

نموذج رقم (1)

إقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

نظام تقاطع من أجل مزودي خدمة الإنترنت

An Interception System for ISPs (ISISP)

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name: ahed Ahmed Bader

اسم الطالب: عاهد أحمد بدر

Signature:

التوقيع:



Date: 25/1/2016

التاريخ: 2016/1/25

Islamic University – Gaza
Deanery of Higher Studies
Faculty of Engineering
Computer Engineering Department



An Interception System for ISPs (ISISP)

Ahed Ahmad A. Elrahman Bader

Supervisor

Prof. Mohammad Mikki

1436H (2015)



نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ عاهد أحمد عبدالرحمن بدر لنيل درجة الماجستير في كلية الهندسة قسم هندسة الحاسوب وموضوعها:

نظام تقاطع من أجل مزودي خدمة الانترنت

An Interception System for ISPs (ISISP)

وبعد المناقشة التي تمت اليوم الثلاثاء 19 ربيع الآخر 1437 هـ، الموافق 2016/01/19م الساعة الحادية عشرة صباحاً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

| | |
|------------------------|-----------------|
| أ.د. محمد أمين مكي | مشرفاً و رئيساً |
| د. أيمن أحمد أبو سمرة | مناقشاً داخلياً |
| د. يوسف نبيل أبو شعبان | مناقشاً خارجياً |

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية الهندسة / قسم وموضوعها: هندسة الحاسوب.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يستخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق ،،،

نائب الرئيس لشئون البحث العلمي والدراسات العليا

أ.د. عبد الرؤوف علي المناعمة

Acknowledgement

In The Name of **ALLAH**, the Most Beneficial and Merciful. We are very thankful to all those who have helped me in giving me support throughout performing my thesis. First of all I would like to thank my university supervisor Prof. Mohammad Mikki, who cultivated my mind with skills, providing us this opportunity to complete master degree thesis with complete support and guidance during entire period. His comments and proper feedback made me achieve this goal.

I extraordinary thankful to my parents and my wife who had been praying during my degree studies and in hard times.

TABLE OF CONTENTS

| | |
|---|------------|
| Acknowledgement | ii |
| TABLE OF CONTENTS | iii |
| LIST OF FIGURES | v |
| ARABIC ABSTRACT | vi |
| ABSTRACT | vii |
| Chapter 1 Introduction | |
| 1.1 Internet Security: What it is, and why you need it. | 1 |
| 1.2 Malicious Activity by Source | 2 |
| 1.3 Understanding how the Internet works and the security | 4 |
| 1.4 Spyware: It's the new threat your anti-virus software | 5 |
| 1.5 Local Area Network Attacks | 5 |
| 1.6 Thesis objectives | 7 |
| 1.7 Problem statement | 8 |
| 1.8 resources and tools | 8 |
| 1.9 Thesis structure | 9 |
| Chapter 2 Related Work | |
| 2.1 Wireshark | 10 |
| 2.1.1 A brief history of Wireshark | 11 |
| 2.1.2 Some intended purposes | 11 |
| 2.1.3 Features | 11 |
| 2.2 SkyGrabber | 13 |
| 2.2.1 Product features | 13 |
| 2.3 Free Network Analyzer | 14 |
| 2.3.1 How Network Analyzer Works | 14 |
| 2.4 NetworkMiner Analysis Tool | 16 |
| 2.4.1 About the SPID Algorithm | 16 |
| Chapter 3 THEORETICAL BACKGROUND | |
| 3.1 Network | 18 |
| 3.2 The Open System Interconnected Model (OSI) | 18 |
| 3.2.1 Physical Layer | 21 |
| 3.2.2 Data Link Layer | 21 |
| 3.2.3 Network Layer | 22 |
| 3.2.4 Transport Layer | 23 |
| 3.2.5 Session Layer | 23 |
| 3.2.6 Presentation Layer | 23 |
| 3.2.7 Application Layer | 23 |
| 3.3 TCP/IP Protocol Suite | 24 |
| 3.3.1 Link Layer | 25 |

| | |
|---|----|
| 3.3.2 Internet Layer | 27 |
| 3.3.3 Transport Layer Protocol | 29 |
| Chapter 4 NETWORK SECURITY ATTACKS | |
| 4.1 Background | 34 |
| 4.2 Reconnaissance Attacks | 37 |
| 4.3 Access Attack | 40 |
| 4.3.1 Password Attack | 41 |
| 4.3.2 Trust Exploitation | 41 |
| 4.3.3 Port Redirection | 42 |
| 4.3.4 Man-in-the-Middle Attack | 42 |
| 4.4 DOS Attacks | 43 |
| Chapter 5 METHODOLOGY | |
| 5.1 Introduction | 45 |
| 5.2 application module | 46 |
| 5.2.1 Listen module | 47 |
| 5.2.2 Unparsed Queue | 47 |
| 5.2.3 Parsing data | 48 |
| 5.2.4 List ALL Recorded packet | 50 |
| 5.2.5 Filter 1 and view data | 51 |
| 5.2.6 HTTP Communications List | 52 |
| 5.2.7 Filter 2 and HTTP Analysis Data View | 55 |
| 5.2.8 Extract Files From packets | 56 |
| 5.3 Active thread in the application | 58 |
| Chapter 6 Testing and analysis | |
| 6.1 introduction | 59 |
| 6.2 main form | 59 |
| 6.3 packets filter | 63 |
| 6.4 HTTP analysis form | 64 |
| 6.5 HTTP communications filter | 66 |
| 6.6 extracting files from packets | 68 |
| Chapter 7 Conclusion and future work | |
| 7.1 Conclusion | 70 |
| 7.2 future work | 71 |
| REFERENCES | 72 |

LIST OF FIGURES

| | | |
|-------------|--|-----------|
| 2.1 | Wireshark captures packets and lets you examine | 14 |
| 2.2 | SkyGrabber captures packets | 15 |
| 2.3 | Free Network Analyzer | 17 |
| 2.4 | NetworkMiner | 19 |
| 3.1 | OSI Reference Model Layer Architecture | 21 |
| 3.2 | OSI System Architecture | 22 |
| 3.3 | OSI Framework Architecture | 23 |
| 3.4 | Layer difference between OSI and TCP/IP Suite | 27 |
| 3.5 | Different Layers Protocols in TCP/IP suite | 28 |
| 3.6 | Resolution Protocols Working Scenarios | 29 |
| 3.7 | ARP Packet | 30 |
| 3.8 | IP Datagram Delivery | 31 |
| 3.9 | IP Datagram | 32 |
| 3.10 | TCP dump Output of IP Datagram | 33 |
| 3.11 | TCP Header | 34 |
| 3.12 | UDP Header | 36 |
| 3.13 | TCP Connection via SSH Tunnel | 37 |
| 3.14 | Transport Level Web Security | 37 |
| 4.1 | Basic types of Security Attacks | 39 |
| 4.2 | Passive Attacks | 40 |
| 4.3 | Active Attacks | 41 |
| 4.4 | Trust Exploitation Attack | 47 |
| 4.5 | Port Redirection Attack | 47 |
| 4.6 | Distributed Denial of Service Attack | 49 |
| 5.1 | application modules | 51 |
| 5.2 | Listen to interface module | 52 |
| 5.3 | mutual exclusion | 53 |
| 5.4 | parsing module | 55 |
| 5.5 | List ALL Recorded packet data flow | 56 |
| 5.6 | filter1 and view data normal way | 57 |
| 5.7 | Filter 1 on update parameters data flow | 57 |
| 5.8 | filter form for filter 1 parameters | 58 |
| 5.9 | HTTP Communication List data flow | 60 |
| 5.10 | filter 2 and HTTP Analysis Data View normal way | 61 |
| 5.11 | Filter 2 on update parameters HTTP Analysis Data View | 61 |
| 5.12 | filter form for filter 2 parameters | 62 |
| 5.13 | extraction file from received packets | 63 |

| | | |
|-------------|---|-----------|
| 5.14 | extract file by user | 64 |
| 5.15 | extract file by host | 64 |
| 6.1 | main form application | 66 |
| 6.2 | interface selection combo box | 67 |
| 6.3 | the result just after press start button | 67 |
| 6.4 | the details of packet after user select item from grid | 68 |
| 6.5 | the Data of the packet | 68 |
| 6.6 | the Hexadecimal of the Data | 68 |
| 6.7 | the Packet Detail | 69 |
| 6.8 | packet tooltip | 69 |
| 6.9 | packet filter form | 70 |
| 6.10 | the main form after filtering by source address | 70 |
| 6.11 | the main form after filtering by HTTP protocol and data | 71 |
| 6.12 | HTTP communication grid | 71 |
| 6.13 | HTTP communications form after user choose item | 72 |
| 6.14 | request header of HTTP communication | 72 |
| 6.15 | response header of HTTP communication | 73 |
| 6.16 | HTTP communications filter form | 73 |
| 6.17 | the HTTP communication form after HTTP “Response is | 74 |
| 6.18 | extract files form | 75 |
| 6.19 | the file that extracted from the packets | 76 |

Abstract

Due to rapid changes and consequent new threats to networks there is a need for the design of systems that enhances network security. These systems make network administrators fully aware of the potential vulnerability of their network.

This master's thesis builds an Interception System for ISPs (ISISP) which is an active defense and complex network surveillance platform designed for ISPs to meet their most rigorous security requirements. It is motivated by the great concern for government agencies, ecommerce companies and web developer organizations. It is also used by network administrators to understand the vulnerabilities affecting computer networks in a way similar to a hacker's attack but is meant to improve network security rather than attacking it.

ISISP is a lawful interception system with the main task of obtaining network communications, giving access to intercepted traffic to lawful authorities for the purpose of data analysis and/or evidence. Such data generally consist of signaling, network management information, or the content of network communications.

ISISP intends to develop an interception system that intercepts ISP's IP traffic for the purpose of gathering and analyzing network vulnerability real time data, rebuilt the web pages that the user ask and send some alarm for some type of malicious behavior

Chapter1

Introduction

Anyone who uses the Internet is at risk of virtual attacks. A decade ago, the Internet was something only technical people talked about. It was a new limitless source of information, with very few users. Today, the Internet has already become an essential part of our lives. It's where we access our banking records, credit card statements, tax returns and other highly sensitive personal information. By the end of this decade, over 2 billion people will be connected to the Internet that's about one third of the world's current population [1].

But with all the good things the Internet offers us, it also opens the door to serious, potentially devastating threats. Unlike corporate and government computer systems, few personal computers have any safeguards beyond basic virus protection. That means anytime you're online, you are a potential target for online criminals and hackers. And if you have high-speed Internet access, your computer is online most of the time, making Internet criminals and hackers a 24-hour-a-day, year-round threat to you, your personal information, and your family [2].

1.2 Malicious Activity by Source

Malicious activity usually affects computers that are connected to high-speed broadband Internet, because these connections are attractive targets for attackers. Broadband connections provide larger bandwidth capacities than do other connection types, plus faster speeds, the potential for constantly connected systems, and typically a more stable connection. Symantec categorizes malicious activities as follows:

- **Malicious code:** This includes programs such as viruses, worms, and Trojans that are covertly inserted into programs. The purposes of malicious code include destroying data, running destructive or intrusive programs, stealing sensitive information, and compromising the security or integrity of a victim's computer data [3].
- **Spam zombies:** These are remotely controlled, compromised systems specifically designed to send out large volumes of junk or unsolicited email messages. These email messages can be used to deliver malicious code and phishing attempts [4].
- **Phishing hosts:** Phishing hosts are computers that provide website services in order to illegally gather sensitive user information while pretending that the attempt is from a trusted, well known organization by presenting a website designed to mimic the site of a legitimate business [5].
- **Bot-infected computers:** Malicious programs have been used to compromise computers to allow an attacker to control the targeted system remotely. Typically, a remote attacker controls a large number of compromised computers over a single reliable channel in a botnet, which can then be used to launch coordinated attacks [6].

- Network attack origins: These measure the originating sources of attacks from the Internet. For example, attacks can target SQL protocols or buffer overflow vulnerabilities [7].
- Web-based attack origins: These measure attack sources that are delivered via the web or through HTTP. Typically, legitimate websites are compromised and used to attack unsuspecting visitors [8].

Commentary

- In 2014, the United States and China remained the top two sources overall for malicious activity. The overall average proportion of attacks originating from the United States in 2014 increased by 0.4 percentage point compared with 2013, while the same figure for China saw an increase by 1.2 percentage points compared with 2013. Countries ranking in the top 10 for 2013 continued to appear in the same range in 2014 [5].
- The United States remains in first position as a source of all activities except for spam zombies, bots, and network attacks. Vietnam remains in first position for spam zombies, and China remains primary for bots and network attacks.
- Of all bot activity, 16.5 percent originated in China: China was the main source of bot-infected computers, an increase of 7.3 percentage points compared with 2013.
- Of all web-based attacks, 21.1 percent originated in the United States: Web-based attacks originating from the United States decreased by 5.1 percentage points in 2014.
- Of all network attacks, 28.7 percent originated in China: China has the largest population of Internet users not only in the Asia region but also globally, which attributes to the high rates of attacks.

- Of all phishing websites, 46.6 percent were hosted in the United States: The United States is the second largest population of Internet users in the world, which could be one of the reasons that it accounts for highest number of phishing websites.
- Of all spam zombies, 10.1 percent were located in Vietnam, an increase of 5.1 percentage points compared with 2013. The proportion of spam zombies located in the United States dipped by 0.4 percentage point to 3.9 percent, resulting in the United States being ranked in ninth position in 2014, the same as in 2013.
- Of all malicious code activities, 19.8 percent originated from the United States, an increase of 2.9 percentage points compared with 2013, giving the country the same ranking as in 2013. With 12.2 percent of malicious activity originating in India, the country was ranked in second position.[9]

1.3 Understanding how the Internet works and the security threats you face

When you access the Internet, your computer sends a message over the Web that uniquely identifies your computer and where it is located. This allows the information you've requested to be returned to you. Often, this requested information carries with it unwanted hidden software created by hackers and online criminals. This software installs itself on your computer and can either be just a nuisance or pose a more serious threat to you, your identity and sensitive financial information. Usually the nuisances are visible and easy to identify, while the more dangerous threats are typically invisible, silent, and difficult to detect until it's too late. The key to a safe, enjoyable Internet experience is understanding the difference between what a threat is and what isn't [10].

1.4 Spyware: It's the new threat your anti-virus software won't find

If you're a casual computer user, chances are you've heard about viruses and what they can do to your computer. Viruses are serious threats that attack your computer and data, and generally disrupt your life; but they aren't used to steal your sensitive personal information. Internet criminals create spyware to do this. They want you to believe that anti-virus software is all the protection you need. As important as it is to your security, anti-virus software can't detect or stop this newer, more sophisticated threat from entering your computer. Stopping spyware requires even greater protection [11].

1.5 Local Area Network Attacks

Some of the most popular LAN attacks include:

- Session hijacking
- IP spoofing
- Man-In-The-Middle attack

In the following we present these attacks in some detail.

Session Hijacking

Session hijacking is also referred to as TCP session hijacking, a security attack on a user session over a protected network. The most common method of session hijacking is called IP spoofing, when an attacker uses source-routed IP packets to insert commands into an active communication between two nodes on a network and disguising itself as one of the authenticated users. This type of attack is possible because authentication typically is only done at the start of a TCP session. Another type of session hijacking is known as a man-in-the-middle attack, where the

attacker, using a sniffer, can observe the communication between devices and collect the data that is transmitted [12].

IP Spoofing

IP spoofing is a technique used to gain unauthorized access to computers, where by the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host. To engage in IP spoofing, a hacker must first use a variety of techniques to find an IP address of a trusted host and then modify the packet headers so that it appears that the packets are coming from that host [13].

IP spoofing, also known as IP address forgery or a host file hijack, is a hijacking technique in which a cracker masquerades as a trusted host to conceal his identity, spoof a Web site, hijack browsers, or gain access to a network. Here's how it works: The hijacker obtains the IP address of a legitimate host and alters packet headers so that the legitimate host appears to be the source. When IP spoofing is used to hijack a browser, a visitor who types in the URL (Uniform Resource Locator) of a legitimate site is taken to a fraudulent Web page created by the hijacker. For example, if the hijacker spoofed the Library of Congress Web site, then any Internet user who typed in the URL www.loc.gov would see spoofed content created by the hijacker. If a user interacts with dynamic content on a spoofed page, the hijacker can gain access to sensitive information or computer or network resources. He could steal or alter sensitive data, such as a credit card number or password, or install malware . The hijacker would also be able to take control of a compromised computer to use it as part of a zombie army in order to send out spam.

Web site administrators can minimize the danger that their IP addresses will be spoofed by implementing hierarchical or one-time passwords and data encryption/decryption techniques. Users and administrators can protect themselves and their networks by installing and implementing firewalls that block outgoing packets with source addresses that differ from the IP address of the user's computer or internal network [14].

Man-in-the-middle

A man in the middle attack is one in which the attacker intercepts messages in a public key exchange and then retransmits them, substituting his own public key for the requested one, so that the two original parties still appear to be communicating with each other.

In a man in the middle attack, the intruder uses a program that appears to be the server to the client and appears to be the client to the server. The attack may be used simply to gain access to the message, or enable the attacker to modify the message before retransmitting it. Man in the middle attacks are sometimes known as fire brigade attacks. The term derives from the bucket brigade method of putting out a fire by handing buckets of water from one person to another between a water source and the fire [15].

1.6 Thesis objectives

ISISP is used for packet forensics that allows multiple policies to operate simultaneously on the entire IP packet stream. This means that while network managers/governmental agents search for different strings inside each IP packet, they can also intercept VoIP calls, extract dialed digits and correlate DHCP log-ins with IP addresses. Each policy can have different

resulting actions, such as forwarding packets to another analysis system or generating security threat reports.

- Building an ISP Internet traffic monitoring / network behavior recording components
- IP traffic record keeping
- Lawful IP packet interception
- The ability to restore the files transferred over the LAN (HTML files, images, videos, ...)

1.7 Problem statement

The ISP is suffering of the leak of the systems that support low level of packet sniffing and mentoring and most of the present systems has not the Ability to restore the real files from the packets and these system is very expensive.

My system will work on sniffing all the packets comes throw the interface and record it the files , supporting packets filtering and recover all files from the sniffed baskets with low cost

1.8 resources and tools

The following resources and methods will be required

1. Visual studio 2012
2. SQL server 2008
3. High traffic LAN
4. Server with windows serves 2012

1.9 Thesis structure

This thesis consist of six chapters: Introduction, related works, theoretical background, network security attacks, proposed system and conclusions and future work. The main points discussed in the chapters are

2. Chapter 1 "Introduction" gives a short introduction about the internet security , Malicious Activity , Local Area Network Attacks
3. Chapter 2 "related works" presents related work to the thesis
4. Chapter 3 "theoretical background" gives a short background about The Open System Interconnected Mode (OSI) and TCP/IP Protocol
5. Chapter 4 "network security attacks" explain Reconnaissance Attacks, types of Access Attack and DOS attack
6. Chapter 5 "methodology" explain the methodology of the system and the structures of the application
7. Chapter 6 "experimental result" traveling inside the application and shows how the application work and the results
8. Chapter 7 "Conclusion and future work" summaries and future works expected

Chapter 2

Related Work

2.1 Wireshark

Wireshark is a network packet analyzer that tries to capture network packets and tries to display that packet data as detailed as possible. You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course). In the past, such tools were either very expensive, Wireshark is perhaps one of the best open source packet analyzers available today [16].

2.1.1 A brief history of Wireshark

In late 1997 Gerald Combs needed a tool for tracking down network problems and wanted to learn more about networking so he started writing Ethereal (the original name of the Wireshark project) as a way to solve both problems.

Ethereal was initially released after several pauses in development in July 1998 as version 0.2.0. Within days patches, bug reports, and words of encouragement started arriving and Ethereal was on its way to success. Not long after that Gilbert Ramirez saw its potential and contributed a low-level dissector to it.

In October, 1998 Guy Harris was looking for something better than tcpview so he started applying patches and contributing dissectors to Ethereal. In late

1998 Richard Sharpe, who was giving TCP/IP courses, saw its potential on such courses and started looking at it to see if it supported the protocols he needed. While it didn't at that point new protocols could be easily added. So he started contributing dissectors and contributing patches.

The list of people who have contributed to the project has become very long since then, and almost all of them started with a protocol that they needed that Wireshark or did not already handle. So they copied an existing dissector and contributed the code back to the team.

In 2006 the project moved house and re-emerged under a new name: Wireshark.

In 2008, after ten years of development, Wireshark finally arrived at version 1.0. This release was the first deemed complete, with the minimum features implemented. Its release coincided with the first

Wireshark Developer and User Conference, called Sharkfest.

2.1.2 Some intended purposes

Here are some examples people use Wireshark for:

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals

2.1.3 Features

The following are some of the many features Wireshark provides: available for UNIX and Windows, capture live packet data from a network interface,

open files containing packet data captured with tcpdump /WinDump, Wireshark, and a number of other packet capture programs, import packets from text files containing hex dumps of packet data, display packets with very detailed protocol information, save packet data captured, export some or all packets in a number of capture file formats, filter packets on many criteria, search for packets on many criteria, colorize packet display based on filters, create various statistics [17, 18].

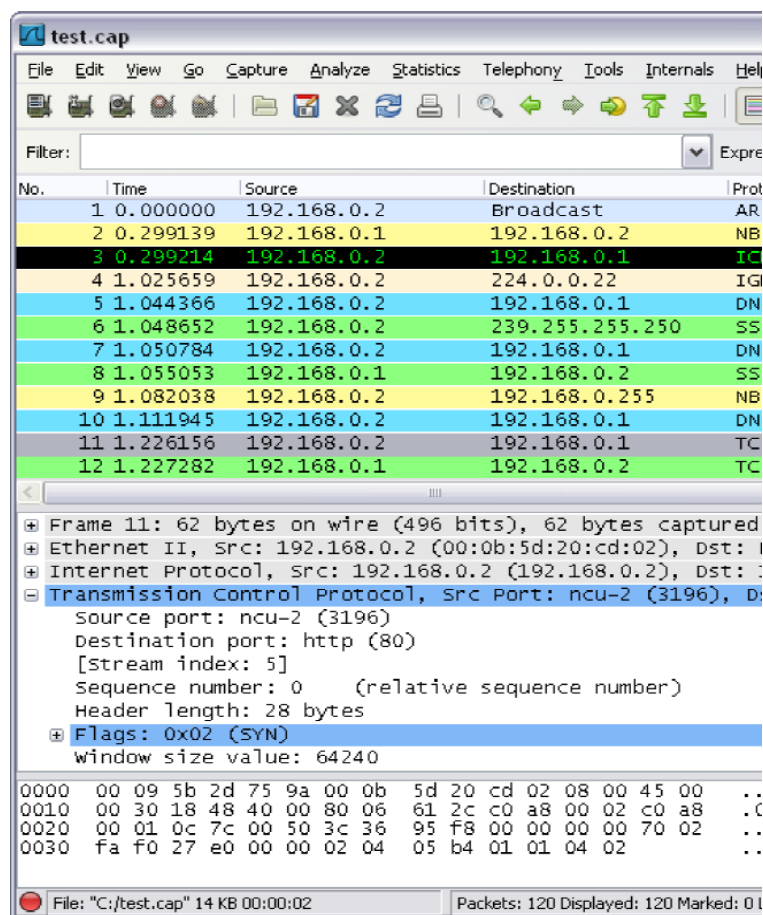


Figure 2.1 Wireshark captures packets and lets you examine their contents.

2.2 SkyGrabber

SkyGrabber is offline satellite internet downloader and allows to accept satellite internet data, assemble it in files (avi,mp3,mp4,etc.) and save files onto your hard disk. The program has user-friendly interface, diverse filter system and satellite provider manager inside.

SkyGrabber works only with free-to-air satellite internet data

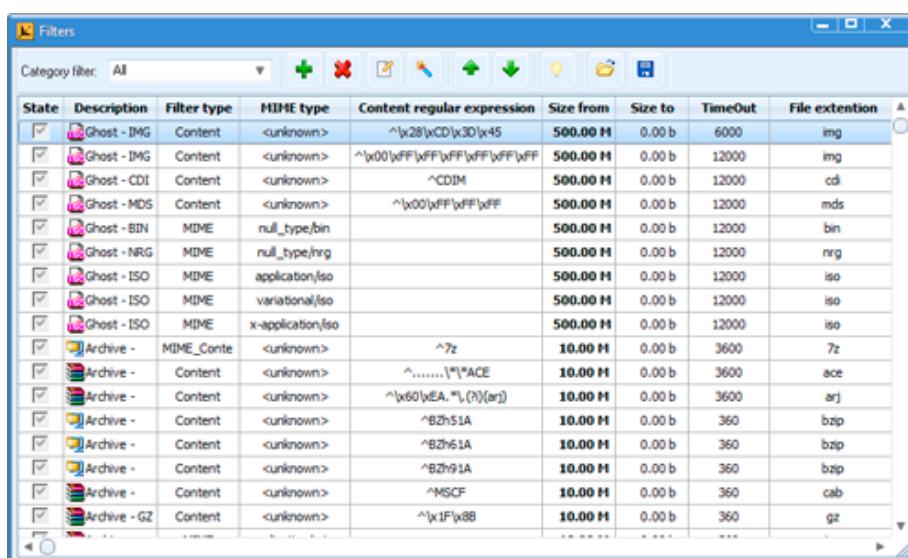


Figure 2.2 SkyGrabber captures packets

2.2.1 Product features

- Assemble TCP/IP sessions in files
- Lock frequency to accept satellite internet data
- Support DiceqC (uncommitted/committed)
- Satellite provider manager
- Filter manager by file type
- Filter manager by IP addresses (MAC addresses)

- Monitoring system resource information (CPU usage, Memory usage, Free disk space)
- Monitoring satellite signal information (Level, Quality)
- Displaying progress bar of downloaded files

2.3 Free Network Analyzer

Free Network Analyzer is a software network packet sniffer and protocol analyzer for Windows platform. Using this free network monitoring software you may intercept any data transmitted via wired broadcast or wireless LAN (WLAN) and Internet connections of your computer. Network sniffer allows you to capture, filter and display any traffic data flowing through your network adapters. It decodes captured network communication packet's raw data, displaying the binary, hex, decimal and text field values in the each packet, and analyzes its contents according to the RFC and other specifications. Packets data is parsed, extracted and represented in simple human-readable form, allowing you to perform effective forensic analysis of any data transferred via your PC network interfaces.

2.3.1 How Network Analyzer Works

Free network protocol analyzer installs NDIS filter driver over the network adapter device driver and then monitors all requests passed via Windows network interface.

Program parses, decodes and analyses entire content of all packets passing through network adapters. Any traffic which flows via opened network ports may be also captured and analyzed, allowing you to view and trace all data transferred by network applications or devices. This free network data

explorer supports advanced data filtering, highlighting and searching for patterns with regular expressions, which makes this software extremely useful for deep network traffic analysis.

Free network protocol analyzer software is designed for effective intercepting, capturing, decoding and monitoring of network communications. This free network traffic monitoring software processes monitored data in real-time even for high data rates; it remains responsive during 1 Gbit/s network communications monitoring even on budget desktop PC. Superior performance of this network protocol analyzer makes it extremely useful for real-time network monitoring applications.

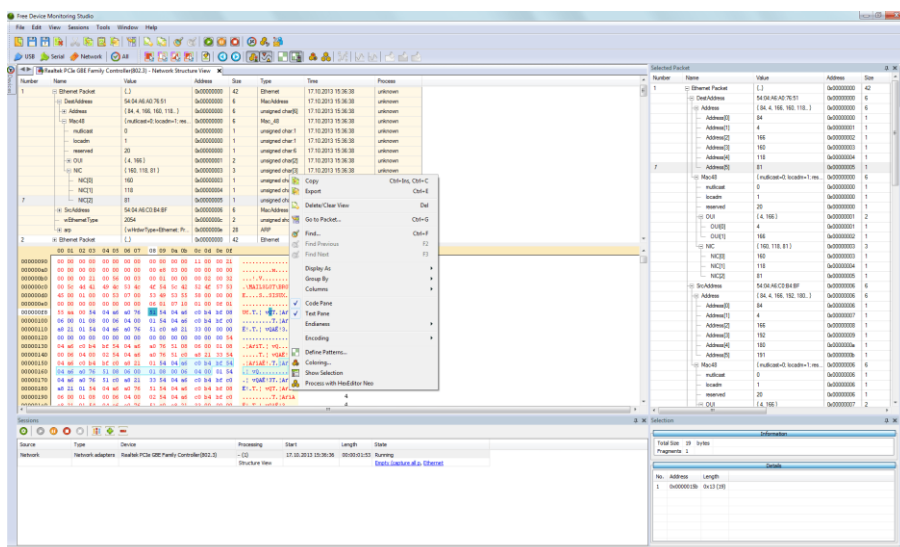


Figure 2.3 Free Network Analyzer

2.4 NetworkMiner Analysis Tool

NetworkMiner is a network forensic analysis tool that they have developed in order to facilitate the task of performing network forensic investigations as well as conducting incident response (Hjelmvik, 2008).

NetworkMiner is designed to collect data about hosts on a network rather than to collect data regarding the traffic on the network. It has a graphical user interface where the main view is host centric (information grouped per host) rather than packet centric (information showed as a list of packets/frames).

One of the most appreciated functions in NetworkMiner is the ability to easily extract files from captured network traffic in protocols such as HTTP, FTP, TFTP and SMB. NetworkMiner actually reassembles files to disk on the fly as it parses a PCAP file. A lot of other useful information like user credentials, transmitted parameters, operating systems, hostnames, and server banners etcetera can also be extracted from network traffic with NetworkMiner. All of this is of course performed fully passive, so that no traffic is emitted to the network while performing the network forensic analysis.

NetworkMiner is available at SourceForge1 as open source under a GPL license [19].

2.4.1 About the SPID Algorithm

The Statistical Protocol IDentification (SPID) Algorithm was designed by Russ McRee with the purpose of identifying the application layer protocol in TCP sessions without relying on port numbers. The SPID algorithm performs

protocol identification based on simple statistical measurements of various protocol attributes. These attributes can be defined by all sorts of packet and flow data, ranging from traditional statistical flow features to application level data measurements, such as byte frequencies and offsets for common byte-values (Hjelmvik and John, 2009).

Russ McRee has made a proof-of-concept implementation of the SPID algorithm available on SourceForge2 as open source under a GPL license

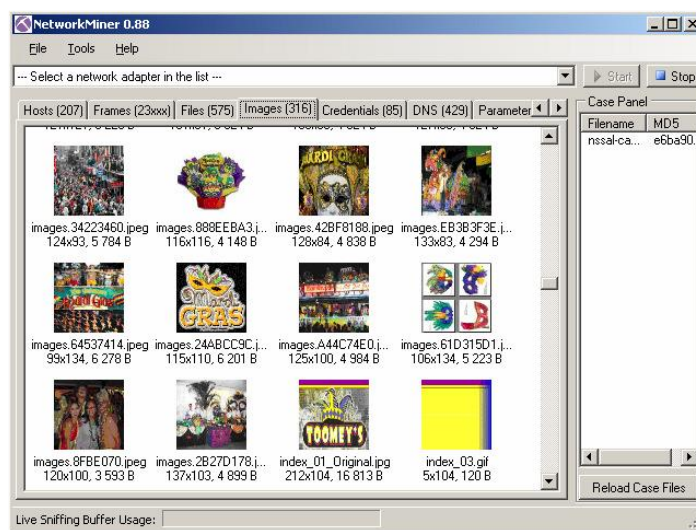


Figure 2.4 NetworkMiner

Chapter 3

THEORETICAL BACKGROUND

In this chapter we will describe the basic concept of data communication network. The network layer protocols are the major part in a communication network. This chapter includes the description of the role of network layer protocols in a communication model; it also explains the functional parameters of these protocols in different level of data communication. These parameters are in the form of protocol header fields. We will study the header field of these protocols and analyze that how an attacker can use or change these protocol header fields to accomplish his/her malicious goals. The in-depth study of the structure of OSI layer protocols & TCP/IP layer protocols can carry out this objective.

3.1 Network

The network consists of collection of systems connected to each other through any communication channel. The communication channel may consist of any physical “wired” or logical “wireless” medium and of any electronic device known as node. Computers and printers are some of the examples of nodes in a computer network and if we talk about the telecommunication network these may be mobile phones, connecting towers equipment and main control units. The characteristic of a node in the network is that; it has its own identity in the form of its unique network identification. The main functionality of any network is to divide resources among the nodes. The network under certain rules finds resources and then shares it between the nodes in such a way that authenticity and security issues are guaranteed.

The rules for communication among network nodes are the network protocols. A protocol is the complete set of rules governing the interaction between two systems [3]. It varies for varying different working assignments between nodes communication.

3.2 The Open System Interconnected Model (OSI)

In 1997, The International Standard Organization (ISO) designed a standard communication framework for heterogeneous systems in network.

As per functionality of communication system in open world, this system is called Open System Interconnection Model (OSI). The OSI reference model provides a framework to break down complex inter-networks into such components that can more easily be understood and utilized [3]. The purpose of OSI is to allow any computer anywhere in the world to communicate with any other, as long as both follow the OSI standards [20].

The OSI reference model is exploited into seven levels. Every level in OSI Model has its own working functionality; these levels are isolated but on the other hand cascaded to each other and have communication functionality in a proper flow between them. With reference to above standard communication framework, this set of layers known as OSI layers. Functionality of each layer is different from each and each layer has different level and labels. (Shown in Figure 3.1)

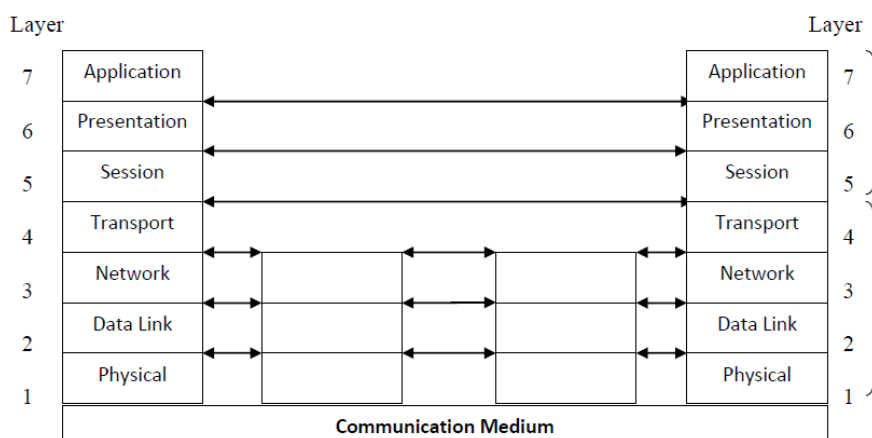


Figure 3.1: OSI Reference Model Layer Architecture

On the other hand if we see the system architecture of OSI, three level of abstraction is explicitly recognized; the architecture, the service specifications, and the protocols specifications (see Figure 3.2) [4]. The OSI service specifications are responsible for specific services between user and system in a specific layer. Parallel OSI protocol specifications are responsible that, which type of protocol is running against the specific communication service. So it is clear that the combination of these two parts become OSI system architecture.

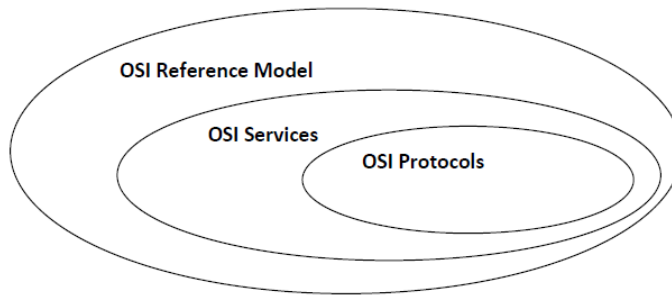


Figure 3.2: OSI System Architecture

It is patent that the OSI reference model consists of seven layers and each layer offers different functionalities, different services with different protocols. Whereas each layer, with the exception of the lowest, covers a lower layer, effectively isolating them from higher layers functions.[5]. Similarly the design principle of information hiding; the lower layer are concerned with greater level of details, upper layer are independent of these details. Within each layer, both services are provided to the next higher layer and the protocol to the peer layer in other system are provided [6] (see Figure 3.3). Therefore we may say that as any change occurs in any layer-N, then it may effect only on its lowest layers-N-1. Due to isolation, the higher layers-N+1 is not affected or it can say that remaining reference model will not effect [21].

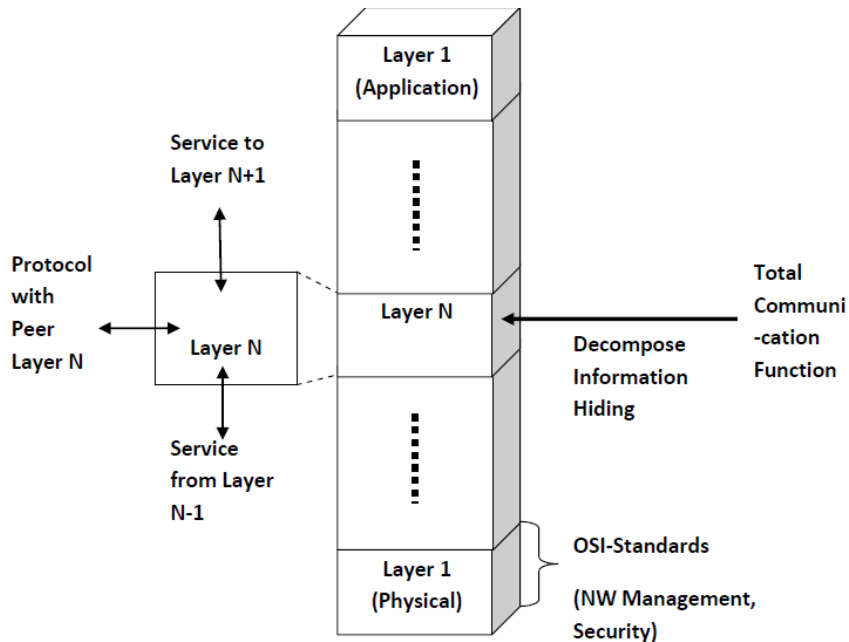


Figure 3.3: OSI Framework Architecture

3.2.1 Physical Layer

The lowest layer in OSI model is Physical Layer; it facilitates the connectivity between system interface cards and physical mediums. This layer understands and transforms electrical/electronic signals in the form of bits. So that it administrates physical “wire” and/or logical “wireless” connection establishment between the hardware interface cards and communication medium; example of physical layer standard includes RS-232, V.24 and V.35 interfaces [22].

3.2.2 Data Link Layer

In OSI Reference Model the Data Link Layer is the second layer. Data Link layer is responsible for control methods which provides proper format of data and it can access data flow errors in physical layer. The data format in data link layer is in the form of frames. Therefore we say that the data link layer is responsible for defining data formats to include the entity by which information is transported. Error control procedures and other link control procedures may occur in physical layer [5]. Like cyclic redundancy check (CRC); the error checking mechanism that run at the time of transmission of a frame from source side. The same mechanism will run at the destination

side if they found any difference after comparison then receiver makes a request to source to send that frame again. Data link layer is responsible for following service [23].

- Encapsulation
- Frame Synchronization
- Logical link control (LLC)
 - Error control
 - Flow control
- Media Access Control (MAC)
 - Collision Detection
 - Physical Addressing

The data link layer is further subdivided into two layers, Logical link Control (LLC) and Media Access Control. The logical link control is responsible for flow control and error detection in data. Whereas media access control is responsible for controlling the traffic congestion and physical address reorganization.

3.2.3 Network Layer

The third layer in OSI Reference Model is the Network Layer. This layer is responsible to make a logical connection between source and destination. The data at this layer is in the form of packets. The network layer protocols provide the following services

Connection mode: The network layer has two types of connection between source and destination, first one is known as connectionless communication which does not provide connection acknowledgement. The example of connectionless communication is Internet Protocol (IP). The second type of connection is connection-oriented which provides connection acknowledgement. TCP is an example of this connection.

IP Addressing: In computer networks every node has its own unique ID. By this unique ID sender and receiver always make right connection. This is because of the functionality of network layer protocol, which has source address and destination address in their header fields. So there is less chance of packet loss, traffic congestion and broadcasting [22].

3.2.4 Transport Layer

The fourth layer in OSI reference model is Transport Layer. It contains two types of protocols, first is Transport Control Protocol (TCP) which is connection oriented protocol and supports some upper layer protocols like HTTP and SMTP. The second is User Datagram Protocol (UDP) which is a connection less protocol. Like TCP it also supports some upper layer protocols such as DNS, SNMP and FTP. The main thing in transport layer protocols is that they have port addresses in their header fields [24].

3.2.5 Session Layer

The fifth layer in OSI Reference Model is Session Layer. The Session Layer is responsible for session management i.e. start and end of sessions between end-user applications. It is used in applications like live TV, video conferencing, VoIP etc., in which sender establishes multiple sessions with receiver before sending the data. Session Initiation protocols (SIP) is an example[23].

3.2.6 Presentation Layer

The sixth layer in OSI Reference Model is Presentation Layer. This layer is responsible for presentation of transmitted/received data in graphical mode. Data compression and decompression is the main functionality of this layer. The data encryption is done before transmission in presentation layer [25].

3.2.7 Application Layer

The seventh and the last layer of OSI Reference Model is Application Layer. This layer organizes all system level applications like FTP, E-mail services etc. [25].

3.3 TCP/IP Protocol Suite

The TCP/IP Protocol Suite was developed before OSI reference model [8]. The OSI reference model consists of seven layers whereas TCP/IP protocol suite has only four layers (Figure 3.4) [9]. In comparison to OSI reference model, TCP Suite has high level of communication traffic awareness between sources to destination. The TCP/IP Suite has administrative communication controlled and reliable data processing. It has dozens of layer components and communication set of rules which provide reliable service performance and data security [10]. Each layer in TCP/IP suite is responsible for a specific communication service and all these layers are cascaded and support each other (Figure 3.5) [10]. The main protocols of this suite are TCP and UDP, which exist in transport layer. TCP is an acknowledgeable protocol that provides reliability in data transmission while UDP is non-acknowledgeable protocol and is used in data streaming services like video conferencing, VOIP, etc.

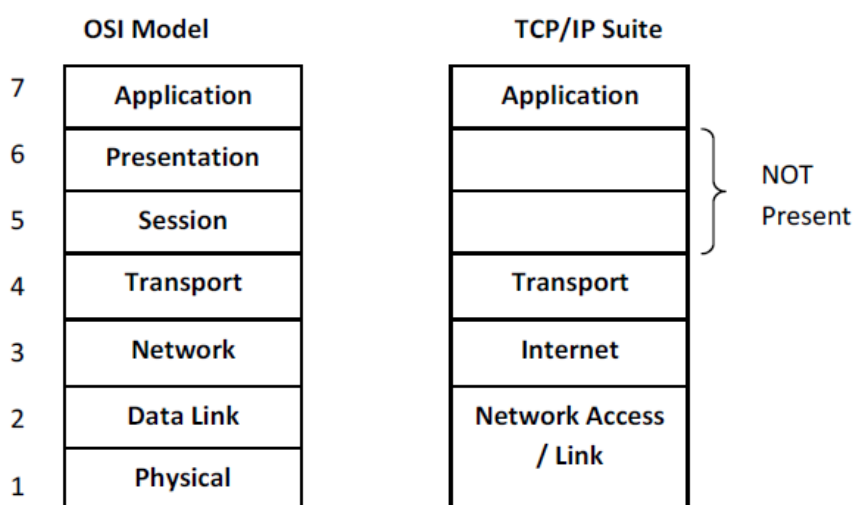


Figure 3.4: Layer difference between OSI and TCP/IP Suite

The layer structure of TCP/IP suite is similar to OSI Model. In TCP/IP Suite the Link Layer covers the last two layers (physical and data link layer) of OSI model. Presentation and Session Layers of OSI model do not exist in TCP/IP protocol suite. [26]

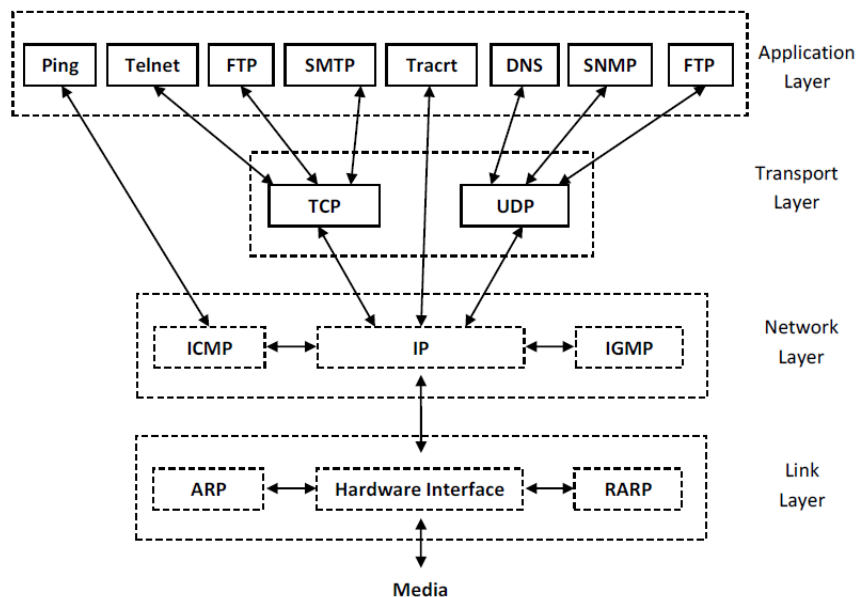


Figure 3.5: Different Layers Protocols in TCP/IP suite

3.3.1 Link Layer

This layer is also known as data link layer or network interface layer. Link layer interfaces the network interface card and the communication medium. The important role of link layer is address resolution that provides mapping between two different forms of addresses with ARP and RARP protocols (see Figure 3.6) [10]. For proper functionality; it has complete information of network interface cards, i.e. driver details and kernel information. It interprets between two systems in network for the sake of information of source address and destination address from software address to hardware address to send information on physical medium, because the kernel only recognizes the hardware address of network interface cards not the IP address or Physical address. Address resolution Protocols (ARP) translates an IP Address to a Hardware Address whereas Reverse Address Resolution Protocol (RARP) converts a hardware address to IP Address [27]. (See Figure 3.6)

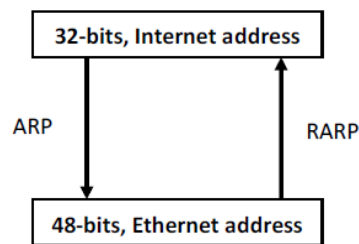


Figure 3.6: Resolution Protocols Working Scenarios

Address Resolution Protocol

The interpretation of data transmitted to communication medium from network layer depends on ARP and RARP link layer protocols. Network layer has source and destination address which is also called the logical address or 32-bits of IP Address, but before sending the information on a network via communication medium it is required to change this address IP address into 48-bits of hardware address which is also called Ethernet address or MAC Address. The reason for changing the address is that, the communication medium is directly connected to the Ethernet interface cards and it may assess the data via serial communication lines. ARP operation; a network device during transmission in a communication medium performs sequence of operations . Packet format of ARP is also clarified this (Figure 3.7) 28].

- ARP request: A broadcast request in the form of Ethernet frames for the whole network. Request is basically a query for getting a hardware address against an appropriate IP.
- ARP reply: Appropriate hardware address generates a send back rep; response to sender against its query, in the form of its hardware address with its IP address.
- Exchange: request-reply information.
- Send: IP datagram to destination host.

In Data link layer, Ethernet and Token ring have the same hardware length, as well if it sends a query request then operation has 1 notation and in query response that has changed to in notation of 2.

| | | | |
|-------------------------------|-----------------|-------------------------------|----|
| 0 | 8 | 16 | 31 |
| Hardware Type | | Protocol Type | |
| Hardware Length | Protocol Length | Operation | |
| Sender Hardware Address (0-3) | | | |
| Sender Hardware Address (4-5) | | Sender IP Address (0-3) | |
| Sender IP Address (2-3) | | Target Hardware Address (0-1) | |
| Target Hardware Address (2-5) | | | |
| Target IP Address | | | |

Figure 3.7: ARP Packet

Reverse Address Resolution Protocol

RARP packet format and operation is similar to ARP operation but has reverse working functionality. RARP generates a query for IP address against appropriate MAC address. This design is for diskless workstation which has a big usage in corporate environment [10]. In this scenario the diskless workstation can get their IP address from server against their specific hardware addresses.

3.3.2 Internet Layer

The second layer of TCP/IP suite protocol structure is Internet or network layer. It generates a service request to Data Link layer protocol and provides services against Transport layer application request. The role of internet layer protocol (IP) is very important in internetworking data transmission and in receiving prospects; datagram delivery is the main task of this layer [29]. (Figure 3.8)

Internet protocol

Internet protocol is an important protocol of the internet layer as well for the whole internetworking communication. The protocol structure of internet layer is IP datagram and each IP datagram consists of the source IP address and destination IP address which is of 32-bit physical address. [10]. Consider the layer traffic scenario; it receives UDP/TCP segment request form transport layer and add some layer information tags as a prefix and convert

it into IP datagram [5]. That is concerned with the exact datagram delivery in the form of source and destination IP address. Figure 3.9 shows the whole datagram packet

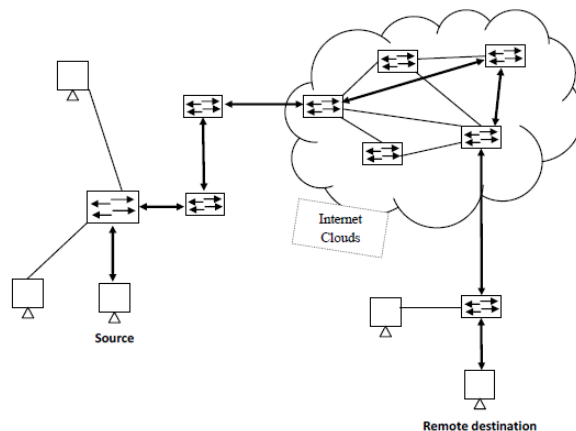


Figure 3.8: IP Datagram Delivery

The “*version*” notifies the current IP version that exists in IP datagram; either it is version 4 or 6. The “*type of service*” indicates multiple services like delay, throughput and cost etc. “*Time to live*” is a countdown counter that gradually down to zero. Two conditions exists here, either packet successfully reached to its destination or discarded before TTL reached to zero. If TTL counter reaches to 0 IP packet discarded from the network. The main advantage of TTL is that it overcomes the network traffic congestion issue. “*Flags*” contain 3 bit length as shown in IP datagram figure; they play an important role in successfully transmission of data packet at destination end. The 32-bit “*source address*” and “*destination address*” are the physical addresses of source and destination. These fields perform an efficient role to hitch-hike of IP traffic on network. A hacker can exploit the IP datagram by make some changes in it when the packet is traveling in communication medium in the form of hex code. Hacker can do this with the help of any network sniffing application or by use of TCP-dump and mapping application. By using TCP-dump, malicious hacker can see the IP header datagram information and then can change the values by his/her malicious mind. Let’s take an example [12] Examine the IP traffic with TCP-dump application gives all necessary information which could help in malicious act. This is the output of TCP-dump and it is in Hex-code for better understanding we may change it into binary and decimal code. From the figure 3.10 the information we can

get; IP version (either 4 or 6), total length of IP packet, TTL of the packet, type of protocol either TCP or UDP, source and destination address “4500 00b2 4ea6 2000 8006 ee3f c0a8 4803 c0a8 4804” [29]

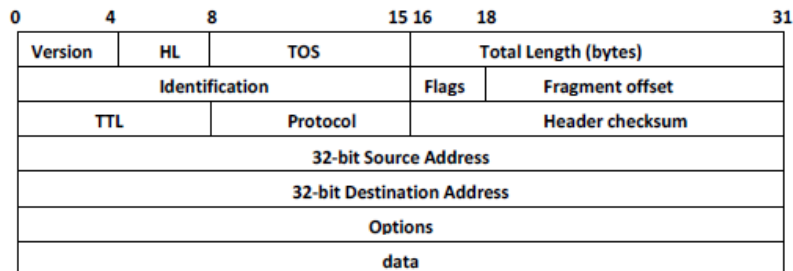


Fig 3.9: IP Datagram

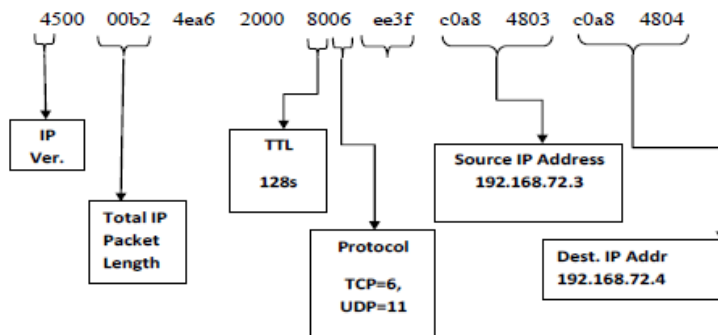


Figure 3.10: TCP dump Output of IP Datagram

3.3.3 Transport Layer Protocol

Transport layer is the third layer in TCP/IP protocol suite which consists of two protocols, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Different functionalities of these protocols comes up with different architectural models or headers. The main feature of Transport layer protocol is, introducing a new protocol header that was not present in lower layer protocols, that is a port concept. Working with Internet Protocol features; IP address and port function from transport layer protocol, against a specific services or application run on server that we get “socket” concept [30].

TCP Protocol

The Transmission Control Protocol (TCP) removes the existing drawbacks in internet protocol (IP); TCP makes connection before sending the data to destination that is why known as connection oriented protocol. Connection oriented functionality provides the reliable data transmission in communication. Similarly TCP notifies and remove errors with the help of error detection function which detects the errors during transmission. Acknowledgement from receiver shows that packet is successfully delivered to the destination that is why we say that TCP is a reliable protocol. If sender receives acknowledgement with error packet then sender sends again error-less packet toward receiver. Below is the figure showing some more features of TCP Protocol (Figure 3.11) [5].

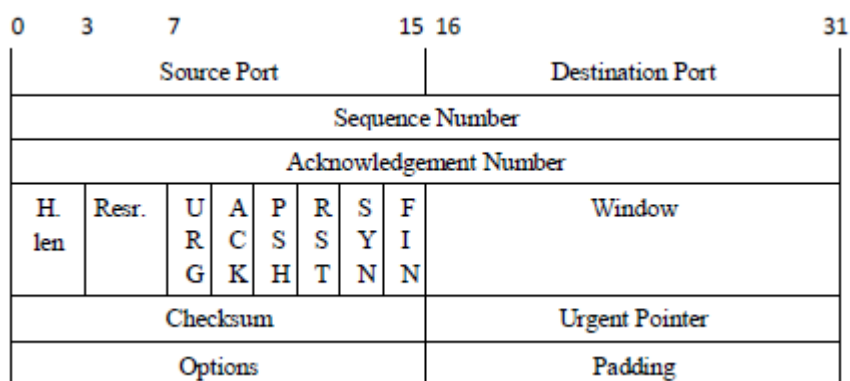


Figure 3.11: TCP Header

Source and Destination Port: TCP protocol provides the source and destination services port. There are many applications, services and system processes that are associated with unique ports. By allocating specific ports with specific services on server end administrators can allow/deny system services to specific client or group of clients, i.e. web services.

Sequence and Acknowledgement Number: The error detection is done by sequence number and acknowledgement number. The sequence number controls the sequence of sent packets, if sender receives any acknowledgment with error-packet then sender once again send error-less packet so that maintain the exact sequence number of the packet.

Code bits or flag indication: There are six different flags indicator in TCP header which identifies different functions in TCP packet. Following are six flags [5].

1. URG Bit: This flag indicates the urgent data request; it assigns higher priority to urgent packets.
2. ACK Bit: This flag indicates that the datagram has an acknowledgement of previous packet or that the datagram contain some specific acknowledgement number value.
3. PSH Bit: This flag indicates push bit. It work against with URG bit.
4. RST Bit: RST bit indicates reset request for connection from sender to receiver.
5. SYN Bit: Synchronization of sequence numbers of TCP packet.
6. FIN Bit: It is a notification shows that sender wants to stop the data.
7. Checksum: The checksum field in TCP header provides error detection in TCP packets. In prospective of network security. The above fields in TCP header are very important for any hacker. They can use these flags to achieve their malicious goal [31, 32].

UDP Protocol

Second protocol of transport layer is User Datagram Protocol (UDP). UDP is a connection less protocol i.e., it does not establish a connection between sender and receiver before sending the data. Many applications which does not require connection establishment before transmission uses UDP examples of applications are Mobile TV, VoIP etc.

As compared to TCP header UDP has very simple header (Figure 3.12). UDP does not provide reliability of data [10] because of no acknowledgement flag. But infect UDP is faster than TCP and in some applications speed is more important than reliability like in VoIP Applications. However source port, destination port and checksum fields have same functionality as in TCP header which we have already described above.

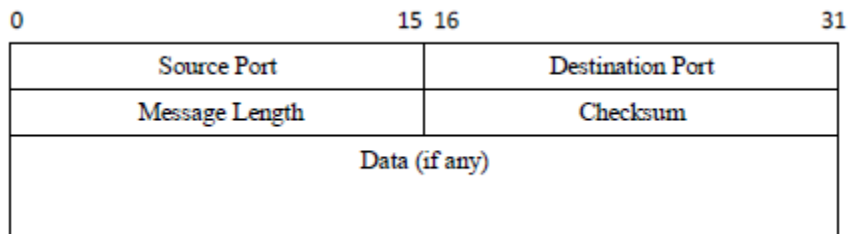


Figure 3.12: UDP Header

Security Level Protocols

There are some protocols which provide security at transport layer. Protocols are:

- Secure Socket Layer (SSL)
- Transport Layer Security (TLS)
- Secure Shell (SSH)

All above security protocols work on transport layer. Secure Shell (SSH) is a protocol that provides secure network communication channel at transport layer. It is used in sever-client connection environment. Initially user authentication is done at both ends similar like in TCP communication where three-way handshake connection is establish before data transmission, then SSH makes a secure tunnel between server and client before communication starts. In SSH, the connection established on the base of SSH user and server authentication after this it provides a communication tunnel for data transmission [33, 34] (Figure 3.13).

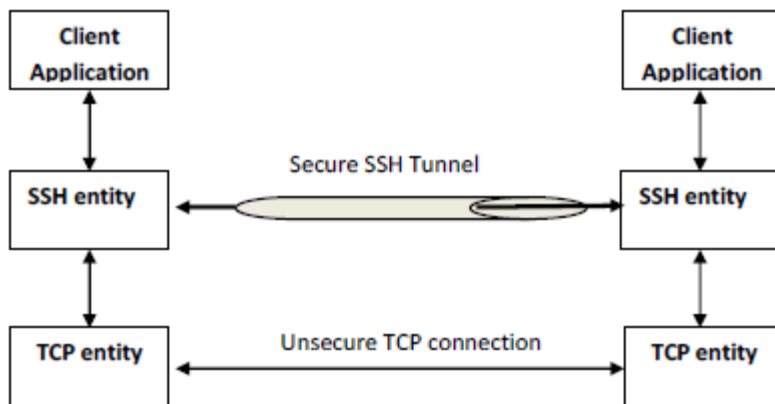


Figure 3.13: TCP Connection via SSH Tunnel

Similarly SSL and TLS provide security to web applications at transport level. (Figure 3.14) .

| | | |
|------------|-----|------|
| HTTP | FTP | SMTP |
| SSL or TLS | | |
| TCP | | |
| IP | | |

Figure 3.14: Transport Level Web Security

Chapter 4

NETWORK SECURITY ATTACKS

4.1 Background

To compromise between opening a system and lock it down so that no one can use it, is called security and any action that compromises the security is called a security attack. A system which is providing the services required by the user accurately and preventing the illegal use of system resources is called a secure system. Attacks can be categorized into following basic categories [35].

- **Interruption:** For using the data or resources it is necessary that they are available 24 hour/day 7day/weak for the authorized parties, when and where they need it. Attack on the availability of data is called interruption. Availability can be affected by intentional or un-intentional acts. Examples of un-intentional acts are, accidentally system crash, deletion and overwriting of data and sometime due to non-human factors like flood, fires and earthquakes. Whereas destruction of infrastructure due to wars, strikes and some attacks by hackers that crashes the system, such as denial of service (DOS) and distributed denial of service (DDOS) attacks are the examples of intentional acts. Protection against availability attacks includes backup and restoration [36].
- **Interception:** The core concept is that the data should be hiding from unauthorized users. If someone who is unauthorized to see private data, can see or copy the data that can further be used in intensive active attack. Such an attack is known as attack on confidentiality. Data integrity can be accomplished by strong authentication and strict access controls, because some time authorized users may also a threat for confidentiality of data. They can obtain another person's credentials [37].
- **Modification:** Integrity of data deals with prevention of intentional or unintentional modification of data. Attack on integrity of data called modification. Different algorithms used for validation of data that can resist in alteration of data. Protection of data from modification is

foremost concern than detection. Integrity of data could maintain at many layers of OSI system model [38].

- Fabrication: Attack on authenticity called fabrication. Authenticity means that message is coming from the apparent source. It assures that you are who you say you are. User name and password is the most common way to achieve authentication, some other techniques are like smart cards and digital certificates.

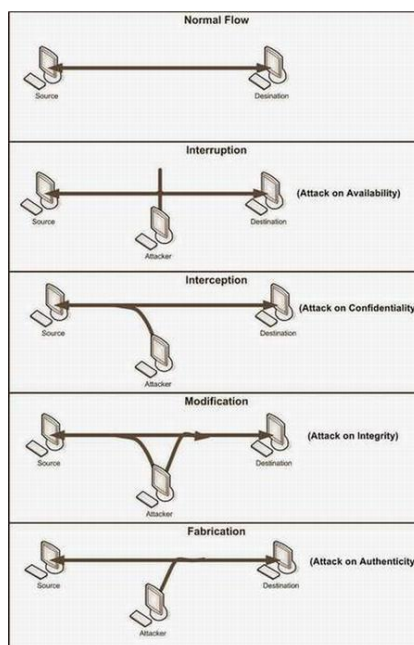


Figure.4.1. Basic types of Security Attacks

On the basis of these four attacks we can further classify security attacks as *passive attacks* and *active attacks*. Passive attacks are only involved in monitoring of the information (interception). The goal of this attack is to obtain transmitted information. Two types of passive attacks are “*release of message content*” and “*traffic analysis*”. Passive attacks are hard to detect because they do not involve in any alteration. Different encryption schemes are used to prevent against these attacks [39].

Figure 4.2 shows passive attacks types.

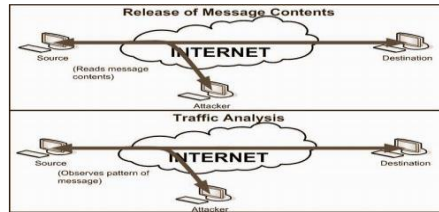


Figure 4.2 Passive Attacks

Active attacks are involved in modification of data (interception, modification, fabrication) or creation of false data. These attacks are further subdivided into four categories, "*masquerade*", "*replay*", "*modification of data*" and "*denial of service*". When an unauthorized user tries to pretend as an authorized user is called masquerade attack. Replay attacks involved in capturing the message between two communication parties and replay it to one or more parties. Bring the network down to its knees by flooding the useless traffic in network is called denial of service attack. Figure 4.3 are shows active attacks.

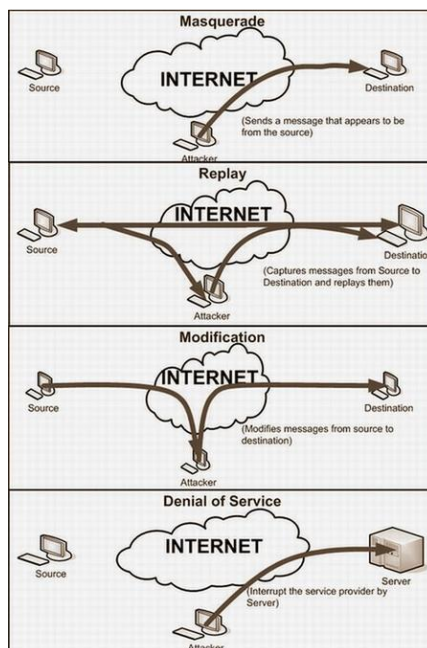


Figure 4.3 Active Attacks

4.2 Reconnaissance Attacks

Gathering information against a targeted host or network is called reconnaissance attack. Attacker analyze the target host and try to discover the details like alive IP addresses, open ports of the network, failure of operating system, and types of services and protocols running on the network. Reconnaissance attacks are common they are not so much dangerous because they are not involved in any kind of alteration or destruction of data but on the other hand they show the vulnerabilities in the network. They allow hackers to see which ways are open to access the system and provide enough information to them which they can further use in denial of service (DOS) attacks [40]. Some basic reconnaissance attacks are:

- A. Packet Sniffers
- B. Port scan and ping sweep
- C. Internet information queries

A. Packet Sniffers

Packet sniffer is a tool or device that can be used for capturing the packet at data link layer. Packet sniffer is not only a hacker's tool but it can be used both by the hacker for eavesdropping and by the administrators for network monitoring and troubleshooting. Tcpdump, windump, wireshark (ethereal) and Dsniff are examples of different sniffing tools. Sniffing can be of two types depending on the network [41].

- Passive Sniffing
- Active Sniffing

Passive Sniffing

Passive sniffing is used in hubbed networks. The drawback of using the hub in network was that, the hub broadcast a packet to each and every machine on the network. There is a filter on each machine which decides whether to accept or discard the packet. If a packet addresses to a specific machine then filter decide to accept it otherwise discard the packet. Sniffer disables this filter so that network traffic can be analyzed. This stage is called "promiscuous mode" [16]. Hence if „Bob“ on computer A sends a message to „John“ on computer B, a sniffer on computer C can easily capture the contents of that message even without knowing Bob and John. Passive

sniffing is hard to detect because it generates no traffic on network. This type of sniffing worked well when hubs were used. To avoid passive sniffing most of the networks nowadays are using switches instead of hubs.

Active Sniffing

Active sniffing is performed on switched network. A switch limits the sniffer to see the broadcast packets. Switch worked as a central entity, rather than broadcasting it simply get message from source machine and send it directly to the addressed machine. So if computer C is in promiscuous mode it cannot see the message form Bob to John. It does not mean that sniffing is not possible in switched networks. Media Access Control (MAC) flooding and poisoning of the Address Resolution Protocol table (ARP) are the ways to hack a switched network.

- MAC Flooding
- Spoofed ARP Messages

Switches worked on the basis of MAC addresses. They maintain an address resolution protocol (ARP) table in a special type of memory called Content Addressable Memory (CAM). ARP table has all the information that which IP address is mapped to which MAC address.

The act of overloading the CAM is known as MAC flooding. Low memory in older or cheaper switches can cause MAC flooding. Flooding of too many MAC addresses can fill up the memory so that switch cannot hold more entries. At this stage switch goes to a failopen mode [38] and cannot perform IP to MAC mappings, starts behaving like a hub and starts transmitting the data to all machines. In MAC flooding attacker inject large amount of traffic which may draw attention towards hacker. This traffic can be detected by any sniffer detecting software [42].

The other technique to hack a switch network is called ARP poisoning. A review of ARP is that it is almost similar to Domain Name Server (DNS). DNS resolves domain names to IP addresses while ARP resolves IP addresses to MAC addresses. Hacker fools the switch and tries to pretend the destination machine. He tries to convince the switch that the IP address of another trusted host belongs to him. A very interesting thing is that it is also up to the attacker that which IP address he wants to redirect to his system, spoofing the system, spoofing the default gateways will redirect all host messages towards the attacker. However for this, attacker has to poison host ARP table. The other way is to poison the ARP cache of a central entity of the network, hacker express that the IP address of switch (or router) is mapped with his

MAC address. Through this way all the traffic first goes towards the attacker then the router

B. Port Scan and Ping Sweep

Port scan and ping sweep are two common network probes typically used to run various test against a host or device to find vulnerable services. They are helpful to examine the IP address and the services which are running on a device or host. In port scanning hacker sends a packet to each target port and reply message indicates that either the port is open or closed which is further helpful to launch an attack against a specific service. For example if hacker finds that port 143 (IMAP port) is open then on next step he/she tries to find out that which version of IMAP is running if that version is vulnerable then hacker can access the machine as a super user using an “*exploit*” program (program that automatically break the security hole). The most popular probing tool is Nmap (Network Mapper) [43].

Different types of Nmap scans are

- **TCP Connect Scan:** It makes a complete TCP connection that's why is easy to detect.
- **TCP SYN Scan:** Attacker sends a SYN to each target port, if target port is open target sends SYN-ACK. The attacker then sends a RESET packet and aborts the connection. This type of scan is also called half-open because attacker connects to the port and breaks the connection just before full connection. These types of scans are hard to detect.
- **FIN Scan:** attacker sends a fin packet which is able to pass by firewalls without modification; open ports ignore the packet while close ports sent back a RESET packet.
- **ACK Scan:** ACK scan is good in network where firewalls are running. It did not classify the port as open or closed, if reset comes back from the target it classify the port as “unfiltered” otherwise “filtered”.

The method of finding that which IP addresses are alive is called ping sweep. Attacker sends an ICMP packet to each machine (with in a range) to a targeted network. The aim is to find out the machines which are alive and which are not alive. These ICMP replies from different machines are logged into a file for future reference. Network administrators may also use ping sweep to figure out which systems are alive and which are not for diagnostic reasons.

Fping is a tool used for performing ping sweep. Working on round robin function, takes a list of IP addresses, sends a ping packet to an IP address and immediately proceed toward next IP address. To detect ping sweep there are different tools available. Example is Ippl a protocol logger that have the ability to log ICMP, TCP and UDP packets.

C. Internet Information Queries

DNS queries provide the particular information of domain and the addresses associated with that particular domain. IP queries display the range of IP addresses and for which domain that addresses are associated. Ping sweep presents a clear picture of a particular environment. After these queries port scan start by the hacker which leads him to find out which ports are open and which services are running on these ports. Finally the whole information can be helpful when hacker tries to compromise any system through these services [44].

4.3 Access Attack

We discussed earlier that there are vulnerabilities in services which are running on a system like web services, FTP services and any authentication services that authenticates the user and hacker can exploit these vulnerabilities. Access attacks occur when a malicious hacker exploit these vulnerabilities and succeed to access the confidential information of any organization [45].

Different types of network attacks are

- Password Attacks
- Trust Exploitation
- Port Redirection
- Man-in-the-middle attack

4.3.1 Password Attack

Several methods can be used for password attack. Trojan horse, IP spoofing and packet sniffers can show the detail of the user like user name and password. We may refer password attack as, repeated attempts to find the user information (user name or password). Once an attacker succeeds then he/she has the same access right which compromised account has [46]. Most common weaknesses in an organization are

- Weak passwords.
- Default Passwords (most of the devices and applications have set on their default password which we forgot to change).
- Password are stored as plain-text.

4.3.2 Trust Exploitation

When a hacker attacks on a computer which is outside a firewall and that computer has a trust relationship with another computer which is inside the firewall, the hacker can exploit this trust relationship. We can mitigate this type of attack by using private VLANs between switches or by limiting the trust relationship between systems which are inside and outside the firewall. We can also reduce this by eliminating useless trust relations between different servers. For example if our AAA (Authentication, Authorization, and Accounting) server is inside the DMZ (Demilitarized Zone), there is no need to have a relation of AAA server with the file server [47]. Figure 4.4 explaining trust exploitation phenomena

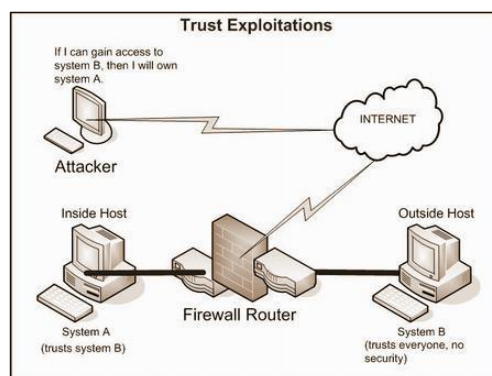


Figure 4.4 Trust Exploitation Attack

4.3.3 Port Redirection

It is another type of trust exploitation attack in which a hacker bypasses the security mechanism. Consider the below network in which hacker on the outside have the ability to access the public computer but not the computers which are in DMZ or which are inside the firewall. If public computer compromised by the hacker then hacker installs a software that can redirect the traffic towards the hacker, directly to the inside computers. In this way hacker makes a tunnel for communication and bypasses the security firewall. See figure 4.5 for port redirection attack [48].

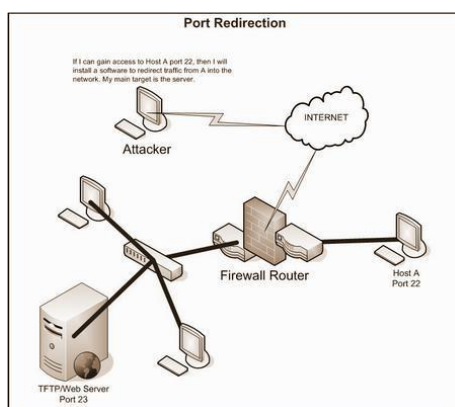


Figure 4.5 Port Redirection Attack

4.3.4 Man-in-the-Middle Attack

When hackers succeed to intrude himself between two communication parties this type of attack is called MITM (Man-in-the-Middle) attack. In this way hacker can intercept data between source and destination host, can modify data and retransmit it to the destination host and can also inject any type of false data. MITM attacks can affect on availability, confidentiality, integrity and authenticity of data. Strong cryptography can mitigate this type of attack. SSL, SSH and use of IPSec also gives end to end security (entire connection is encrypted) [49].

4.4 DOS Attacks

Types of attack that bring the network down in such a way that resources are not available even for authenticated users are known as DOS attacks. Malicious hacker saturated the target machine with useless traffic so that it cannot respond or too slow to respond and sometimes unavailable. Attacker may target a single machine to make it impossible for outgoing connections on the network or may attack on the whole network to make it impossible for incoming and outgoing traffic. For example attack on web site of any organization. *Ping of death*, *SSPing*, *Land*, *Win Nuke* and *SYN flood* are some of the examples of DOS attacks. In SYN flood attack hacker sends a SYN packet to target host which then responds with SYN acknowledgement, at the end attacker does not send any ACK packet to the target host that causes the connection to remain in half open state. TCP connection does not remove this connection from its table and wait to expire this session, attacker takes the advantage of this and continues sending new SYN packets until TCP SYN queue is filled and cannot accept new connections [33]. The common method for blocking DOS attack is to place a filter which examines the pattern of data; if same pattern of data came frequently then filter can block that message [50, 51].

Distributed Denial of Service (DDOS)

In DDOS attacks several compromised systems are used to launch an attack against a targeted host or network. For targeting a host attacker first compromises some other hosts on network and installs some software for controlling them usually these compromised hosts are called agents or zombies. Using these agents attacker launches overwhelm attack against the target. Compromised systems control with different software like *Trino* and *Shafit*. Examples of DDOS attacks are *SMURF*, *MYDoom* and *TFN*. DDOS attacks are very hard to defend. To trace out the intruder is also very difficult as they are on the back side and using other hosts against the victim [49]. Figure 4.6 is describing distributed denial of service attack.

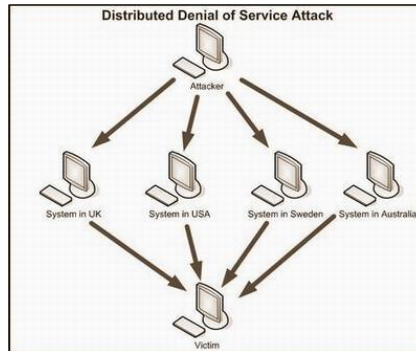


Figure 4.6 Distributed Denial of Service Attack

Chapter 5

METHODOLOGY

5.1 Introduction

IUG_shark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course).

This application introduce many helpful tools such as

- Capture live packet data from a network interface.
- Display packets with very detailed protocol information.
- Support both LAN and wireless interface
- Support many type of packet filters (source, destination, protocol type, packet direction, ...)
- HTTP analysis (extract the HTTP data from the packets (method ,ULR ,file name ,content type ,compression type))
- Support filter on the HTTP data (HTTP methods, content type ,HTTP response ,user IP , Host website ,content size)
- Extract files from packets (extract file, extract all file by user ,extract all file by host)
- support high traffic data because of the good memory management

5.2 application module

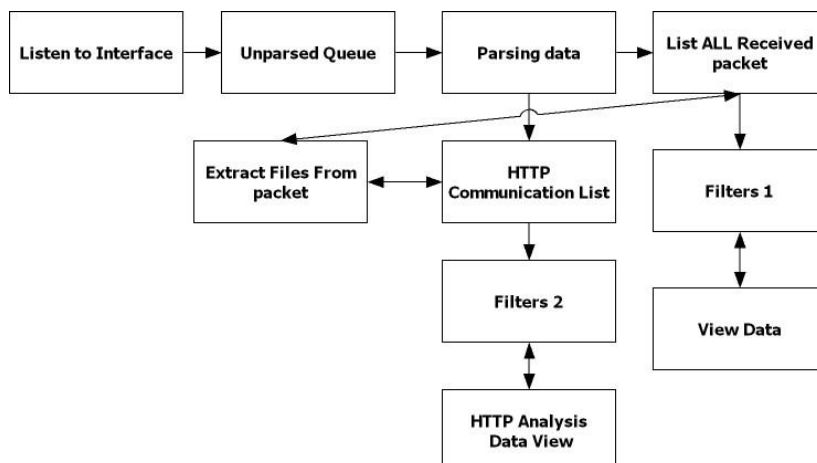


Figure 5.1 application modules

My Application consist of many modules as shown in figure 5.1 the first module is *Listen to interface* which target is to choose the interface and create thread to start listening, when data is transmitted or received the thread will copy the packet and push it to the *unparsed queue* witch target is to contain the data with mutual exclusion because of multithreading. *The parsing module* is a thread that reads the packets from *unparsed queue* and analysis it then extract the IP packet header, TCP/UDP header and HTTP header and add them to the All Received packets list and *HTTP communications list*.

Filter 1 module gets the filter parameters from GUI form and filter the all recorded packets and Display them with very detailed protocol information.

HTTP communication List store all information about all communications between the uses and the web host and display completed communications only, *filter 2* module gets the filter parameters from HTTP Analysis form and filter the communications List and Display them with very detailed protocol information.

Extract Files From packet module get the sonication information from communication list and then get all packets related the sort it according to *Sequence Number* and combine the data as binary to file.

5.2.1 Listen module

The listen module consist of the following steps

- Identify and select Network interface
- Create listening thread
- check listening status
- start listening
- On receive data add it to unparsed list

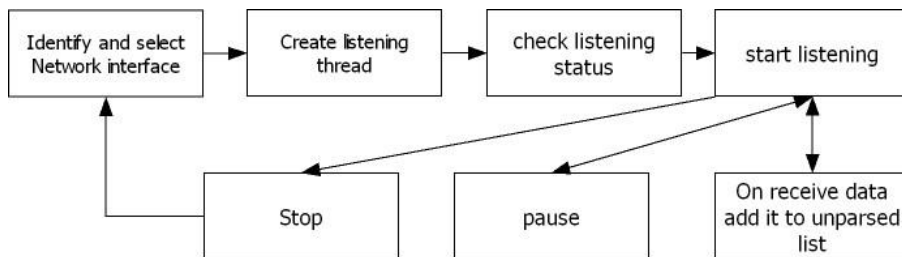


Figure 5.2 Listen to interface module

Figure 5.2 shows the steps of the listen module and the data flow beginning from selecting the interface to listen until adding the data to the unparsed list

5.2.2 Unparsed Queue

It is normal queue I used it as data store between listening to packets and parsing data. And because of multithreading I have uses mutual exclusion to prevent data modification collision as shown in figure 5.3

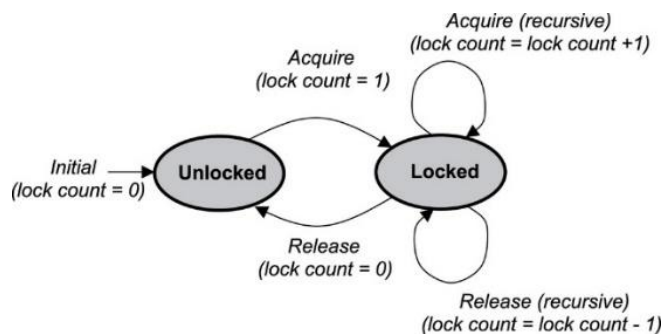


Figure 5.3 mutual exclusion

5.2.3 Parsing data

The parsing module is a thread that was created in the mainframe module. The main target for this module is to read the packets captured by listening module and analysis then extract the IP packet header, TCP/UDP header and HTTP header and add them to the All Received packets list and HTTP communications list as shown in figure 5.4

IP Header class consist of the following variables

- VersionAndHeaderLength 8 bits for version and header length
- DifferentiatedServices 8 bits for differentiated services (TOS)
- TotalLength 16 bits for total length of the header & message
- Identification 16 bits for identification
- FlagsAndOffset 16 bits for flags and fragmentation offset
- TTL 8 bits for Time To Live (TTL)
- Protocol 8 bits for the underlying protocol
- Checksum 16 bits containing the checksum of the header
 - (checksum can be negative so taken as short)
- SourceIPAddress 32 bit source IP Address
- DestinationIPAddress 32 bit destination IP Address
- Direction one bit direction of the packet
- Data the data transmitted with the packet

TCP Header class consist of the following variables

- SourcePort 16 bit for the source port number
- DestinationPort 16 bits for the destination port number
- SequenceNumber 32 bits for the sequence number
- AcknowledgementNumber 32 bits for the acknowledgement number
- DataOffsetAndFlags 16 bits for flags and data offset

- Window 16 bits for the window size
- HeaderLength Header length
- MessageLength Length of the data being carried
- TCPData array of bytes as data carried by the TCP packet

UDP Header class consist of the following variables

- SourcePort 16 bits for the source port number
- DestinationPort 16 bits for the destination port number
- Length Length of the UDP header
- Checksum 16 bits for the checksum
 - (checksum can be negative so taken as short)
- UDPData array of bytes as data carried by the UDP

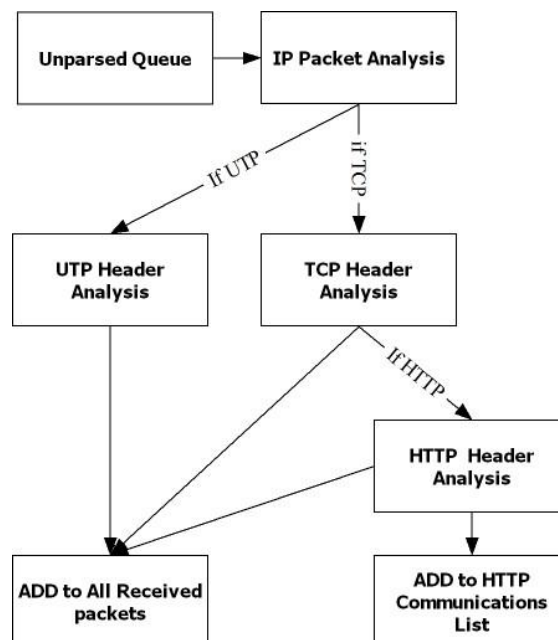


Figure 5.4 parsing module

The step of this module as the following

- Create parsing thread
- Loop on unparsed packet if any packet was received
- Dequeue the packet from the queue
- Apply IP header analysis
- Check transfer protocol
 - If the protocol is TCP apply TCP header analysis
 - If protocol is HTTP port (80 or 8080)
 - Apply HTTP header analysis
 - Add the connection information to the HTTP communication list
 - If the protocol is UTP apply UTP header analysis
- Add packet to all received packet list

5.2.4 List ALL Recorded packet

- This list is container that store all packets after parsing
- Only the parsing thread is writing on this list
- Filter 1 module is using it to filter the output view on the mainframe form
- The file extractor thread is reading from this list in order to get all related packets to the files

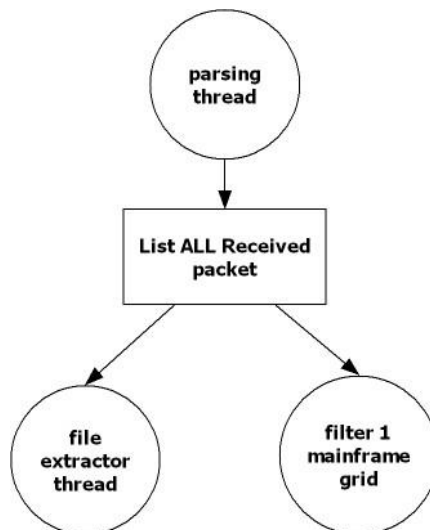


Figure 5.5 List ALL Recorded packet data flow

Figure 5.5 shows the data flow to the list of all recorded packets and shows that parsing thread writing the data to it only so we do not need mutual exclusive to be applied to it and many thread (file extractor and filter1) read from it.

5.2.5 Filter 1 and view data

In the normal way the parsing data is stored directly to the all packet received packet and another copy is sent to filter 1, and according to filter1 parameters the packet is added to the grid or not as shown in figure 5.6

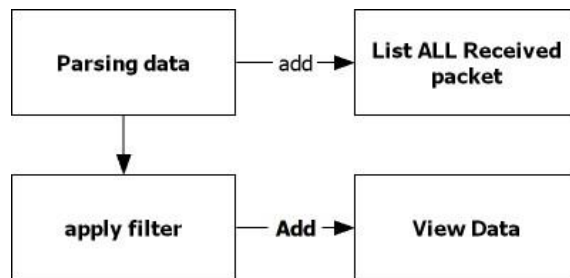


Figure 5.6 filter1 and view data normal way

When the filter 1 parameters is updated I clear the view grid first the apply new query depend on the new parameters to the all received packets as shown in figure 5.7

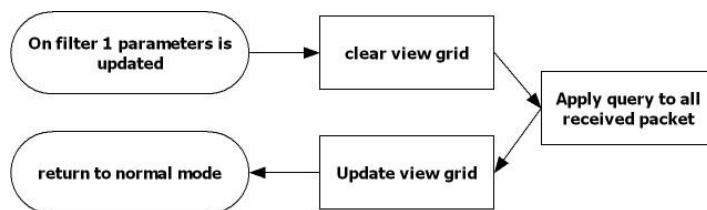


Figure 5.7 Filter 1 on update parameters data flow

As shows in figure 5.8 the filter parameters can be one or more of the following

- Source IP
- Destination IP (target IP)
- Source port
- Destination port (target port)
- The protocol (TCP,UDP,DNS,HTTP,HTTPS)
- The direction of the data (upload, download)

The screenshot shows a window titled "FilterForm". It has a standard Windows-style title bar with minimize, maximize, and close buttons. The main area contains several input fields: "Source IP" and "Target IP" each followed by a dotted box for IP address entry; "Source Port" and "Target Port" each followed by a text box for port entry; and a "Direction" dropdown menu. To the right, under the heading "Protocols", there is a list of checkboxes for "TCP", "UDP", "HTTP", "HTTPS", and "DNS". At the bottom of the window, there are three buttons: "Filter" with a magnifying glass icon, "Clear" with a broom icon, and "Close" with a red power button icon.

Figure 5.8 filter form for filter 1 parameters

5.2.6 HTTP Communications List

This is a list that stores all communications between all uses and the hosts, I store both the request parameters and response one for any HTTP communication which will help me in analysis HTTP contents and files extraction

HTTP Analysis Data consist of the following variables

| The Request data | |
|--------------------|---|
| RequestPacketIndex | index of the packet at all received packet list |
| Req_HTTPMethod | Requests a web application override the method specified in the request (typically POST) with the method given in the header field (typically PUT or DELETE). Can be used when a user agent or firewall prevents PUT or DELETE methods from being sent directly |
| Req_RequiredURL | The requested URL |

| | |
|---------------------|--|
| Req_HTTPVersion | Used HTTP Version |
| Req_Host | The domain name of the server (for virtual hosting), and the TCP port number on which the server is listening "Host: en.wikipedia.org:80" |
| Req_UserAgent | The user agent string of the user agent "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:12.0) Gecko/20100101 Firefox/21.0" |
| Req_Accept | Content-Types that are acceptable for the response "Accept: text/plain" |
| Req_AcceptLanguage | List of acceptable human languages for response "Accept-Language: en-US" |
| Req_AcceptEncoding | List of acceptable encodings "Accept-Encoding: gzip, deflate" |
| Req_Cookie | An HTTP cookie previously sent by the server "Cookie: \$Version=1; Skin=new;" |
| Req_Connection | Control options for the current connection and list of hop-by-hop request fields "Connection: keep-alive" |
| Req_Pragma | Implementation-specific fields that may have various effects anywhere along the request-response chain. "Pragma: no-cache" |
| Req_CacheControl | Tells all caching mechanisms from server to client whether they may cache this object. It is measured in seconds "Cache-Control: max-age=3600" |
| The response data | |
| ResponsePacketIndex | index of the packet at all received packet list |
| Res_StatusCode | CGI header field specifying the status of the HTTP response "Status: 200 OK" |
| Res_Date | Acceptable version in time "Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT" |
| Res_Server | A name for the server "Server: Apache/2.4.1 (Unix)" |
| Res_LastModified | The last modified date for the requested object "Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT" |
| Res_ETag | An identifier for a specific version of a resource "ETag: "737060cd8c284d8af7ad3082f209582d"" |
| Res_AcceptRanges | What partial content range types this server supports "Accept-Ranges: bytes" |
| Res_Connection | Control options for the current connection and list of hop-by-hop request fields "Connection: keep-alive" |

| | |
|---------------------|--|
| Res_ContentType | The MIME type of the body of the request (used with POST and PUT requests) "Content-Type: application/x-www-form-urlencoded" |
| Res_ContentEncoding | The type of encoding used on the data "Content-Encoding: gzip" |
| Res_ContentLength | The length of the request body in octets (8-bit bytes) "Content-Length: 348" |

Note that I just store the information of the communications but the related packets will stores at the all received packet list

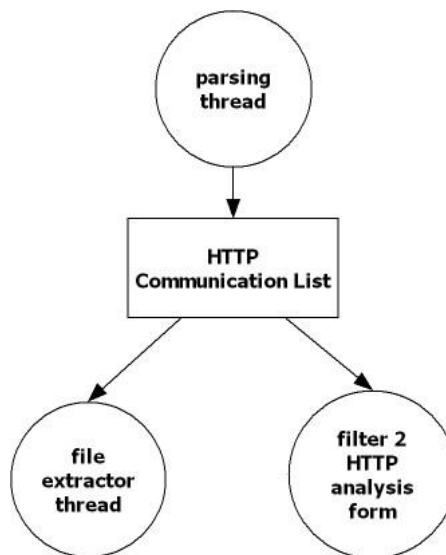


Figure 5.9 HTTP Communication List data flow

Figure 5.9 shows the data flow to the HTTP Communication List and shows that parsing thread writing the data to it only so we do not need mutual exclusive to be applied to it and many thread (file extractor and filter 2) read from it.

5.2.7 Filter 2 and HTTP Analysis Data View

In the normal way the parsing data is stored directly to the HTTP communication list and another copy is sent to filter 2, and according to filter2 parameters the packet is added to the grid or not as shown in figure 5.10

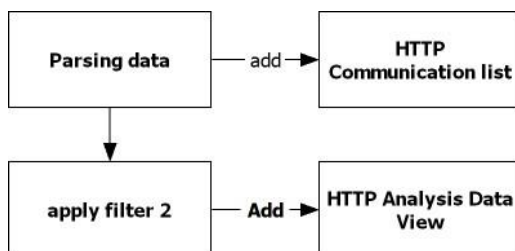


Figure 5.10 filter 2 and HTTP Analysis Data View normal way

When the filter 2 parameters is updated I clear the HTTP Analysis Data View first the apply new query depend on the new parameters to the HTTP communication List as shown in figure 5.11

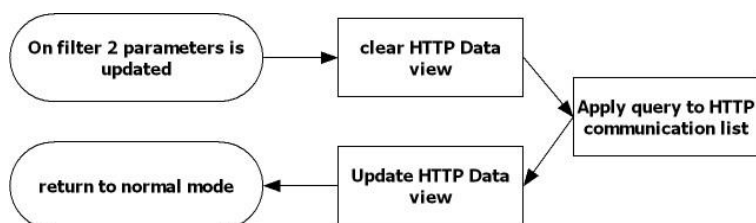


Figure 5.11 Filter 2 on update parameters HTTP Analysis Data View

As shows in figure 5.12 the filter parameters can be one or more of the following

- User IP
- Host (web site name)
- Content size (Grater than / less than)
- HTTP Methods (GET, POST, HEAD, PUT, ...)
- Content type (HTML, PHP, CSS, JS, GIF, JPG,PNG, ICO, ...)
- HTTP Response (SUCCESSFUL, REDIRECTION, CLIENT ERROR , SERVER ERROR)

Figure 5.12 filter form for filter 2 parameters

5.2.8 Extract Files From packets

This module is a very important model which give the ability to restore the files which users request from the pure packets.

I have supported three type of file extractor form packets

- Extract one file
- Extract all files requested by selected user
- Extract all files requested to selected host

The steps to restore one file from packets as the following

- Select the file needed from the HTTP communications list
- Apply a query to all received packets list which have AcknowledgementNumber equal to the AcknowledgementNumber to the selected communication item
- Sort all packet by SequenceNumber
- Make a file at specified path and extract the file name and extension from the HTTP communication data
- Loop on the packets and collect all file bytes in memory stream then write it to the opened file
- If the packet has PSH flag means this is the last packet so close the file
- If the ContentEncoding = gzip then decompress the file

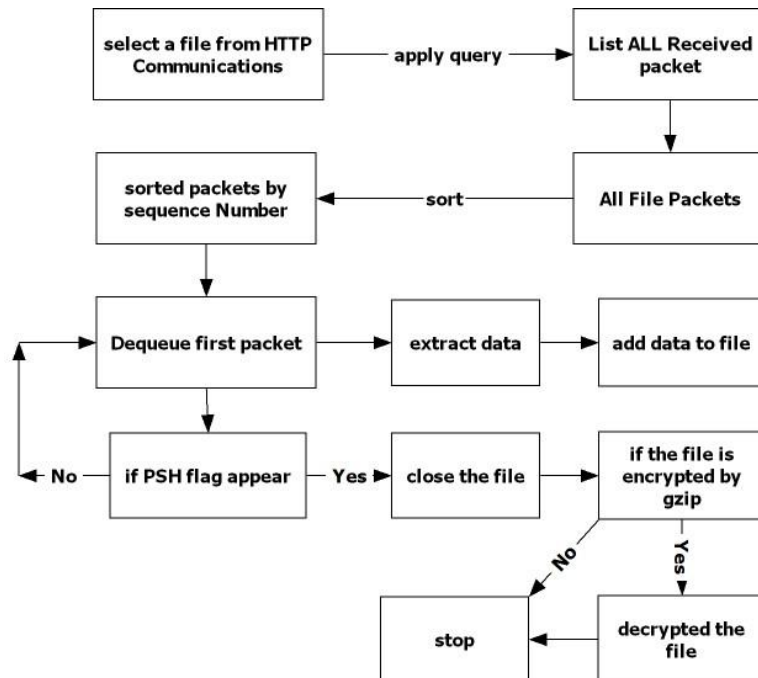


Figure 5.13 extraction file from received packets

Figure 5.13 shows the steps of extraction any file by choosing it from HTTP communications list

The steps to restore all files from the packets by selected user as the following

- Apply query to all communication list requested by the selected user
- Loop one by one and extract the file
- Store all files at the selected destination folder

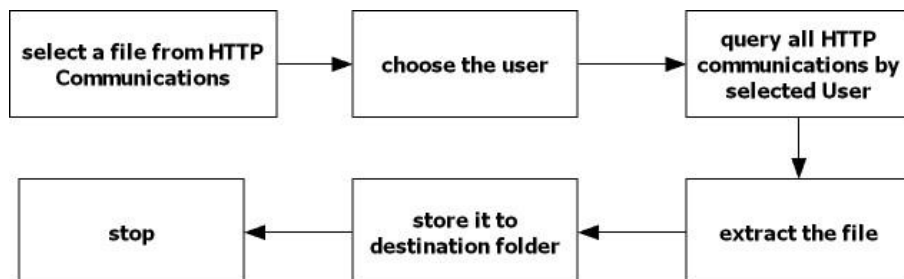


Figure 5.14 extract file by user

Figure 5.14 shows the steps of extraction the files requested by selected user by choosing it from HTTP communications list

The steps to restore all files from the packets by selected host as the following

- Apply query to all communication list requested to the selected host
- Loop one by one and extract the file
- Store all files at the selected destination folder

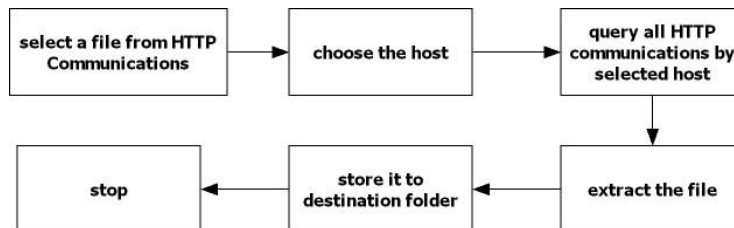


Figure 5.15 extract file by host

Figure 5.15 shows the steps of extraction the files requested to selected host by choosing it from HTTP communications list

5.3 Active thread in the application

- Listing thread
- Parsing
- All received data grid (filter 1)
- All HTTP communications grid (filter 2)
- file extractor
- Main frame form
- HTTP analysis form

Chapter 6

Testing and analysis

6.1 introduction

In this chapter I will examine my application and show the result of all expected behavior of the user, I will examine it on multi user, multi destination, and multi-protocol showing the result data

6.2 main form

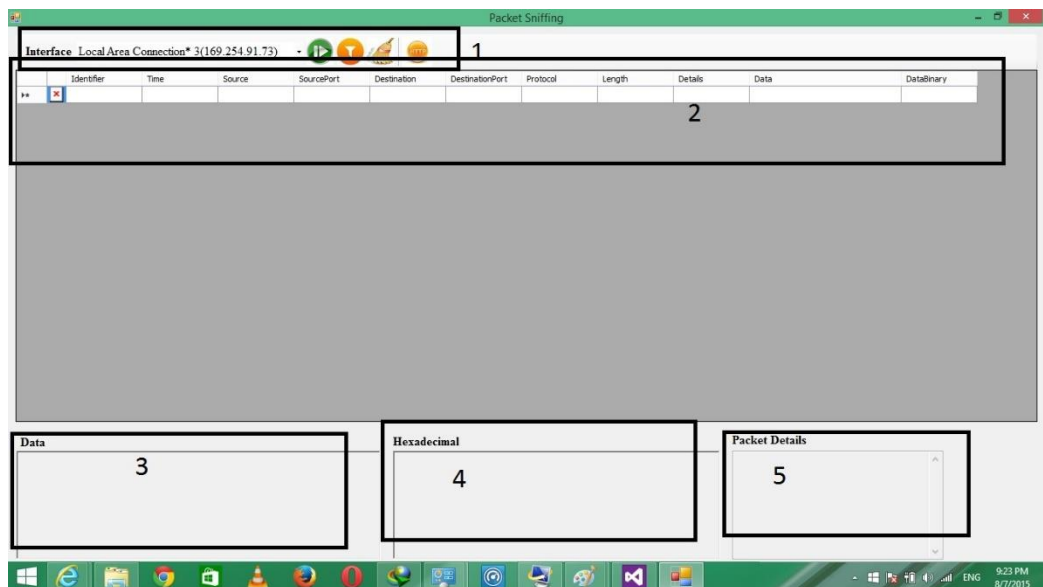


Figure 6.1 Main form application

Figure 6.1 shows the application just started and it consist of 5 main parts

1. toolbar and combo for interface select
2. All received packet grid and filtered data
3. Data of the selected packet
4. Hexadecimal value of the selected
5. Packets details

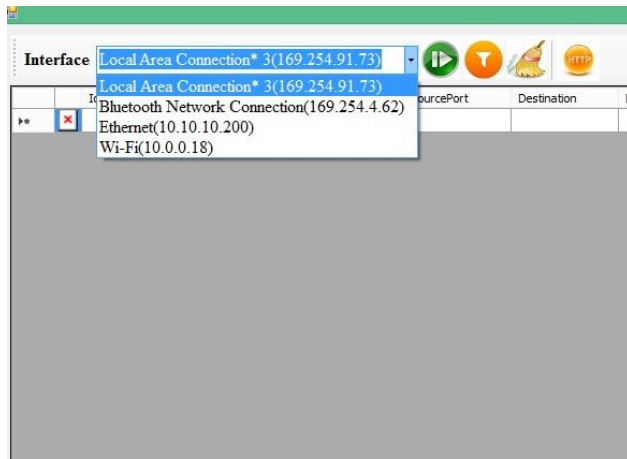


Figure 6.2 Interface selection combo box

Figure 6.2 shows the interface selection combo box which list all network interfaces defined at this machine and its IP address

This is the first step have to be done before start listening and sniffing data. After choosing the interface the user have to press start button which is the green one

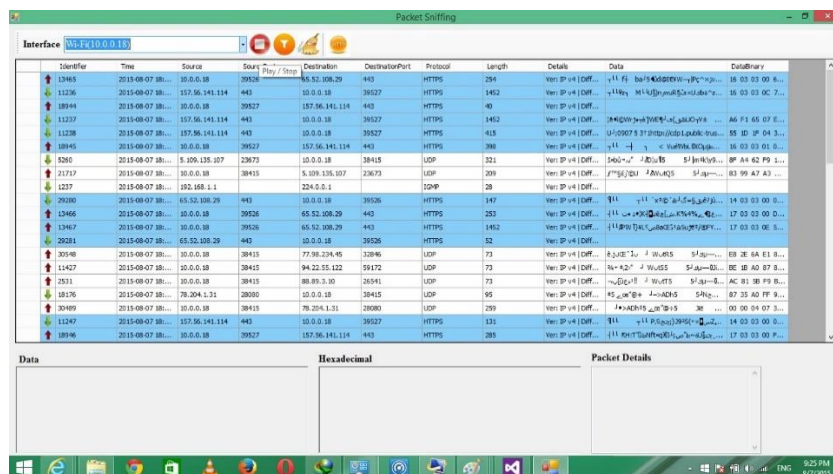


Figure 6.3 The result just after press start button

Just after pressing the start button the data will be appears and added to the all packet received grid as shown in figure 6.3

When the user chose any record from the grid the details will be updated to data, hexadecimal and packet detail as shown in figures (6.4, 6.5, 6.6, 6.7)

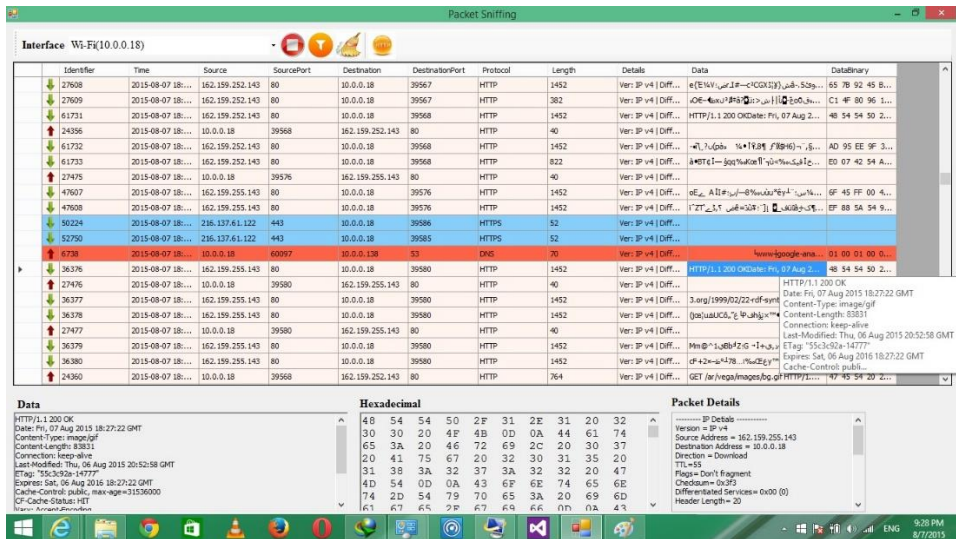


Figure 6.4 The details of packet after user select item from grid

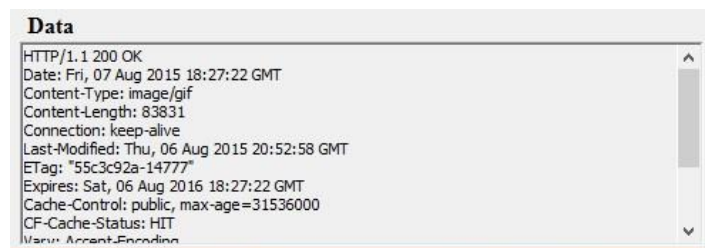


Figure 6.5 The Data of the packet

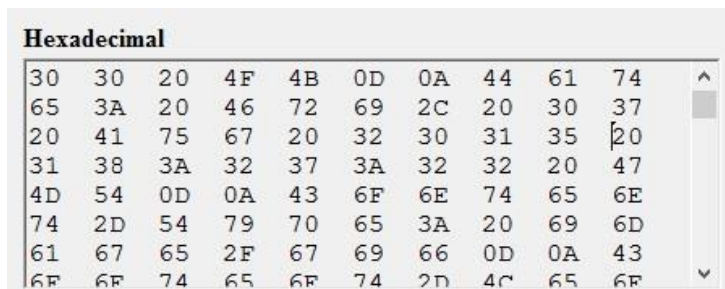


Figure 6.6 The Hexadecimal of the Data

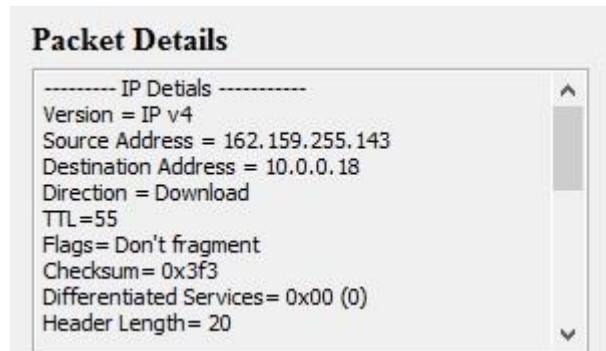


Figure 6.7 The Packet Detail

| | | | |
|----------------------|---------------------------------------|------------------|--|
| Ver: IP v4 Diff... | | | |
| Ver: IP v4 Diff... | | | |
| Ver: IP v4 Diff... | www.google-ana... | 01 00 01 00 0... | |
| Ver: IP v4 Diff... | HTTP/1.1 200 OKDate: Fri, 07 Aug 2... | 48 54 54 50 2... | |
| Ver: IP v4 Diff... | | | HTTP/1.1 200 OK |
| Ver: IP v4 Diff... | 3.org/1999/02/22-rdf-synt | | Date: Fri, 07 Aug 2015 18:27:22 GMT |
| Ver: IP v4 Diff... | (joe;uħUC6,"ğ Pıshğıx™ | | Content-Type: image/gif |
| Ver: IP v4 Diff... | | | Content-Length: 83831 |
| Ver: IP v4 Diff... | | | Connection: keep-alive |
| Ver: IP v4 Diff... | Mm@^1ıBb²Z:G →İ+ı,ı | | Last-Modified: Thu, 06 Aug 2015 20:52:58 GMT |
| Ver: IP v4 Diff... | cF+2ı-ı°ı78...ı%oCEğı™ | | ETag: "55c3c92a-14777" |
| Ver: IP v4 Diff... | | | Expires: Sat, 06 Aug 2016 18:27:22 GMT |
| Ver: IP v4 Diff... | GET /ar/vega/images/bg.gif | HTTP/1.1... | Cache-Control: publi... |


Figure 6.8 Packet tooltip

Figure 6.8 shows the tool tip of the packet when the mouse comes over the data field of packet item at the grid

Other operations

- The user can stop sniffing data by pressing the stop button
- The user can clear the packet grid by pressing clear button
- The user can filter the view of the grid by using the filter form as will shown

6.3 packets filter



FilterForm

Source IP: 10 . 0 . 0 . 18

Source Port:

Target IP: . . .

Target Port:

Direction: ALL

Protocols:

- ☐ TCP
- ☐ UDP
- ☐ HTTP
- ☐ HTTPS
- ☐ DNS

Filter Clear Close

Figure 6.9 Packet filter form

Figure 6.9 Shows the option of the filter supported by the application as the following

- Source IP
- Destination IP (target IP)
- Source port
- Destination port (target port)
- The protocol (TCP,UDP,DNS,HTTP,HTTPS)
- The direction of the data (upload, download)

Interface Wi-Fi (10.0.0.18)

Packet Sniffing

| Id | Filter | Time | Source | Source Port | Destination | Destination Port | Protocol | Length | Details | Data | Duration |
|-------|--------|-------------------------|---------|----------------|-------------|------------------|----------|--------|---|------|-------------------|
| 13665 | | 2018-08-07 08:10:00.000 | 193.506 | 63.52.108.29 | 443 | HTTPS | 294 | 40 | Ver: v3 DF: 0 Seq: 11517628 Window: 0 Len: 40 | | 18.03 03 00 00... |
| 13694 | | 2018-08-07 08:10:00.000 | 193.506 | 157.56.141.114 | 443 | HTTPS | 40 | 40 | Ver: v3 DF: 0 Seq: 11517629 Window: 0 Len: 40 | | 18.03 03 00 00... |
| 13695 | | 2018-08-07 08:10:00.000 | 193.506 | 157.56.141.114 | 443 | HTTPS | 396 | 40 | Ver: v3 DF: 0 Seq: 11517630 Window: 0 Len: 396 | | 18.03 03 00 00... |
| 21217 | | 2018-08-07 08:10:00.000 | 184.15 | 5.109.135.107 | 23873 | UDP | 269 | 40 | Ver: v3 DF: 0 Seq: 11517631 Window: 0 Len: 269 | | 18.03 03 01 00... |
| 13666 | | 2018-08-07 08:10:00.000 | 193.506 | 63.52.108.29 | 443 | HTTPS | 281 | 40 | Ver: v3 DF: 0 Seq: 11517632 Window: 0 Len: 281 | | 17.03 03 00 00... |
| 13667 | | 2018-08-07 08:10:00.000 | 193.506 | 63.52.108.29 | 443 | HTTPS | 1462 | 40 | Ver: v3 DF: 0 Seq: 11517633 Window: 0 Len: 1462 | | 17.03 03 00 00... |
| 15048 | | 2018-08-07 08:10:00.000 | 184.15 | 77.08.234.46 | 32896 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517634 Window: 0 Len: 73 | | 18.03 04 00 00... |
| 11427 | | 2018-08-07 08:10:00.000 | 184.15 | 94.22.55.122 | 59172 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517635 Window: 0 Len: 73 | | 18.03 04 00 00... |
| 2531 | | 2018-08-07 08:10:00.000 | 184.15 | 88.89.10.10 | 26541 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517636 Window: 0 Len: 73 | | 18.03 04 00 00... |
| 5049 | | 2018-08-07 08:10:00.000 | 184.15 | 78.204.31.1 | 26800 | UDP | 259 | 40 | Ver: v3 DF: 0 Seq: 11517637 Window: 0 Len: 259 | | 18.03 04 00 00... |
| 13696 | | 2018-08-07 08:10:00.000 | 193.506 | 157.56.141.114 | 443 | HTTPS | 265 | 40 | Ver: v3 DF: 0 Seq: 11517638 Window: 0 Len: 265 | | 17.03 03 01 00... |
| 13697 | | 2018-08-07 08:10:00.000 | 193.506 | 157.56.141.114 | 443 | HTTPS | 413 | 40 | Ver: v3 DF: 0 Seq: 11517639 Window: 0 Len: 413 | | 17.03 03 01 00... |
| 13668 | | 2018-08-07 08:10:00.000 | 193.506 | 63.52.108.29 | 443 | HTTPS | 1462 | 40 | Ver: v3 DF: 0 Seq: 11517640 Window: 0 Len: 1462 | | 17.03 03 01 00... |
| 15049 | | 2018-08-07 08:10:00.000 | 184.15 | 94.145.187.239 | 11437 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517641 Window: 0 Len: 73 | | 18.03 04 00 00... |
| 15145 | | 2018-08-07 08:10:00.000 | 184.15 | 80.232.7.15 | 49229 | UDP | 95 | 40 | Ver: v3 DF: 0 Seq: 11517642 Window: 0 Len: 95 | | 18.03 04 00 00... |
| 10084 | | 2018-08-07 08:10:00.000 | 184.15 | 87.232.39.40 | 6885 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517643 Window: 0 Len: 73 | | 18.03 04 00 00... |
| 3066 | | 2018-08-07 08:10:00.000 | 184.15 | 68.45.104.36 | 53000 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517644 Window: 0 Len: 73 | | 18.03 04 00 00... |
| 30551 | | 2018-08-07 08:10:00.000 | 184.15 | 77.98.234.46 | 32896 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517645 Window: 0 Len: 73 | | 18.03 04 00 00... |
| 11430 | | 2018-08-07 08:10:00.000 | 184.15 | 94.22.55.122 | 59172 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517646 Window: 0 Len: 73 | | 18.03 04 00 00... |
| 2534 | | 2018-08-07 08:10:00.000 | 184.15 | 88.89.10.10 | 26541 | UDP | 73 | 40 | Ver: v3 DF: 0 Seq: 11517647 Window: 0 Len: 73 | | 18.03 04 00 00... |

Data

```

HTTP/1.1 300 OK
Date: Fri, 07 Aug 2015 08:27:22 GMT
Content-Type: image/gif
Content-Length: 100
Connection: keep-alive
Last-Modified: Fri, 07 Aug 2015 08:25:58 GMT
ETag: "553c9a-14777"
Cache-Control: max-age=3600
Cache-Control: max-age=31536000
Content-Encoding: gzip
          
```

Hexadecimal

```

48 54 54 50 2F 31 2E 31 20 32 04
30 30 20 4F 4B 0D 0A 44 61 72
65 3A 20 46 72 69 6E 20 30 37
20 41 75 67 20 32 30 31 35 20
31 31 38 3A 32 37 3A 32 32 07
44 54 0D 0A 43 61 62 74 6E
74 21 64 79 70 65 3A 20 65 6E
67 67 65 3A 67 6F 66 0A 47
         
```

Figure 6.10 The main form after filtering by source address 10.0.0.18

Figure 6.10 shows the main form after filtering by source address

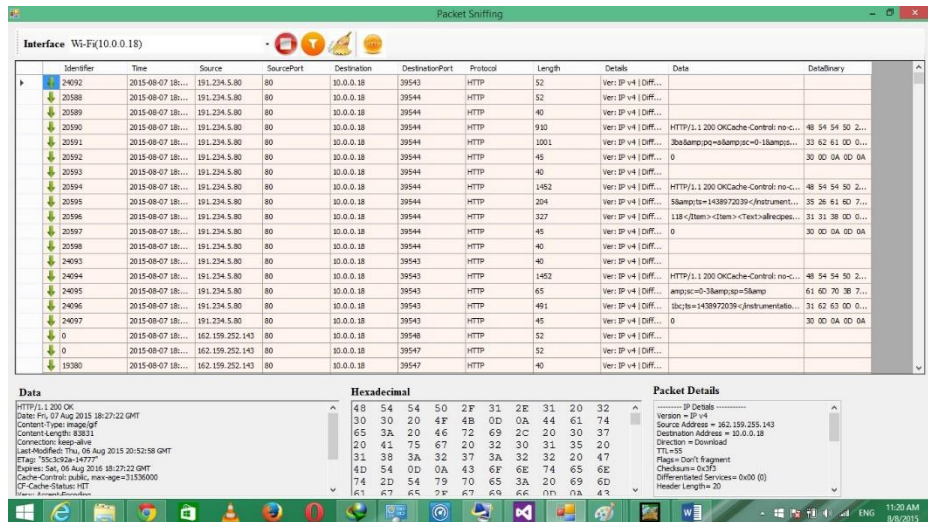


Figure 6.11 The main form after filtering by HTTP protocol and data direction download

Figure 6.11 shows the main form after filtering it by HTTP protocol and data direction download

6.4 HTTP analysis form

By pressing the HTTP analysis icon new form will be appears which is just for HTTP communications

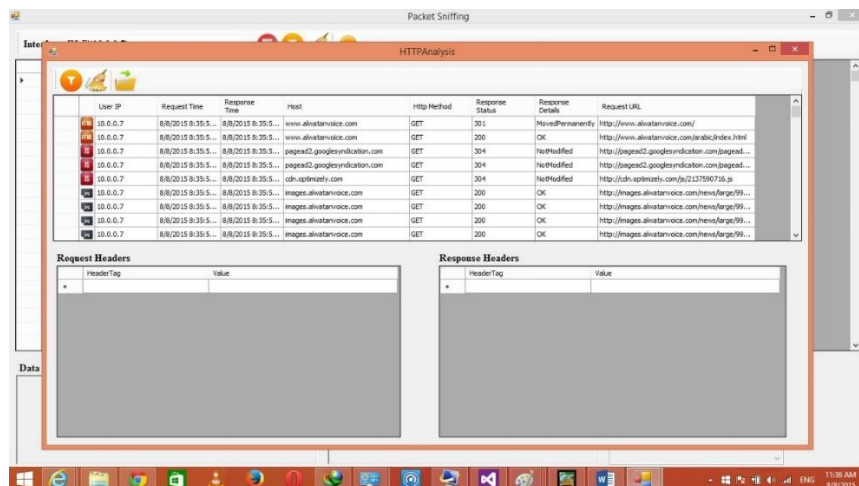


Figure 6.12 HTTP communication grid

Figure 6.12 shows the HTTP analysis form and when the user choose one of the item grid the full detail of the communication will be appears at request header and response header as shown in figures (6.13, 6.14, 6.15)

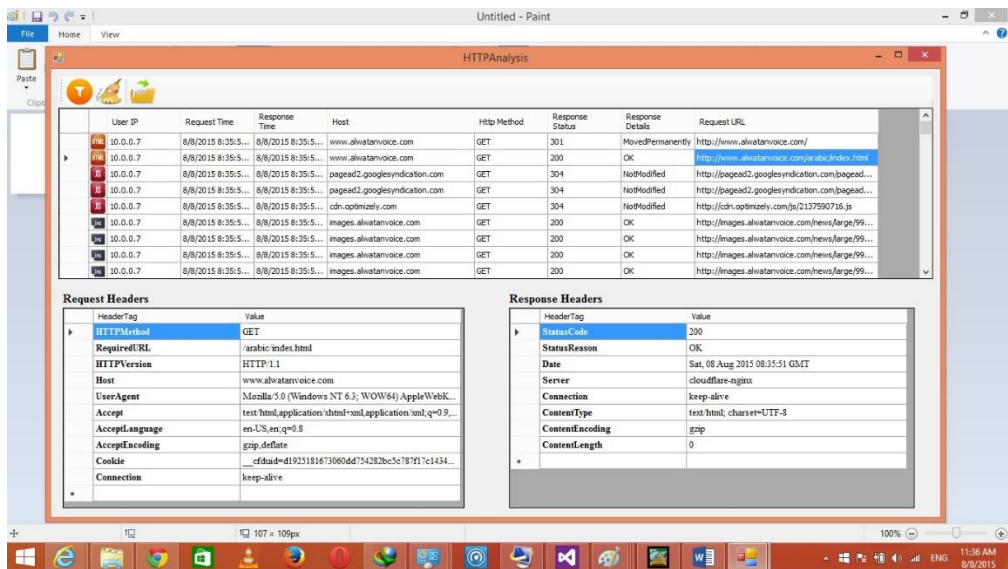


Figure 6.13 HTTP communications form after user choose item

| Request Headers | | |
|-----------------|----------------|---|
| | HeaderTag | Value |
| ▶ | HTTPMethod | GET |
| | RequiredURL | /arabic/index.html |
| | HTTPVersion | HTTP/1.1 |
| | Host | www.alwatanvoice.com |
| | UserAgent | Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebK... |
| | Accept | text/html,application/xhtml+xml,application/xml;q=0.9,... |
| | AcceptLanguage | en-US,en;q=0.8 |
| | AcceptEncoding | gzip,deflate |
| | Cookie | __cfduid=d1925181673060dd754282bc5c787f17c1434... |
| | Connection | keep-alive |
| * | | |

Figure 6.14 Request header of HTTP communication

| Response Headers | |
|------------------|-------------------------------|
| HeaderTag | Value |
| Status Code | 200 |
| Status Reason | OK |
| Date | Sat, 08 Aug 2015 08:35:51 GMT |
| Server | cloudflare-nginx |
| Connection | keep-alive |
| Content Type | text/html; charset=UTF-8 |
| Content Encoding | gzip |
| Content Length | 0 |
| * | |

Figure 6.15 Response header of HTTP communication

- The user may press clear button to clear the grid so all communication will be erased and will still can record all new communications

6.5 HTTP communications filter

HttpAnalysisFilter

User IP:

Host:

Content Size: Greater Tha ▾
Greater Than
Less Than

HTTP Methods

- ☐ GET
- ☐ POST
- ☐ HEAD
- ☐ PUT
- ☐ DELETE
- ☐ OPTIONS
- ☐ CONNECT
- ☐ TRACE

Content Type

- ☐ HTML
- ☐ PHP
- ☐ CSS
- ☐ JS
- ☐ GIF
- ☒ JPG
- ☐ PNG
- ☐ ICO

HTTP Response

- ☒ SUCCESSFUL 2XX
- ☐ REDIRECTION 3XX
- ☐ CLIENT ERROR 4XX
- ☐ SERVER ERROR 5XX

Filter Clear Close

Figure 6.16 HTTP communications filter form

Figure 6.16 shows the option of the filter supported by the application as the following

- User IP
- Host (web site name)
- Content size (Grater than / less than)
- HTTP Methods (GET, POST, HEAD, PUT, ...)
- Content type (HTML, PHP, CSS, JS, GIF, JPG,PNG, ICO, ...)
- HTTP Response (SUCCESSFUL, REDIRECTION, CLIENT ERROR , SERVER ERROR)

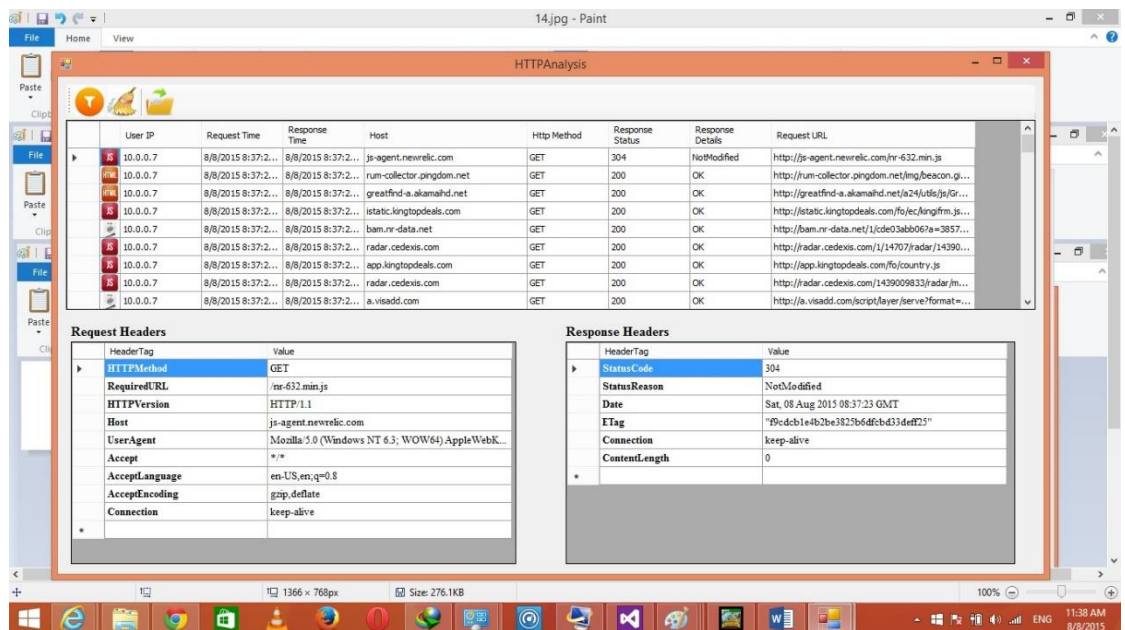


Figure 6.17 The HTTP communication form after HTTP "Response is SUCCESSFUL" filter

Figure 6.17 shows the HTTP communication form after HTTP "Response is SUCCESSFUL" filter applied to the form

6.6 extracting files from packets

When the user press the folder icon from the toolbar new form will be appears as shown in figure 6.18

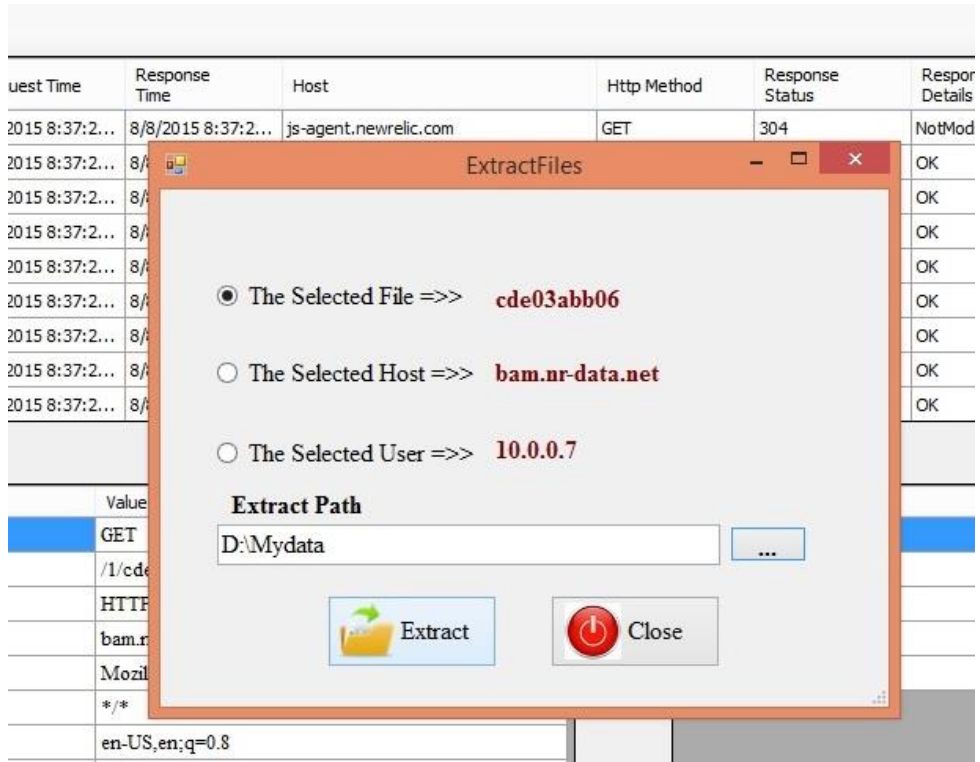


Figure 6.18 Extract files form

Figure 6.18 shows the extract file form and let the user to choose one of three way

- extract the file from the selected item grid and shows the file name and the host name
- extract all files related to the host name of the selected item
- extract all files related to the user of the selected item

After the user select the mode and choose the destination folder he will press extract button and the data will be at the folder selected as shown in figure 6.19

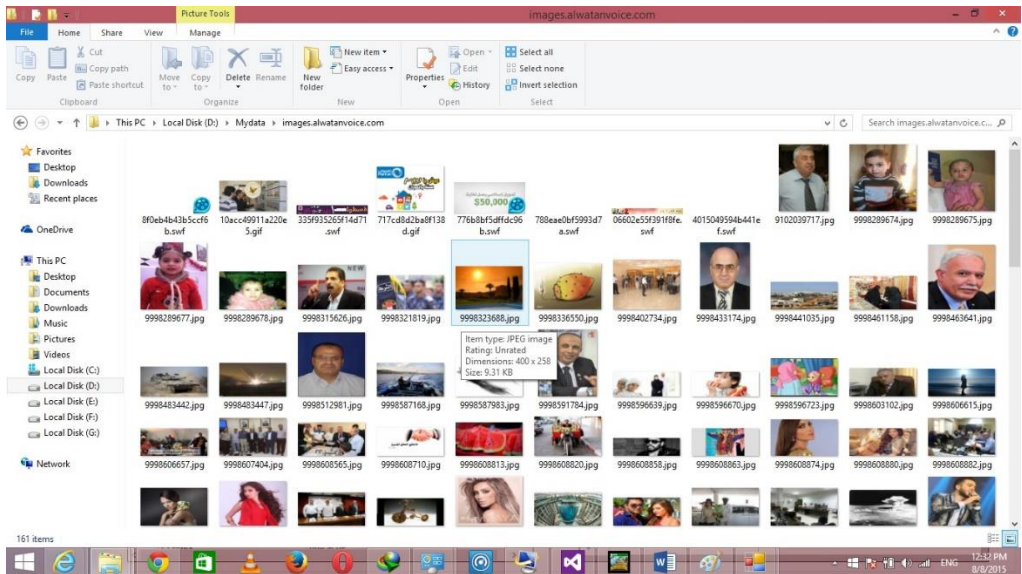


Figure 6.19 The file extracted from the packets

Figure 6.19 shows the folder d:\MY_Data that contain the files extracted from the packets by selecting to extract all files related to the “alwatanvoice.com” host

Chapter 7

Conclusion and future work

7.1 Conclusion

The methods used by network intrusions can be different from one to another. However, the nature of most network intrusions is based on “malicious” network traffic, which either has invalid value inside a field of a packet, or features incorrect combination or sequence of packet segments. With this observation, the network administrator can use my application to monitor and analysis packets over the network and builds network intrusion detection systems.

However, packet filter and data analysis faces new challenges for the intrusion detection purpose, like high-volume data, no packet dropping, and requirement for system flexibility and scalability. In this thesis, I worked on low level packet sniffing and worked on C# which is master on memory management so we comes over this challenges and I borrowed some idea from the adaptive pattern matching technique and applied it to my application module for a large-scale intrusion detection framework.

One of the key components in my approach is the HTTP filer analysis and file extractor form pure packets, this module have great power in monitoring the network in order to follow all files that specific user may download moreover you can do many statistics on these data as most host visited, the contents of most downloaded files, the real traffic of HTTP over all others protocols packets,.... .

7.2 future work

There are many future work appears when I developed my system .The following developed can be done

- In my application I have supported just a selected protocols so one of future work is to support , filter and sniffing all other protocols
- Add more filters to the packets (filter by text content, internet browser used, date and time, ...)
- Add the other protocols analysis (DNS, ICMP, POP3,)
- Extracting the secrets key of HTTPS
- Support the Password sniffing
- decrypting HTTPS data and extract files from it
- Detecting malicious behavior (man in the middle, DNS poisoning, fishing, ...)
- Create alert system to send warnings when malicious behavior accrues
- Add module to edit any received packet and retransmit it

REFERENCES

1. Security Measures in a Secure Computer Communications Architecture Bottino, L.J. ; Fed. Aviation Adm., Atlantic City Int. Airport, NJ 25th Digital Avionics Systems Conference, 2006 IEEE/AIAA
2. Securing IPv6 network infrastructure: A new security model Choudhary, A.R. ; SEGMA Technol. Inc., Silver Spring, MD, USA ; Sekelsky, A. Technologies for Homeland Security (HST), 2010 IEEE International Conference on
3. A malicious activity detection system utilizing predictive modeling in complex environments Almaatouq, A. ; Alabdulkareem, A. ; Nouh, M. ; Alsaleh, M. Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th
4. An approach to detect malicious activities in SCADA systems Pramod, T.C. ; Dept. of Comput. Sci. & Eng., Siddaganga Inst. of Technol., Tumkur, India ; Sunitha, N.R. Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on
5. Evaluation of applicability of modified vector space representation for in-VM malicious activity detection in Cloud Borisaniya, B. ; Comput. Eng. Dept., NIT Surat, Surat, India ; Patel, K. ; Patel, D. India Conference (INDICON), 2014 Annual IEEE
6. MAC aggregation resilient to DoS attacks Kolesnikov, V. ; Bell Labs., Alcatel-Lucent, Murray Hill, NJ, USA ; Wonsuck Lee ; Junhee Hong Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on
7. Understanding complex network attack graphs through clustered adjacency matrices Noel, S. ; Center for Secure Inf. Syst., George Mason Univ., Fairfax, VA ; Jajodia, S. Computer Security Applications Conference, 21st Annual
8. A resource management approach to web browser security Jun Li ; Univ. of Oregon, Eugene, OR, USA ; Dongting Yu ; Maurer, L. Computing, Networking and Communications (ICNC), 2012 International Conference on
9. 2015 Internet Security Threat Report
10. Security threats in cloud computing Shaikh, F.B. ; Dept. of Comput. & Technol., SZABIST, Islamabad, Pakistan ; Haider, S. Internet Technology and Secured Transactions (ICITST), 2011 International Conference for
11. Security assessment of computer networks -an ethical hacker's perspective Rao, G.S. ; Core Design Excellence Group, Tata Consultancy Services, Hyderabad, India ; Naveen Kumar, P. ; Swetha, P. ; BhanuKiran, G. Computer and Communications Technologies (ICCCT), 2014 International Conference on

12. Hijacking spoofing attack and defense strategy based on Internet TCP sessions Yongle Wang ; Xuchang Ploughs the Recent Inf. Sci. Res. Inst., Xuchang, China ; JunZhang Chen Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on
13. IP Spoofing Detection Using Modified Hop Count Mukaddam, A. ; Electr. & Comput. Eng. Dept., American Univ. of Beirut, Beirut, Lebanon ; Elhadj, I. ; Kayssi, A. ; Chehab, A. Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on
14. Intrusion Detection System with packet filtering for IP Spoofing Manusankar, C. ; Dept. of CSE, SNS Coll. of Technol., Coimbatore, India ; Karthik, S. ; Rajendran, T. Communication and Computational Intelligence (INCOCCI), 2010 International Conference on
15. Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in Smart Grid SCADA systems Yang, Y.; McLaughlin, K.; Littler, T.; Sezer, S.; Eul Gyu Im; Yao, Z.Q.; Pranggono, B.; Wang, H.F.Sustainable Power Generation and Supply (SUPERGEN 2012), International Conference on
16. Topology discovery of PROFINET networks using Wireshark Sahin, V.H.; Ozcelik, I.; Balta, M.; Iskefiyeli, M. Electronics, Computer and Computation (ICECCO), 2013 International Conference on
17. Analysis and application of Wireshark in TCP/IP protocol teaching Shaoqiang Wang; DongSheng Xu; ShiLiang Yan E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on
18. Bottleneck Analysis of Traffic Monitoring using Wireshark Dabir, A.; Matrawy, A. Innovations in Information Technology, 2007. IIT '07. 4th International Conference on
19. Wireless network attack: Raising the awareness of Kampung WiFi residents Fahmy, S.; Nasir, A.; Shamsuddin, N. Computer & Information Science (ICCIS), 2012 International Conference on
20. Integrated management for OSI networks Joseph, C.; McFarland, M.; Muralidhar, K.H. Global Telecommunications Conference, 1990, and Exhibition. 'Communications: Connecting the Future', GLOBECOM '90., IEEE
21. A new interoperable management model for IP and OSI architectures Koth, A.M.; El-Sherbini, A.; Kamel, T. AFRICON, 1996., IEEE AFRICON 4th
22. Improving the SRW Waveform via a Physical Layer Retrofit Blyskun, A.; Johnson, M.; Sungill Kim; Speros, J.; Thatte, G.; Williamson, D.R. Military Communications Conference, MILCOM 2013 - 2013 IEEE
23. An architectural framework for data link layer security with security inter-layering Altunbasak, H.; Owen, H. SoutheastCon, 2007. Proceedings. IEEE

24. Economic benefits of optical transport layer reconfigurability and tighter coupling with the service layer Atkinson, G.; Nagarajan, R.; Shaikh, S. Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004, 11th International
25. Study on presentation layer structure of multi-tenant e-business systems Peng Yao; Hao-yu Wen; Qiong-jie Dai Industrial Engineering and Engineering Management (IE&EM), 2011 IEEE 18Th International Conference on
26. Network based packet watermarking using TCP/IP protocol suite Shah, M.K.; Patel, S.B. Engineering (NUICONe), 2011 Nirma University International Conference on
27. Real-Time Adaptive Link Layer Trigger Based Cross Layer Fast Handoff Mechanism in IEEE 802.11 WLANs Xiaoyu Cheng; Duyan Bi Communication Software and Networks, 2009. ICCSN '09. International Conference on
28. Kerberos secured Address Resolution Protocol (KARP) Bakhache, B.; Rostom, R. Digital Information and Communication Technology and its Applications (DICTAP), 2015 Fifth International Conference on
29. Design and realization for Internet of Things Logistic Unified Information System API layer sound recording mode Ying Wei Computer Science and Service System (CSSS), 2011 International Conference on
30. Transport Layer Protocols for Cognitive Networks Sarkar, D.; Narayan, Harendra INFOCOM IEEE Conference on Computer Communications Workshops , 2010
31. The performance evaluation and comparison of TCP-based high-speed transport protocols Zhaojuan Yue; Xiaodan Zhang; Yongmao Ren; Jun Li; Qianli Zhong Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on
32. Enhancement of throughput time using MS-TCP transport layer protocol for 4G mobiles Ghazaleh, H.; Muhanna, M. Systems, Signals and Devices, 2008. IEEE SSD 2008. 5th International Multi-Conference on
33. Network Communication System of Virtual Scene Based on UDP Protocol Zhang Tie-Tou; Tan Ying-Jun Intelligent Systems Design and Engineering Applications (ISDEA), 2014 Fifth International Conference on
34. Implementation and Performance Evaluation of nanoIP Protocols: Simplified Versions of TCP, UDP, HTTP and SLP for Wireless Sensor Networks Jardak, C.; Meshkova, E.; Riihijarvi, J.; Rerkrai, K.; Mahonen, P. Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE
35. A review of security attacks on IEC61850 substation automation system network Rashid, M.T.A.; Yussof, S.; Yusoff, Y.; Ismail, R. Information Technology and Multimedia (ICIMU), 2014 International Conference on

36. Effect of high-frequency high-voltage impulse conditioning on inrush current interruption of vacuum interrupters Yingyao Zhang; He Yang; Yingsan Geng; Zhiyuan Liu; Lijun Jin Dielectrics and Electrical Insulation, IEEE Transactions on
37. Research on collision effects of the interception projectile against the attacking object in active EM armor Wang Chengxue; Xue Luqiang; Sun Xuefeng; Zhu Liangming Electromagnetic Launch Technology (EML), 2012 16th International Symposium on
38. The Research and Analysis of the New Modification Theory of Toroidal Worm-Gearing Wen QingMing; Xu Hua; Tang WeiXiang System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2010 International Conference on
39. Fabrication of micro open structure using 3D laser scanning method in nano-stereolithography Cheol Woo Ha; Dong-Yol Yang Manipulation, Manufacturing and Measurement on the Nanoscale (3M-NANO), 2014 International Conference on
40. Reconnaissance Scan Detection Heuristics to disrupt the pre-attack information gathering Udhayan, J.; Prabu, M.M.; Krishnan, V.A.; Anitha, R. Network and Service Security, 2009. N2S '09. International Conference on
41. Design and Implementation of V6SNIFF: An Efficient IPv6 Packet Sniffer Seong-Yee Phang; HoonJae Lee; Hyotaek Lim Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on
42. Detection and prevention of active sniffing on routing protocol Ramakrishna, P.; Maarof, M.A. Research and Development, 2002. SCORED 2002. Student Conference on
43. Network Traffic Analysis Using Refined Bayesian Reasoning to Detect Flooding and Port Scan Attacks Dai-ping Liu; Ming-wei Zhang; Tao Li Advanced Computer Theory and Engineering, 2008. ICACTE '08. International Conference on
44. Query expansion for intelligent information retrieval on Internet Jae-Hyun Lim; Hyon-Woo Seung; Jun Hwang; Kim, Chang-Il; Heung-Nam Kim Parallel and Distributed Systems, 1997. Proceedings., 1997 International Conference on
45. On the public information embedding capacity region under multiple access attacks Yangfan Zhong; Yadong Wang; Alajaji, F.; Linder, T. Information Theory, 2008. ISIT 2008. IEEE International Symposium on
46. ProcurePass: A User Authentication Protocol to Resist Password Stealing and Password Reuse Attack Kassim, M.M.; Sujitha, A. Computational and Business Intelligence (ISCBI), 2013 International Symposium on
47. Trust and Privacy Exploitation in Online Social Networks Wong, K.; Wong, A.; Yeung, A.; Wei Fan; Su-Kit Tang IT Professional 2014, Volume: 16, Issue: 5

48. On remote exploitation of TCP sender for low-rate flooding denial-of-service attack Kumar, V.A.; Jayalekshmy, P.; Patra, G.K.; Thangavelu, R.P. Communications Letters, IEEE 2009, Volume: 13
49. Detection of stealth Man-in-the-Middle attack in wireless LAN Kumar, V.; Chakraborty, S.; Barbhuiya, F.A.; Nandi, S. Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on
50. DoS and DDoS Attack's Possibility Verification on Streaming Media Application Chen Lei; Ye Dejian Information Science and Engineering, 2008. ISISE '08. International Symposium on
51. Filtering spoofed traffic at source end for defending against DoS / DDoS attacks Malliga, S.; Tamilarasi, A.; Janani, M. Computing, Communication and Networking, 2008. ICCCN 2008. International Conference on