

IMPLEMENTATION OF FIREWALL ON LINUX SYSTEM

A PROJECT REPORT

Submitted by

NAME	REGISTRATION NO.	EMAIL
SAURAV VARSHNEY	16BCI0175	saurav.varshney2016@vitstudent.ac.in
ABHISHEK DUBE	16BCI0196	abhishek.dube2016@vitstudent.ac.in
VIPUL GOEL	16BCI0115	vipul.goel2016@vitstudent.ac.in

Course Code: CSE2008

Course Title: NETWORK SECURITY

Under the guidance of

PROF.ANIL KUMAR K
Associate Professor, SCOPE
VIT University, Vellore.



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Department of
SCHOOL OF COMPUTER SCIENCE
AND ENGINEERING

OCTOBER,2018

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1.	INTRODUCTION	3
	1.1 ABSTRACT	4
2.	BACKGROUND OF WORK	5
	2.1 ABOUT THE APP	5
3.	OVERVIEW OF WORK	6
	3.1 PROBLEM DESCRIPTION	6
	3.2 WORKING MODEL	7
	3.3 DESIGN DESCRIPTION	8
4.	IMPLEMENTATION	9
	4.1 SOURCE CODE	9
	4.2 SNAPSHOTS	11
5.	CONCLUSION	14
6.	REFERENCES	14

1. INTRODUCTION

In computing, a firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. A firewall typically establishes a barrier between a trusted internal network and entrust external network, such as the Internet.

Firewalls are often categorized as either network firewalls or host-based firewalls.

Network firewalls filter traffic between two or more networks and run on network hardware. Host-based firewalls run on host computers and control network traffic in and out of those machines.

With the constant development of network technology today, network not only brings us a convenient and efficient life, and is accompanied by a variety of network security problems. Firewall, as a main way to prevent network attacks, is often used to prevent illegal connection and separates the internal network from the insecure networks, to protect the safety of the Linux systems which used in small and medium-sized enterprise. In this paper, the main content is to complete the function of firewall which is based on the Linux operating system, using Netfliter as firewall architecture, and the IPtable as a user space module tool. Firstly, this paper briefly analyzes the Netfilter/IPtable architecture and principle and working process of state detection technology, then, configure the firewall. At the last, the firewall experiment verified the effectiveness and safety of the design of the firewall.

1.1 ABSTRACT

We will build a firewall using Linux and Iptables that has the elegance and effectiveness of a top-end security organization.

Good firewall policy has three primary sections:

1. **Management Rules:** allow you to administer the system
2. **Access Rules:** control access to and from resources
3. **A Default Deny Rule:** drops all traffic that reaches it

Linux's iptables is a common built-in to many flavors of linux. Among other open-source firewalls, it is standard for linux distributions such as Ubuntu and Fedora. Functionally, iptables discards network "packets" according to CHAINS of rules stored in the PC's memory. These chains organize the rules and determine the order in which they are binding.

There are 3 types of built-in chains in iptables

- INPUT = packets coming INTO the PC.
- FORWARD = packets passing THROUGH PC (if it's a router).
- OUTPUT = packets leaving OUT the PC.

In order to achieve this we will ADD appropriate rules in an IP-table in linux using shell scripting.

It will allow only those IP's to communicate with our system over the network which are listed in the IP-table with ACCEPT as action. We will add an IP address in the IP-table with an appropriate action which are ACCEPT or DROP. The ACCEPT action permits the packets from that IP address to pass through the firewall. Whereas the DROP rejects the packets.

2. BACKGROUND OF WORK

2.1 ABOUT FIREWALL

A firewall is a network security system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented as both hardware and software, or a combination of both. Network firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria.

Hardware and Software Firewalls

Firewalls can be either hardware or software but the ideal configuration will consist of both. In addition to limiting access to your computer and network, a firewall is also useful for allowing remote access to a private network through secure authentication certificates and logins.

Hardware firewalls can be purchased as a stand-alone product but are typically found in broadband routers, and should be considered an important part of your system security and network set-up. Most hardware firewalls will have a minimum of four network ports to connect other computers, but for larger networks, a business networking firewall solution is available.

3. OVERVIEW OF WORK

3.1 PROBLEM STATEMENT

While firewalls protect the integrity and data of the network that sits behind it from outside attacks, inter-networks present other dangers, for example, eavesdropping. As organizations broaden the base of measures and countermeasures used to implement a comprehensive network and computer security policy, firewalls will need to communicate and interact with other devices. Intrusion detection devices running on or separate from the firewall must be able to reconfigure the firewall to meet a new perceived threat (just as dynamic filtering firewalls today "reconfigure" themselves to meet the needs of a user). Firewalls will have to be able to communicate with network security control systems, reporting conditions and events, allowing the control system to reconfigure sensors and response systems. A firewall could signal an intrusion detection system to adjust its sensitivity, as the firewall is about to allow an authenticated connection from outside the security perimeter. A central monitoring station could watch all this, make changes, react to alarms and other notifications, and make sure that all antivirus software and other content screening devices were functioning. Some products have started down this path already. The Intrusion Detection System and firewall reconfiguration of network routers based on perceived threat is a reality today. The evolution continues and firewalls are changing rapidly to address the next Internet years . As the use of Internet and internet worked computers continues to grow, the use of Internet firewalls will also grow in parallel. They not be the only security mechanism, but will cooperate with others on the network.

3.2 WORKING MODEL

Firewalls are generally categorized as network-based or host-based. Network-based firewalls are positioned on the gateway computers of LANs, WANs and intranets.

Firewalls also vary in type depending on where communication originates, where it is intercepted, and the state of communication being traced.

Network layer or packet filters

Network layer firewalls, also called packet filters, operate at a relatively low level of the TCP/IP protocol stack, not allowing packets to pass through the firewall unless they match the established rule set. The firewall administrator may define the rules; or default rules may apply. The term "packet filter" originated in the context of BSD operating systems.

Network layer firewalls generally fall into two sub-categories, stateful and stateless.

Commonly used packet filters on various versions of Unix are *ipfw* (FreeBSD, Mac OS X), *NPF* (NetBSD), *PF* (Mac OS X (> 10.4), OpenBSD, and some other BSDs), *iptables/ipchains* (Linux) and *IPFilter*

Application-layer

Application-layer firewalls work on the application level of the TCP/IP stack (i.e., all browser traffic, or all telnet or FTP traffic), and may intercept all packets traveling to or from an application.

Application firewalls function by determining whether a process should accept any given connection. Application firewalls accomplish their function by hooking into socket calls to filter the connections between the application layer and the lower layers of the OSI model.

Proxies

A proxy server (running either on dedicated hardware or as software on a general-purpose machine) may act as a firewall by responding to input packets (connection requests, for example) in the manner of an application, while blocking other packets.

A proxy server is a gateway from one network to another for a specific network application, in the sense that it functions as a proxy on behalf of the network user.

Network address translation

Firewalls often have network address translation (NAT) functionality, and the hosts protected behind a firewall commonly have addresses in the "private address range", as defined in RFC 1918. Firewalls often have such functionality to hide the true address of computer which is connected to the network.

3.4 . DESIGN DESCRIPTION

Our project is based on using linux iptables.

- Linux is one of the most used operating system.
- Managing the network traffic is one of the important part and toughest job to deal with.
- So In order to do that configuring a firewall is essential or without that system will be vulnerable.
- The default firewall used in Linux OS is IP tables.
- IP tables are used to manage packet filtering, DNAT, SNAT rules.
- IP tables come with all Linux distributions.

A packet or a datagram is a logical representation of the physical phenomena. It forms a unit of containment whereby a data can be examined, routed and filtered in regards to its destination, source and content.

IPtables discard the network traffic according to the CHAINS that are rules stored in PC memory. These chains organize the rules and determine the order in which they are binding.

The flow of data is governed by standard protocols that are bind to specific ports.

These ports are represented with the number and can be filtered and can be filtered by opening and closing these ports to accept or reject packets whose field data match that port.

Chains are grouping of rules that govern the network traffic by opening and closing the ports that can be applied or bound to an interface in a particular order.

There are 3 types of built-in chains.

- INPUT: packets coming into the pc
- FORWARD: packets passing through pc
- OUTPUT: packets leaving out PC

4.IMPLEMENTATION

4.1 SOURCE CODE

```
#!/bin/sh

echo "This is a shell script code to execute iptable function"

while [ "true" ]
do
    echo "The iptable functions are: \n 1. LIST the iptable\n 2.
ADD a rule to iptable\n 3. FLUSH rules from the iptable\n 4.
DELETE a particular rule\n 5. EXIT "
    echo "Enter your choice(1, 2, 3, 4, 5):"
    read option
    case $option in
        1)
            echo "current firewall rules are:\n "
            sudo iptables -L
            echo "\n"
            ;;
        2)
            COMMAND="sudo iptables"
            echo "Welcome!!! Create your own firewall"
            echo "Enter the name of the chain in iptable you
want add a rule"
            read CHAIN
            echo "Enter action(ACCEPT/DROP) the packets"
            read ACTION
            echo "Enter the protocol"
            read PROTOCOL
            echo "Enter the source address"
            read S_ADDR
            echo "Enter the source PORT"
            read S_PORT
            echo "Enter the destination address"
            read D_ADDR
            echo "Enter the destination PORT"
            read D_PORT
            echo "Enter the interface"
            read INTRF
            echo "$CHAIN, $ACTION, $PROTOCOL, $S_ADDR,
$S_PORT, $D_ADDR, $D_PORT, $INTRF"
            if [ "$CHAIN" ]
            then
                COMMAND="$COMMAND -A $CHAIN"
            fi
            if [ "$ACTION" ]
            then
                COMMAND="$COMMAND -j $ACTION"
            fi
        *)
            echo "Invalid choice. Please enter a valid choice."
    esac
done
```

```

        fi
        if [ "$PROTOCOL" ]
        then
            COMMAND="$COMMAND -p $PROTOCOL"
        fi
        if [ "$S_ADDR" ]
        then
            COMMAND="$COMMAND -s $S_ADDR"
        fi
        if [ "$S_PORT" ]
        then
            COMMAND="$COMMAND --source-port
$S_PORT"
        fi
        if [ "$D_ADDR" ]
        then
            COMMAND="$COMMAND -d $D_ADDR"
        fi
        if [ "$D_PORT" ]
        then
            COMMAND="$COMMAND --destination-port
$D_PORT"
        fi
        if [ "$INTRF" ]
        then
            COMMAND="$COMMAND -i $INTRF"
        fi
        $COMMAND
        echo $COMMAND
        echo "rule added"
    ;;
3)
    sudo iptables -F
    echo "rules flushed"
    ;;
4)
    echo "Enter the name of the chain from which you
want to delete a rule"
    read CHAIN
    echo "Enter the rule index"
    read INDEX
    sudo iptables -D $CHAIN $INDEX
    echo "rule deleted successfully"
    ;;
5)
    echo "Thank you"
    break
esac
done

```

4.2 SNAPSHOTS

1. Displaying the default rules in the iptables.

```
saurov@saurov-Lenovo-IdeaPad-S510p: ~/Desktop
File Edit View Search Terminal Help
This is a shell script code to execute iptable function
The iptable functions are:
  1. LIST the iptable
  2. ADD a rule to iptable
  3. FLUSH rules from the iptable
  4. DELETE a particular rule
  5. EXIT
Enter your choice(1, 2, 3, 4, 5):
1
current firewall rules are:

Chain INPUT (policy ACCEPT)
target     prot opt source                               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                               destination
```

2. Flushing rules.

```
saurov@saurov-Lenovo-IdeaPad-S510p: ~/Desktop
File Edit View Search Terminal Tabs Help
saurov@saurov-Lenovo-IdeaPad-S510p: ~/... x saurov@saurov-Lenovo-IdeaPad-S510p: ~/... x
The iptable functions are:
  1. LIST the iptable
  2. ADD a rule to iptable
  3. FLUSH rules from the iptable
  4. DELETE a particular rule
  5. EXIT
Enter your choice(1, 2, 3, 4, 5):
3
rules flushed
The iptable functions are:
  1. LIST the iptable
  2. ADD a rule to iptable
  3. FLUSH rules from the iptable
  4. DELETE a particular rule
  5. EXIT
Enter your choice(1, 2, 3, 4, 5):
```

3. Adding a rule in the iptables

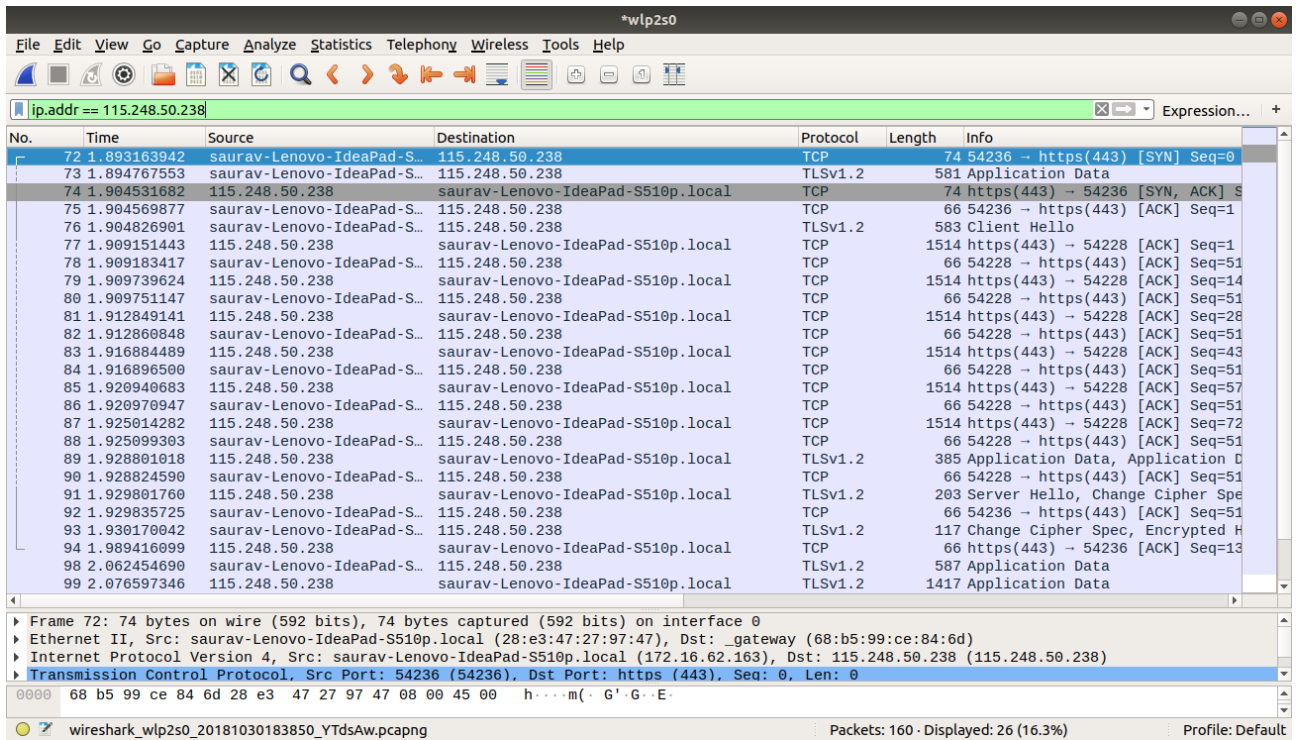
```
saaurav@saaurav-Lenovo-IdeaPad-S510p: ~/Desktop
File Edit View Search Terminal Tabs Help
saaurav@saaurav-Lenovo-IdeaPad-S510p: ~/... x saaurav@saaurav-Lenovo-IdeaPad-S510p: ~/... x
The iptable functions are:
1. LIST the iptable
2. ADD a rule to iptable
3. FLUSH rules from the iptable
4. DELETE a particular rule
5. EXIT
Enter your choice(1, 2, 3, 4, 5):
2
Welcome!!! Create your own firewall
Enter the name of the chain in iptable you want add a rule
INPUT
Enter action(ACCEPT/DROP) the packets
DROP
Enter the protocol
tcp
Enter the source address
115.248.50.212
Enter the source PORT
80
Enter the destination address
Enter the destination PORT
Enter the interface
wlp2s0
INPUT, DROP, tcp, 115.248.50.212, 80, , , wlp2s0
sudo iptables -A INPUT -j DROP -p tcp -s 115.248.50.212 --source-port 80 -i wlp2s0
rule added
```

4. Deleting a rule.

```
saaurav@saaurav-Lenovo-IdeaPad-S510p: ~/Desktop
File Edit View Search Terminal Tabs Help
saaurav@saaurav-Lenovo-IdeaPad-S510p: ~/... x saaurav@saaurav-Lenovo-IdeaPad-S510p: ~/... x
The iptable functions are:
1. LIST the iptable
2. ADD a rule to iptable
3. FLUSH rules from the iptable
4. DELETE a particular rule
5. EXIT
Enter your choice(1, 2, 3, 4, 5):
4
Enter the name of the chain from which you want to delete a rule
INPUT
Enter the rule index
1
rule deleted successfully
```

Testing the Firewall

1. Without Firewall



Wireshark capture showing network traffic without a firewall. The capture shows a series of TCP and TLS connections between saurav-Lenovo-IdeaPad-S510p.local and 115.248.50.238. The traffic includes SYN, ACK, and application data packets.

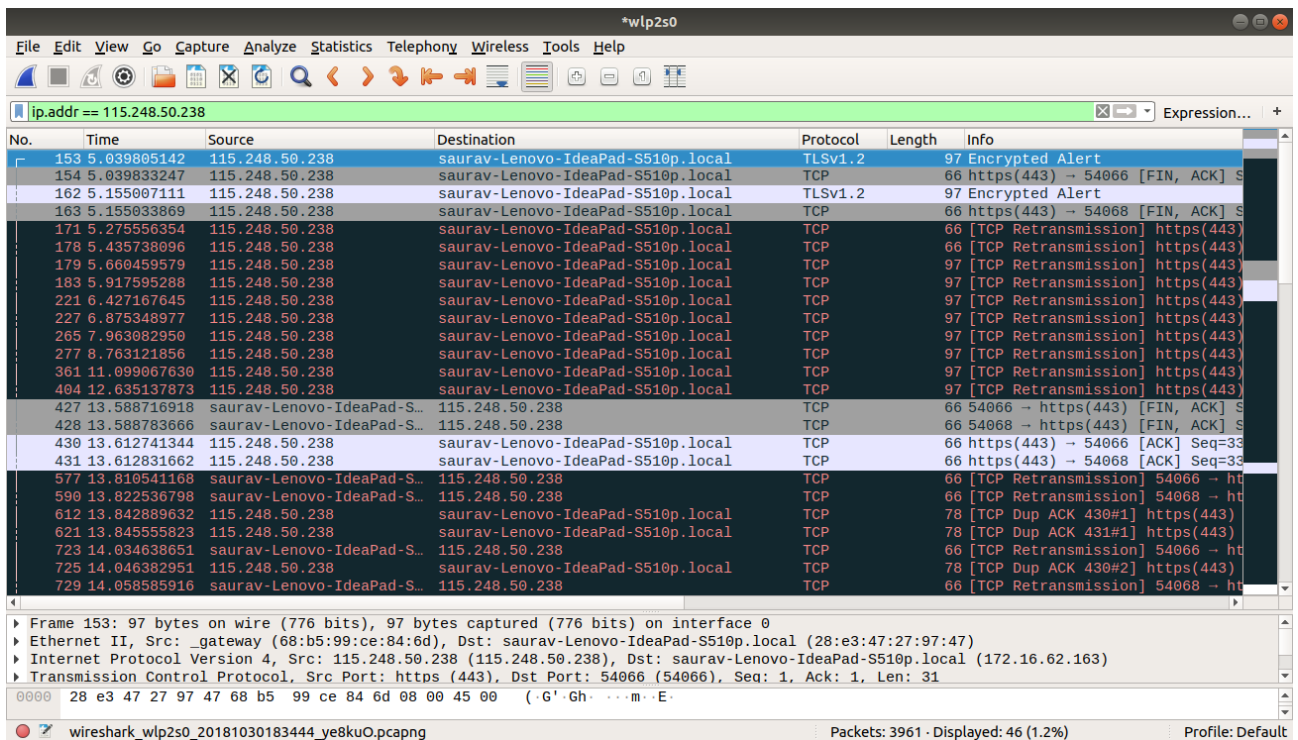
No.	Time	Source	Destination	Protocol	Length	Info
72	1.893163942	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	74	54236 → https(443) [SYN] Seq=0
73	1.894767553	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TLSv1.2	581	Application Data
74	1.904531682	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	74	https(443) → 54236 [SYN, ACK] Seq=1
75	1.904569877	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54236 → https(443) [ACK] Seq=1
76	1.904826901	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TLSv1.2	583	Client Hello
77	1.909151443	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	1514	https(443) → 54228 [ACK] Seq=1
78	1.909183417	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54228 → https(443) [ACK] Seq=51
79	1.909739624	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	1514	https(443) → 54228 [ACK] Seq=14
80	1.909751147	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54228 → https(443) [ACK] Seq=51
81	1.912849141	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	1514	https(443) → 54228 [ACK] Seq=28
82	1.912860848	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54228 → https(443) [ACK] Seq=51
83	1.916884489	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	1514	https(443) → 54228 [ACK] Seq=43
84	1.916896500	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54228 → https(443) [ACK] Seq=51
85	1.920940683	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	1514	https(443) → 54228 [ACK] Seq=57
86	1.920970947	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54228 → https(443) [ACK] Seq=51
87	1.925014282	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	1514	https(443) → 54228 [ACK] Seq=72
88	1.925099303	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54228 → https(443) [ACK] Seq=51
89	1.928801018	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TLSv1.2	385	Application Data, Application D
90	1.928824590	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54228 → https(443) [ACK] Seq=51
91	1.929801760	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TLSv1.2	203	Server Hello, Change Cipher Spe
92	1.929835725	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54236 → https(443) [ACK] Seq=51
93	1.930170042	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TLSv1.2	117	Change Cipher Spec, Encrypted H
94	1.989416099	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	66	https(443) → 54236 [ACK] Seq=13
98	2.062454690	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TLSv1.2	587	Application Data
99	2.076597346	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TLSv1.2	1417	Application Data

Frame 72: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: saurav-Lenovo-IdeaPad-S510p.local (28:e3:47:27:97:47), Dst: _gateway (68:b5:99:ce:84:6d)
Internet Protocol Version 4, Src: saurav-Lenovo-IdeaPad-S510p.local (172.16.62.163), Dst: 115.248.50.238 (115.248.50.238)
Transmission Control Protocol, Src Port: 54236 (54236), Dst Port: https (443), Seq: 0, Len: 0

0000 68 b5 99 ce 84 6d 28 e3 47 27 97 47 08 00 45 00 h...m(·G'G·E·

wireshark_wlp2s0_20181030183850_YTdsAw.pcapng Packets: 160 · Displayed: 26 (16.3%) Profile: Default

2. With a Firewall



Wireshark capture showing network traffic with a firewall. The capture shows a series of TCP and TLS connections between saurav-Lenovo-IdeaPad-S510p.local and 115.248.50.238. The traffic includes SYN, ACK, and application data packets, with some retransmissions and duplicate ACKs.

No.	Time	Source	Destination	Protocol	Length	Info
153	5.039805142	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TLSv1.2	97	Encrypted Alert
154	5.039833247	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	66	https(443) → 54066 [FIN, ACK] Seq=1
162	5.155007111	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TLSv1.2	97	Encrypted Alert
163	5.155033869	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	66	https(443) → 54068 [FIN, ACK] Seq=1
171	5.275556354	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	66	[TCP Retransmission] https(443)
178	5.435738096	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	66	[TCP Retransmission] https(443)
179	5.660459579	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	97	[TCP Retransmission] https(443)
183	5.917595288	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	97	[TCP Retransmission] https(443)
221	6.427167645	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	97	[TCP Retransmission] https(443)
227	6.875348977	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	97	[TCP Retransmission] https(443)
265	7.963082950	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	97	[TCP Retransmission] https(443)
277	8.763121856	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	97	[TCP Retransmission] https(443)
361	11.099067630	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	97	[TCP Retransmission] https(443)
404	12.635137873	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	97	[TCP Retransmission] https(443)
427	13.588716918	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54066 → https(443) [FIN, ACK] Seq=1
428	13.588783666	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	54068 → https(443) [FIN, ACK] Seq=1
430	13.612741344	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	66	https(443) → 54066 [ACK] Seq=33
431	13.612831662	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	66	https(443) → 54068 [ACK] Seq=33
577	13.810541168	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	[TCP Retransmission] 54066 → ht
590	13.822536798	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	[TCP Retransmission] 54068 → ht
612	13.842889632	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	78	[TCP Dup ACK 430#1] https(443)
621	13.845555823	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	78	[TCP Dup ACK 431#1] https(443)
723	14.034638651	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	[TCP Retransmission] 54066 → ht
725	14.046382951	115.248.50.238	saurav-Lenovo-IdeaPad-S510p.local	TCP	78	[TCP Dup ACK 430#2] https(443)
729	14.058585916	saurav-Lenovo-IdeaPad-S...	115.248.50.238	TCP	66	[TCP Retransmission] 54068 → ht

Frame 153: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface 0
Ethernet II, Src: _gateway (68:b5:99:ce:84:6d), Dst: saurav-Lenovo-IdeaPad-S510p.local (28:e3:47:27:97:47)
Internet Protocol Version 4, Src: 115.248.50.238 (115.248.50.238), Dst: saurav-Lenovo-IdeaPad-S510p.local (172.16.62.163)
Transmission Control Protocol, Src Port: https (443), Dst Port: 54066 (54066), Seq: 1, Ack: 1, Len: 31

0000 28 e3 47 27 97 47 68 b5 99 ce 84 6d 08 00 45 00 (.G'.Gh·...m·E·

wireshark_wlp2s0_20181030183444_ye8kuO.pcapng Packets: 3961 · Displayed: 46 (1.2%) Profile: Default

5. CONCLUSION

In saying this, there are multiple security strategies which any e-commerce provider can instigate to reduce the danger of assault and compromise essentially. Awareness of the dangers and the implementation of multi-layered security conventions, detailed and open protection policies and solid authentication and encryption measures will go far to assure the consumer and insure the danger of compromise is kept insignificant.

6. REFERENCES

- [Firewalls in Linux: Principles and Implementation Yordanos G. Beyene](#)