00
psrc = 127.0.0.1
hwdst = 00:00:00:00:00:00
```

Figure 14:- sending ARP packets

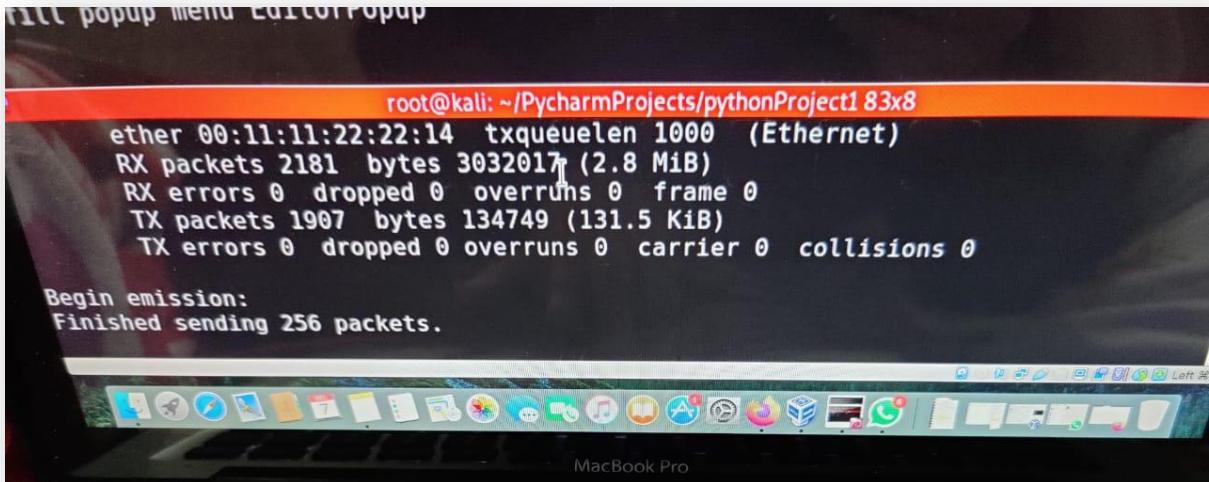


Figure 15:- Results of ARP packets sent

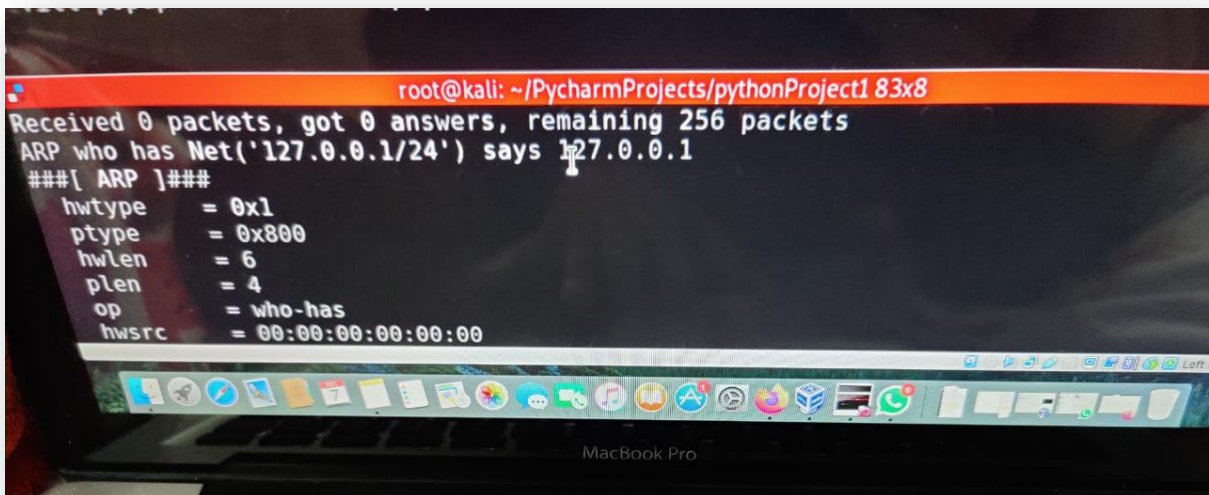


Figure 16:- Retrieval of list of connected devices

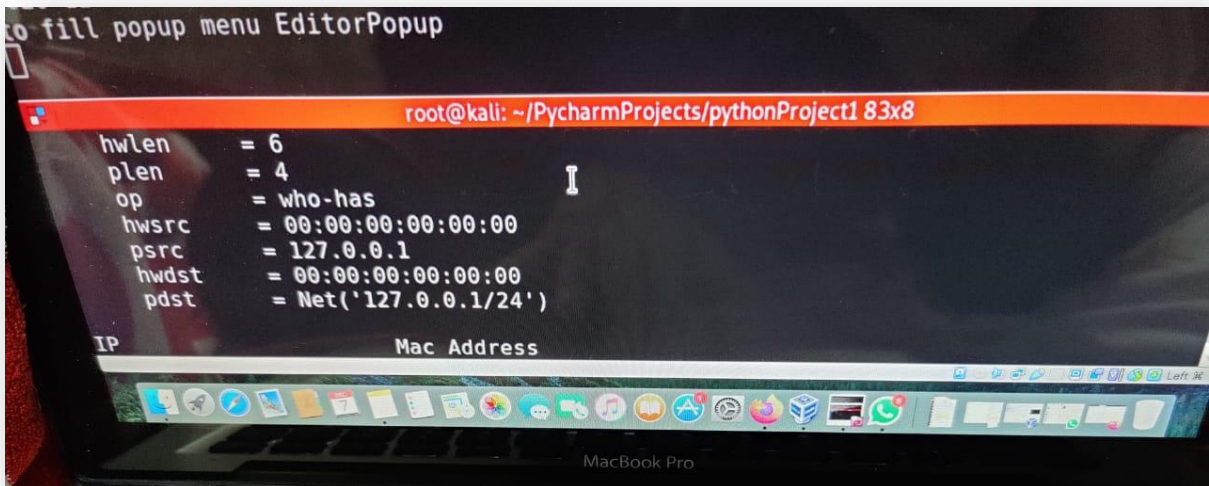


Figure 17:- List of IP and MAC address of connected devices

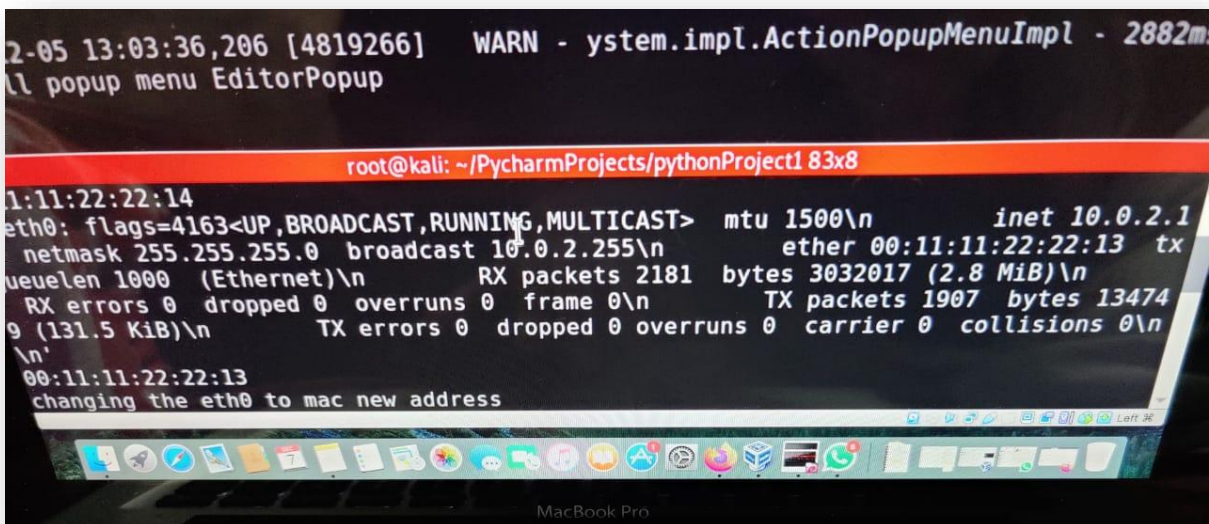


Figure 18:- IP address and mac address retrieval

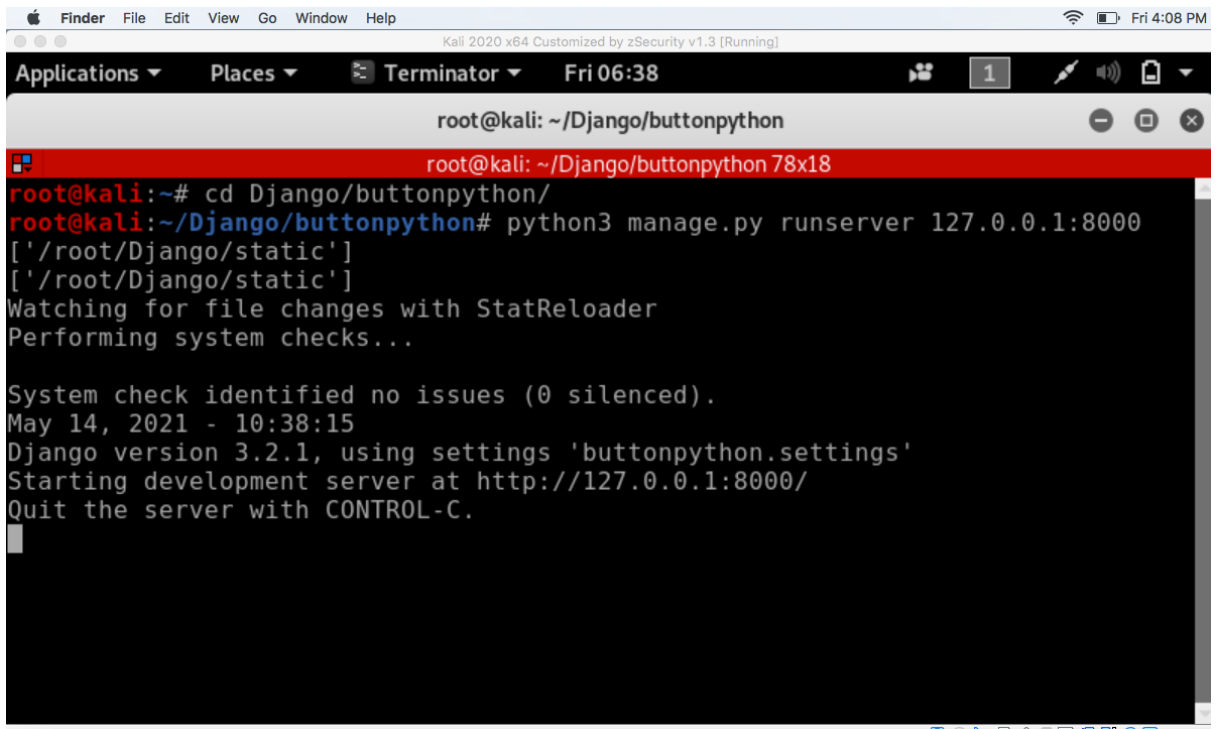


Figure 19:- Running a Django project

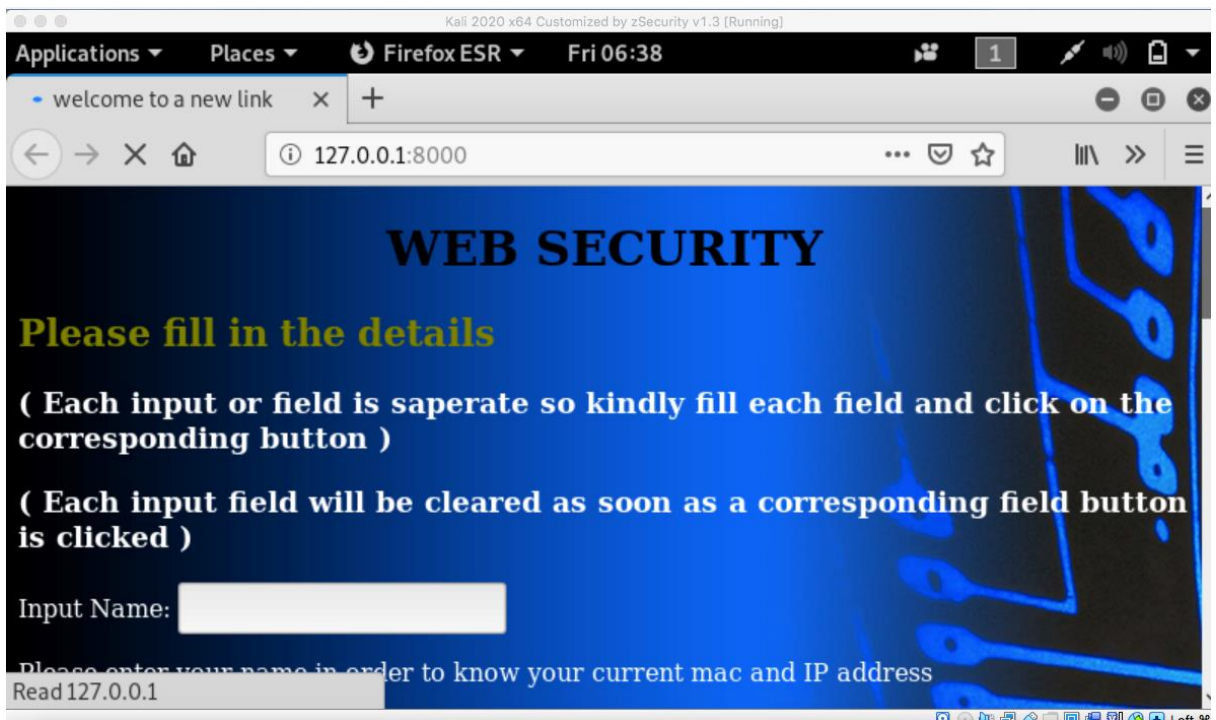


Figure 20:- Field in API to input name

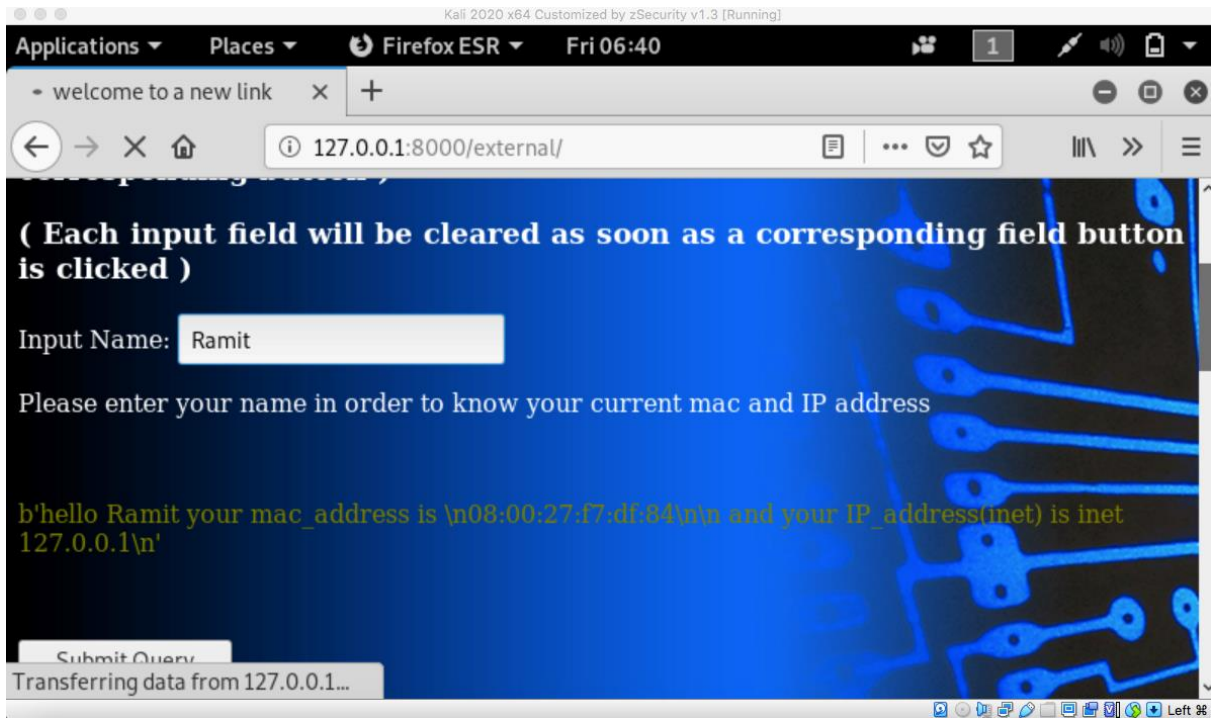


Figure 21:- Retrieval of MAC address using API

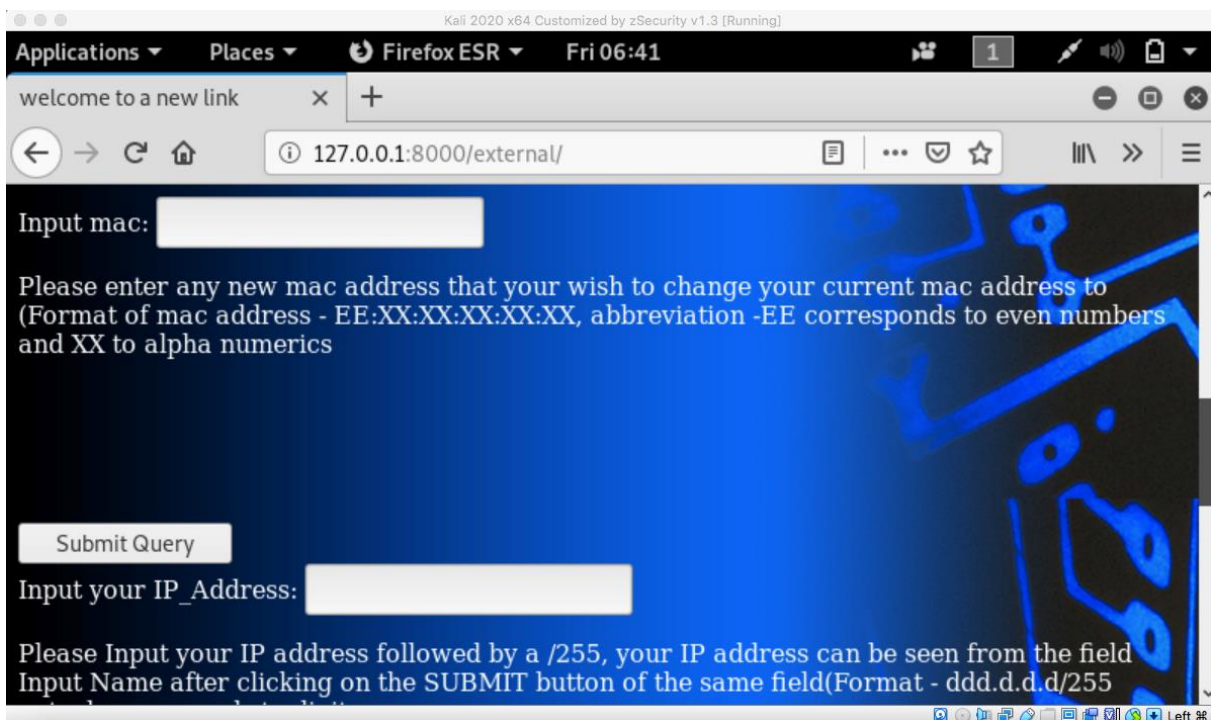


Figure 22:- Field in API to fill with retrieved MAC address



Figure 23:- Change of MAC address using API

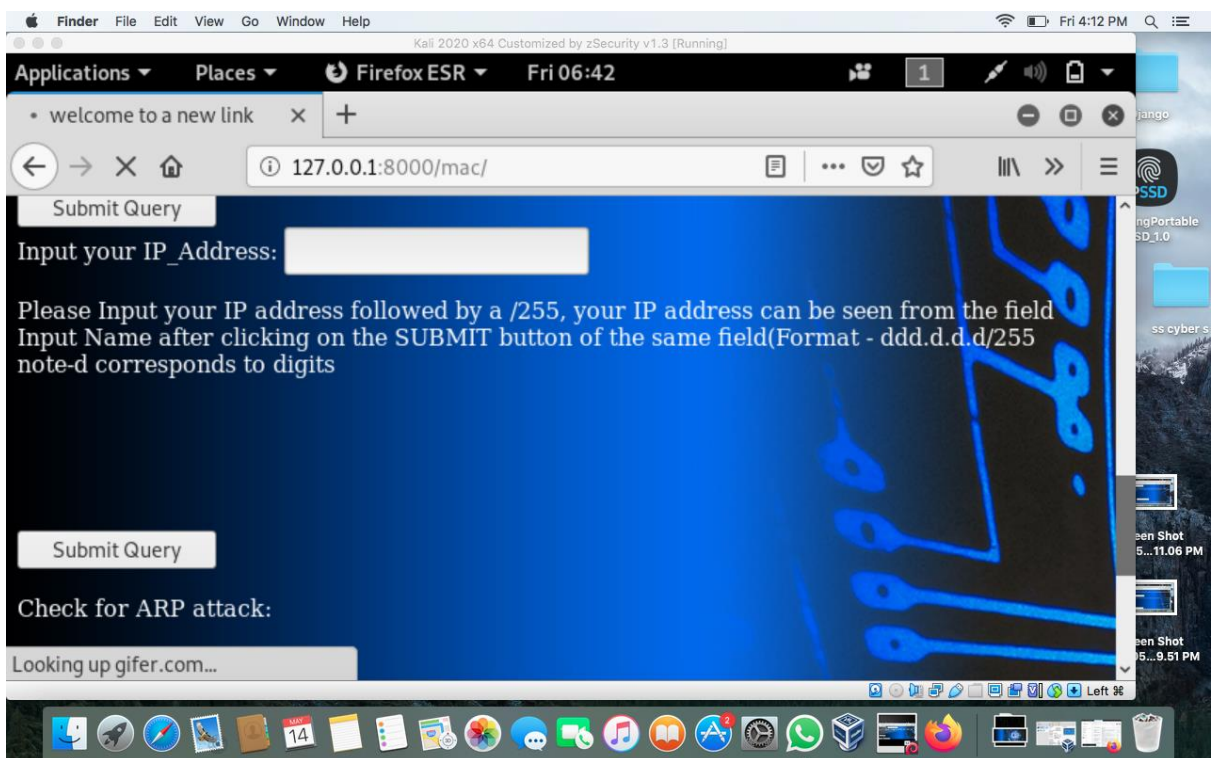


Figure 24:- Field to input retrieved IP address in API



Figure 25:- Retrieval of IP address of connected devices

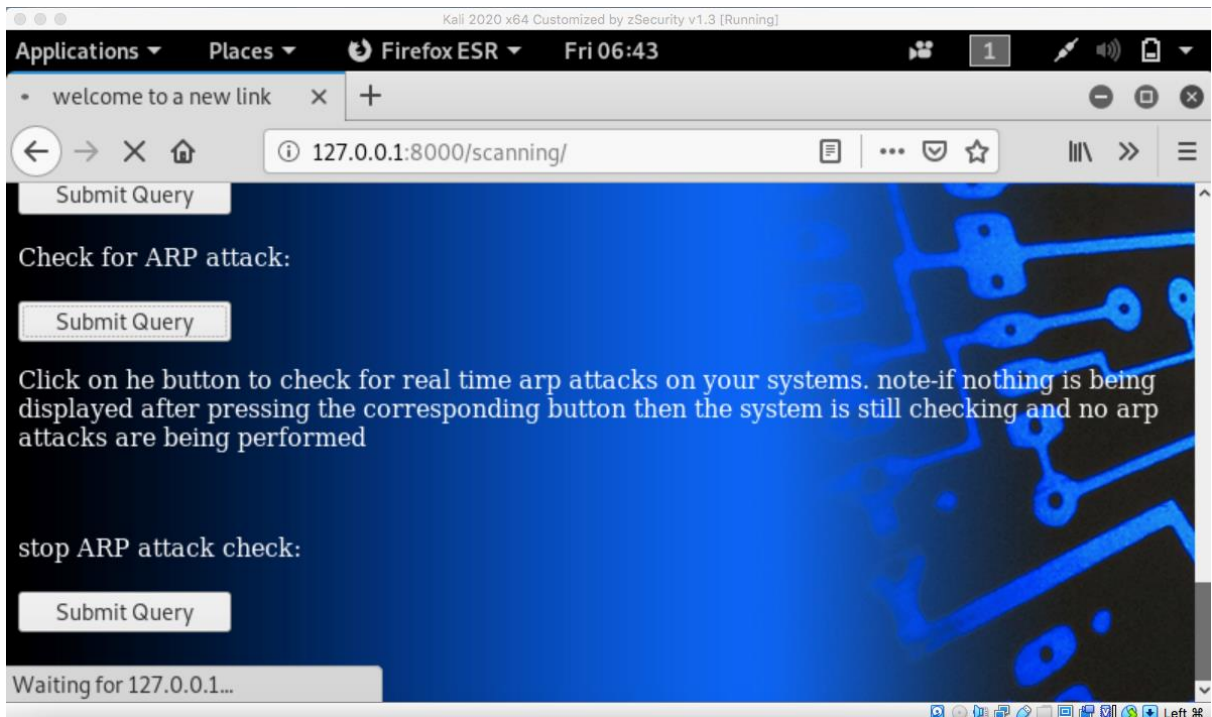


Figure 26:- Checking for ARP attacks against the user

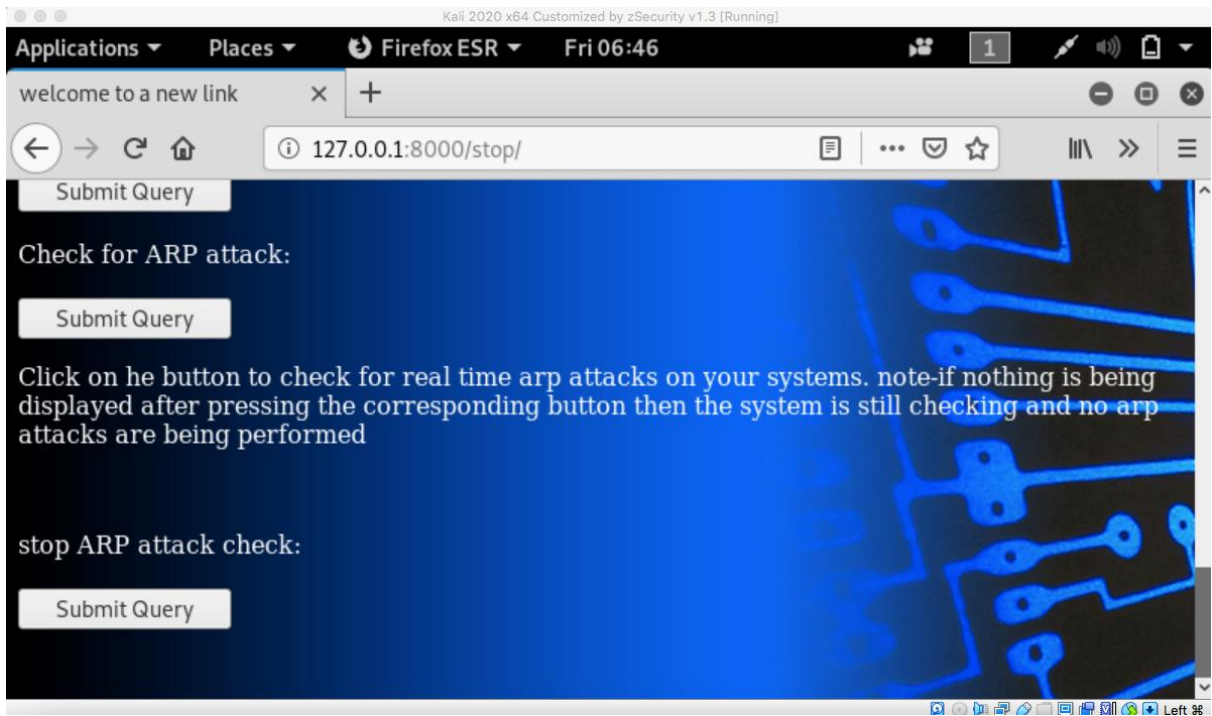


Figure 27:-Stop checking for ARP attacks

CONCLUSION & FUTURE WORK

Conclusion:

In this modern world full of digital threats {esp., cyber threats}, there is a need for upgrading our shielding and tackling systems against them.

Here in this project we have tried to get the mac address of the system and make it anonymous so that its hard for any malicious user to get the track of it and harm the system. Backend as well as the frontend is been developed in this project.

Work Done Till Now:-

Backend and Frontend is created and further updating it. HTML codes as well as Python codes are being combined with each other.

ARP Spoofing/ Poisoning is used in a reversible manner so as to create a more secure network. In this, a virtual system is being created on our own system that helps to make a link between a host computer and a router so that the system is more secure from any direct attack.

Future work:

Further improvement in backend and frontend along with more added features in backend.

Several codes for downloading tools for network security to be implemented using Django, implementation of previously made codes to make a more safe system for the user, using ARP spoofing.

REFERENCES

- [1] Dewar, R. 2014. 'the Triptych of Cyber Security: A Classification of Active Cyber Defense'. 6th International Conference on Cyber Security
- [2] Dunn-Cavelty, M. 2010. 'Cyber Security' in A. Collins, Contemporary Security Studies. Oxford: OUP
- [3] Dunn-Cavelty, M. 2013. From Cyber-Bombs to Political fallout: threat Representations with an impact in Cyber-Security Discourse. International Studies Review, 15, pp. 105-122
- [4] Hansen, L. and Niessanbaum, H. 2009. Digital Disaster, Cyber Security, and the Copenhagen School. International Studies Quarterly, 53, pp. 1155-1175
- [5] McLean, S. 2013. Beware the Botnets: Cyber Security is a Board Level Issue. Intellectual Property & Technology Law Journal, 25 (12), pp. 22-27
- [6] Warner, M. 2012. Cybersecurity: A Pre-history. Intelligence and National Security, 27 (5), pp. 781-799
- [7] Vacca, JR. 2013. Cyber Security and IT infrastructure protection. Waltham: Steven Elliot
- [8] A Sophos Article 04.12v1.dNA, eight trends changing network security by James Lyne.
- [9] Cyber Security: Understanding Cyber Crimes- Sunit Belapure Nina Godbole
- [10] Computer Security Practices in Non Profit Organisations – A NetAction Report by Audrie Krause.

APPENDIX

Code 1:-

```
#!/usr/bin/env python

import subprocess
import optparse
import scapy.all as scapy
import re

def mac_changer(interface,new_address):
    print("changing the " + interface + " to mac new address")
    subprocess.call("ifconfig " + interface + " down ", shell=True)
    subprocess.call("ifconfig " + interface + " hw ether " + new_address,
shell=True)
    subprocess.call("ifconfig " + interface + " up ", shell=True)
    subprocess.call('ifconfig eth0',shell=True)
def downloadmoz():
    subprocess.call("sudo apt update")
    subprocess.call("sudo apt upgrade")
    subprocess.call("sudo apt install sudo apt install firefox")
def downloadburp():
    subprocess.call("sudo apt update && sudo apt install")
    subprocess.call("sudo apt-get install burpsuite")
def help():
    phasor = optparse.OptionParser()
    phasor.add_option('-i', '--interface', dest='interface')
    phasor.add_option('-m', '--mac_address', dest='new_address')
    (options, argument) = phasor.parse_args()
    ## return phasor.parse_args()
    if not options.interface:
        phasor.error("no interface provided", 'please provide a valid
interface')
    elif not options.new_address:
        phasor.error('no mac address provided', 'please provide a mac
address')
    return phasor.parse_args()
(options,arguments)=help()
ifconfig_results=subprocess.check_output(['ifconfig', options.interface])
print(ifconfig_results)
mac_add = re.search(r"\w/\w:\w/\w:\w/\w:\w:\w/\w:\w:\w/\w/\w",
str(ifconfig_results))
print(mac_add.group(0))
def scanning(ip):
    scapy.arping(ip)
    result_arp=scapy.ARP(pdst=ip)
    print(result_arp.summary())
    result_arp.show()
    frame_broadcast = scapy.Ether(dst='ff:ff:ff:ff:ff:ff')
    print(frame_broadcast.summary)
    frame_broadcast.show()
    result_arp_frame_broadcast=frame_broadcast/result_arp
    print(result_arp_frame_broadcast.summary())
    result_arp_frame_broadcast.show()
    call_answered = scapy.srp(result_arp_frame_broadcast,timeout
=1,verbose=False)[0]
    print('IP\t\t\tMac Address')
    for calls in call_answered:
```

```

print(call_answered[1].psrc + '\t\t' + call_answered[1].hwsrc)
#print (call_answered[1].hwsrc)

#print('////////////////////////////////////')
//')
    #print(call_unanswered.summary())

#device = options.interface
#new_address = options.new_address
#device = input("enter the interface name")
#new_address = input('enter the new mac address')
mac_changer(options.interface,options.new_address)
scanning("127.0.0.1/24")
print('do you want to install mozilla firefox(stable)?')
a = input("1/0 \n")
if (a == 1):
    downloadmoz()
else :
    print("mozilla not installed")
b=input('do you want to download burpsuite?1/0 \n')

if (b==1):
    downloadburp()
else:
    print(' burpsuite not installed')

```

Code 2:

```

import subprocess
import sys
import re

#ifconfig_results=subprocess.check_output(['ifconfig'])
#print(ifconfig_results)
#print(mac_add.group(0))
    # subprocess.call("ifconfig " + address + " hw ether " + new_address,
shell=True)
    # subprocess.call("ifconfig " + address + " up ", shell=True)
    #subprocess.call('ifconfig eth0',shell=True)
#mac_changer()
print("hello")
mac= "%s"%(sys.argv[1])
#print("hello2")
#k=ifconfig_results1.stdout
#print("hello3")
#print(mac_add1.group(0))
#ip_add = re.search(r"inet\s\d\d\d.\d.\d.\d", str(ifconfig_results1))
#print("\n and your IP_address(inet) is " + ip_add.group(0))

f=subprocess.run("ifconfig eth0 down " ,

```

```

shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal_newlines=True
)
#ifconfig_results2=subprocess.check_output(["ifconfig eth0
down"],shell=True,universal_newlines=True)
#k=ifconfig_results2.stdout
subprocess.run("ifconfig eth0 hw ether " + mac,
shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal_newlines=True
)
subprocess.run("ifconfig eth0 up ",
shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal_newlines=True
)
ifconfig_results1=subprocess.check_output(['ifconfig'])
mac_add1 = re.search(r"\w\w:\w\w:\w\w:\w\w:\w\w:\w\w", str(ifconfig_results1))
b=str(mac_add1)
print("your mac_address has been changed to" + b)

```

Code 3:

```

import subprocess
import sys
#import optparse
#import scapy.all as scapy
#import re
#import requests
import re
import datetime#import urllib.parse as urlparse
#from bs4 import BeautifulSoup
#def mac_changer(address):
date=datetime.datetime.now()
address = "hello %s your mac_address is " % (sys.argv[1])
e=subprocess.run("ifconfig ",
shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal_newlines=True
)
#out=e.stdout
print(address)
#print(out)
ifconfig_results=subprocess.check_output(['ifconfig'])
#print(ifconfig_results)
mac_add = re.search(r"\w\w:\w\w:\w\w:\w\w:\w\w:\w\w", str(ifconfig_results))
print(mac_add.group(0))
ip_add = re.search(r"inet\s\d\d\d.\d.\d.\d", str(ifconfig_results))
print("\n and your IP_address(inet) is " + ip_add.group(0))

```



```

    # subprocess.call("ifconfig " + address + " hw ether " + new_address,
shell=True)
    # subprocess.call("ifconfig " + address + " up ", shell=True)
    #subprocess.call('ifconfig eth0',shell=True)
#mac_changer()
#mac= "%s"%sys.argv
#f=subprocess.run("ifconfig "+ mac +"down" ,
shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal_newlines=True
)
#ifconfig_results1=subprocess.check_output(['ifconfig' + mac +'down'])
#mac_add1 = re.search(r"\w\w:\w\w:\w\w:\w\w:\w\w", str(ifconfig_results))
#print(mac_add1.group(0))

```

Code 4:

```

from django.contrib import admin
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', views.button),
    url(r'^output',views.output,name="script"),
    url(r'^external', views.external),
    url(r'^mac', views.mac),
    url(r'^scanning', views.scanning),
    url(r'^ip_address', views.ip_address,name='ip_address'),
    url(r'^detector', views.detector),
    url(r'^stop', views.stop),
]

```

Code 5:

```

import scapy.all as scapy
def mac(ip):
    arp_requests = scapy.ARP(pdst=ip)
    fake_add=scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
    broadcast=fake_add/arp_requests
    answers = scapy.srp(broadcast, timeout=1, verbose=False)[0]
    return answers[0][1].hwsrc
def sniffing(interface):
    scapy.sniff(iface=interface, store=False,prn=sniffed_packets)
def sniffed_packets(packet):
    if packet.haslayer(scapy.ARP) and packet[scapy.ARP].op ==2:
        try:

```

```

        actual_mac_address=mac(packet[scapy.ARP].psrc)
        response=packet[scapy.ARP].hwsrc

    if actual_mac_address != response:
        print("you are under an arp soofing attack")

    else :
        print("no ARP attack")
except IndexError:
    pass
sniffing("eth0")

```

Code 5:

```

import subprocess
def downloadmoz():
    output="hi %s %s"
    print(output)
    a=input("do you want to download firefox?True/False ")
    if a == True:
        p=subprocess.run("sudo apt-key adv --keyserver keyserver.ubuntu.com --
recv-keys A6DCF7707EBC211F", shell = True,stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
        #p.kill()
        q=subprocess.run("sudo apt install
firefox",shell=True,stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        #q.kill()
        ## p= subprocess.Popen(*popenargs,**kwargs)
        #return p.wait()
        #return q.wait()
    else :
        print("mozilla not installed")
def burp():
    j= input("do you want to install burpsuite?True/False")
    if j==True:
        g=subprocess.run('sudo apt-get install openjdk-8-
jre',shell=True,stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        #g.kill()
    else:
        print('burp not installed')

burp()

```

```
#subprocess.Popen('clear',shell=True)
downloadmoz()
```

Code 6:

```
import subprocess
import re

subprocess.run("ifconfig ",
shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal_newlines=True
)
#out=e.stdout
#print(out)
ifconfig_results=subprocess.check_output(['ifconfig'])
#print(ifconfig_results)
ip_add = re.search(r"inet\s\d\d\d.\d.\d.\d", str(ifconfig_results))
print("Your IP_address(inet) is " + ip_add.group(0))
```

Code 7:

```
import scapy.all as scapy
import sys
ip_address = "%s" % (sys.argv[1])

def scanning(ip):
    scapy.arping(ip)
    result_arp = scapy.ARP(pdst=ip)
    #print(result_arp.summary())
    result_arp.show()
    frame_broadcast = scapy.Ether(dst='ff:ff:ff:ff:ff:ff')
    #print(frame_broadcast.summary)
    frame_broadcast.show()
    result_arp_frame_broadcast = frame_broadcast / result_arp
    #print(result_arp_frame_broadcast.summary())
    result_arp_frame_broadcast.show()
    call_answered = scapy.srp(result_arp_frame_broadcast, timeout=1,
verbose=False)[0]
    print('IP\t\t\tMac Address')
    for calls in call_answered:
        print(call_answered[1].psrc + '\t\t' + call_answered[1].hwsrc)
        # print(call_answered[1].hwsrc)
        #
```

```

print('////////////////////////////////////')
    # print(call_unanswered.summary())

# device = options.interface
# new_address = options.new_address
# device = input("enter the interface name")
# new_address = input('enter the new mac address')
# mac_changer(options.interface,options.new_address)
scanning(ip_address)

```

Code 8:

```

import subprocess

subprocess.call('kill
all',shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,universal_new
lines=True)

```

Code 9:

```

<!DOCTYPE html>
<html>
{% load static %}
<link rel="stylesheet" href="{% static '8oXf.gif'%}">
<head>
<title>welcome to a new link</title>
<style type="text/css">
body {
    background-image: url("https://www.fscj.edu/images/default-source/workforce-
education/progweb_1920x720.jpg?sfvrsn=f74686d5_0")
    background-color: Black;
.p1 {
    font-family: "Times New Roman", Times, serif;
}
h1 {
    color: white;
    text-align: center;
}

input {
color: white;

```

```

}
#example {
color: #0000ff;
}
form {
background-color: Black;
}

</style>

</head>
<body id="bg" style="background-image: url('{% static 'images/new.jpg' %}');">
<div style="background-image: url('https://gifer.com/en/8oXf.gif');"></div>
<h1 class="p1" style="color:black" align = "center" > WEB SECURITY</h1>
<h2 style="color:olive">Please fill in the details</h2>
<h3 style="color:white">( Each input or field is saperate so kindly fill each
field and click on the corresponding button )</h3>
<h3 style="color:white">( Each input field will be cleared as soon as a
corresponding field button is clicked )</h3>
<form action="/external/" method="post">
{% csrf_token %}
<lable style="color:white">Input Name:</lable>
<input type="text" name="param" required> <p style="color:white"> Please enter
your name in order to know your current mac and IP address<br></p>
<p style="color:olive">{{data_external}}<br><br>
{{data1}}
</p>
<br><br>
<input type="submit" name="submit1",value='external script'>
<br><br>
</form>
<form action="/mac/" method="post">
{% csrf_token %}
<lable style="color:white">Input mac:</lable>
<input type="text" name="param1" required> <p style="color:white"> Please enter
any new mac address that your wish to change your current mac address to (Format
of mac address - EE:XX:XX:XX:XX:XX, abbreviation -EE corresponds to even numbers
and XX to alpha numerics<br></p>
<p style="color:olive">{{data_mac}}<br><br>
{{data2}}

```

```

</p>
<br><br>

<input type="submit" name="submit",value='Register! '>
</form>

<form action="/scanning/" method="post">
{% csrf_token %}
<lable style="color:white">Input your IP_Address:</lable>
<input type="text" name="param2" required> <p style="color:white"> Please Input
your IP address followed by a /255, your IP address can be seen from the field
Input Name after clicking on the SUBMIT button of the same field(Format -
ddd.d.d.d/255 note-d corresponds to digits <br></p>
<p style="color:olive">{{data_scanning}}<br><br>
{{data3}}
</p>
<br><br>
<input type="submit" name="submit1",value='external script'>
<br><br>
</form>
<form action="/detector/" method="post">
{% csrf_token %}
<lable style="color:white">Check for ARP attack:</lable>
<br><br>
<input type="submit" name="submit1",value='external script'><p
style="color:white"> Click on he button to check for real time arp attacks on
your systems. note-if nothing is being displayed after pressing the
corresponding button then the system is still checking and no arp attacks are
being performed<br>
<br><br>
</p>

</form>
<form action="/stop/" method="post">
{% csrf_token %}
<lable style="color:white">stop ARP attack check:</lable>
<br><br>
<input type="submit" name="submit1",value='external script'>
<br><br>
</form>

```

```
</body>
</html>>
```

Code 10:

```
from django.shortcuts import render
from subprocess import run ,PIPE
import sys
def button(request):
    return render(request, 'page.html')

def ip_address(request):
    import sys
    ip_add= run([sys.executable,
'//root//PycharmProjects//pythonProject1//ip_address.py'], shell=False,
                stdout=PIPE)
    print(ip_add)

    return render(request, "page.html", {'data4': ip_add.stdout})

def output(request):
    import requests

    data = requests.get("https://reqres.in/api/users")
    print(data.text)
    data = data.text
    return render(request, "page.html", {'data': data})

def mac(request):
    inp1 = request.POST.get('param1')
    out2=run([sys.executable,
'//root//PycharmProjects//pythonProject1//ip_address.py',inp1], shell=False,
            stdout=PIPE)
    print(out2)
    out1=run([sys.executable,'//root//PycharmProjects//pythonProject1//mac_butto
n.py', inp1],shell=False,stdout=PIPE)
    print(out1)
    return render(request, 'page.html', {'data2': out1.stdout},{'data4':
out2.stdout})
```

```

def scanning(request):
    inp2 = request.POST.get('param2')
    out2=run([sys.executable, '//root//PycharmProjects//pythonProject1//network_s
canning.py', inp2], shell=False, stdout=PIPE)
    print(out2)
    return render(request, 'page.html', {'data3': out2.stdout})

def detector(request):
    #dec = request.POST.get('param_dec')
    # inp2 = request.POST.get('param1')
    out_dec = run([sys.executable,
'//root//PycharmProjects//pythonProject1//arp.py'], shell=False,
                stdout=PIPE)
    #out1 = run([sys.executable,
'//root//PycharmProjects//pythonProject1//adresa.py', inp2], shell=False,
                #stdout=PIPE)
    print(out_dec)
    # print(out1)

    return render(request, 'page.html', {'data_dec': out_dec.stdout})

# return render(request, 'page.html', {'data2':out1.stdout})
def stop(request):
    #dec = request.POST.get('param_dec')
    # inp2 = request.POST.get('param1')
    out_stop = run([sys.executable,
'//root//PycharmProjects//pythonProject1//stop.py'], shell=False,
                  stdout=PIPE)
    #out1 = run([sys.executable,
'//root//PycharmProjects//pythonProject1//adresa.py', inp2], shell=False,
                #stdout=PIPE)
    print(out_stop)
    # print(out1)

    return render(request, 'page.html', {'data_dec': out_stop.stdout})

```