VoIP Forensic Analyzer

M Mohemmed Sha
Dept. of Computer Science & Information
Prince Sattam Bin Abdulaziz University
P O Box.54, Saudi Arabia.Pin:11991

Manesh T
Dept. of Computer Science & Information
Prince Sattam Bin Abdulaziz University
P O Box.54, Saudi Arabia.Pin:11991

Saied M. Abd El-atty*
Dept. of Computer Science & Information
Prince Sattam Bin Abdulaziz University
P O Box.54, Saudi Arabia.Pin:11991
*Faculty of Electronic Engineering,
Menoufia University, 32952 Menouf,
Egypt

***Abstract*—People have been utilizing Voice over Internet Protocol (VoIP) in most of the conventional communication facilities which has been of assistance in the enormous attenuation of operating costs, as well as the promotion of next-generation communication services-based IP. As an intimidating upshot, cyber criminals have correspondingly started interjecting the environment and creating new challenges for the law enforcement system in any Country. This paper presents an idea of a framework for the forensic analysis of the VoIP traffic over the network. This forensic activity includes spotting and scrutinizing the network patterns of VoIP-SIP stream, which is used to initiate a session for the communication, and regenerate the content from VoIP-RTP stream, which is employed to convey the data. Proposed network forensic investigation framework also accentuates on developing an efficient packet restructuring algorithm for tracing the depraved users involved in a conversation. Network forensics is the basis of proposed work, and performs packet level surveillance of VoIP followed by reconstruction of original malicious content or network session between users for their prosecution in the court.**

Keywords—Forensics; Packet Reordering; Session Initiation; Real Time Transfer

I. Introduction

Network forensics is a technique of identifying network anomalies and breaches from the pattern of packets based-network. To understand network usage, a content level analysis of individual traffic flow must be conducted to retrieve the behavior and interest pattern of any intruder from the information stored in the networks as packets. This activity is entitled retrospective network analysis or network forensics [1].

During the misuse of any network activity, network forensic process accesses forensic data through an offline packet level inspection. An identical forensic framework captures network session for tracing the malicious contents involved in the particular network usage. It also grabs down complete identity credentials to prosecute the culprits in the court of law. Currently, a majority of the online telephony is through VoIP mechanism, which is of less operational cost and great flexibility. VoIP applications have grown-up in popularity in the recent years because they facilitate free voice and video chat and connect an establishment between VoIP and Public Switched Telephone Network (PSTN).

A. Challenges of VoIP Forensic Frameworks

In VoIP telephony scenario, a VoIP client uses port randomization for sending actual voice through RTP communication. As a concern, obtainable frameworks for forensic processing fall short in acquiring the imperative forensic details and reinstating the original data to track down the cyber intruders that abuse VoIP software infrastructure to transact their detestable contents. Unlimited use of VoIP telephony consumes most of the network bandwidth in organization and industries, which tend network towards jam in it also the upsurge of VoIP frameworks, challenge barely the respective country's judiciary system, as there is no single assembly point resembling public switch telephone network for centralized control and monitoring.

B. Our Contribution

This paper mainly outlines the idea behind a structural framework developed for forensic process and investigation of VoIP along with GUI design, which produced thriving forensic outcomes in reconstructing the original voice in VoIP conversation. Details of any immoral users and their activities has been traced from these conversation sessions.

This paper also deliberates a specially designed time stamped VoIP packet rescheduling strategy for procuring malevolent packet contents from the respective network pattern. Projected structural framework hosts a fully-fledged high-speed packet-bagging module of its own, which work independently or can associate with third party high-speed packet bagging software and hardware. Following is the order of the paper: Section II bonds with literature review, Section III labels component overview of VoIP, Section IV gives an idea of the working of VoIP, Section V provides workflow of the forensic analyzer, and Section VI outlines the results and discussions, followed by conclusion and references.

II. Literature Review

There are scores of apposite forensic processing tools to analyze, dissect and investigate network packets or network streams. Former such works do not support reconstruction of actual voice data in VoIP connection or rather discussed and analyzed VoIP environments for tracing forensic details. This section includes brief outline of various approaches, which fall into forensic analysis of VoIP streams and stimulated us to develop VoIP forensic processing and investigative strategies

Prince Sattam bin Abdulaziz University, Kingdom of Saudi Arabia subsequently a comprehensive study. Soundly most of these methods are complex and shareware with restricted functions.

François, J et al. (2010) presented a technique to identify smart devices from a VoIP stream of traffic through a fingerprint process. They were attentive in signaling plane and castoff voice information. This work helped us to study the network pattern of VoIP sessions [2].

Ibrahim, M, et al. (2010) proposed a structural architecture of examining attacks using VoIP. The architecture reinforces information gathering against a possible attack situation. This facilitated us in designing an efficient packet reconstruction algorithm for VoIP sessions [3].

Hofbauer, S,et al. (2012) offered a strategic approach for maintaining communication details and trace VoIP attacks by conserving caller information. This drove us into tracing more forensic information from VoIP sessions [4].

Gao Hongtao(2011) put for a method that forwards VoIP crime analysison a host computer. His work aided us in designing a proposed forensic method of forensic analyzer for tracing malevolent VoIP usage [5].

Azab, A et al.(2012)open component analysis of skype with respect to call progress. This assisted us to recognize call processing in Skype VoIP environments [6].

Y.Q. Wang (Wang et al., 2011) enlightened an in depth computer forensic skills in dealing with the criminal activities in both wireless and wired communication networks. They emphasize on the credibility, the depth, the extensiveness and the legality of the computer forensic activities [7].

Khidir M. Ali (2012) stated various Practices and Managerial Implications in the field of digital forensics which gave the summary of forensic computing and deliberated key issues to be measured in forensic investigations and digital evidence analysis practices [8].

Edewede Oriwoh et al. (2013) described a technique for Forensics Edge Management framework that autonomously provided security and forensic services within the home Internet of Things [9].

III. Component Overview of VoIP

The foremost signaling ingredient of VoIP infrastructure is Session Initiation Protocol (SIP), which was developed by Internet Engineering Task Force that allows end devices to begin and conclude network communication sessions. The servers and user agents are two significant components of SIP architecture. The User Agent Client (UAC) and User Agent Server (UAS), which respectively generate request and responses, represent a subsystem for user agents. SIP servers are not necessarily separate physical devices but different entities or distinct functions [15]. Four of these server functions are defined as follows.

A. Sip Servers

Proxy servers: They are truly hosts in networks and act as intermediary for clients to initiate request for other clients or hosts in the same network and route SIP generated requests towards UAS, and route SIP generated responses to UAC.

Redirect servers: They act as logical hosts by which client are guided towards a set of alternative Uniform Resource Indicators to accomplish a particular task.

Registrar servers: They accept and process registration based communication messages so that location of end users can be traced.

Location servers: They link address and respective network domain by providing a conceptual database of addresses. Location servers work in parallel with the registration servers to find out alternative set of URIs for a request when generated [15].

B. Sip Messages

Messages of SIP have two defined structures such as requests from client to server and responses from server to clients indicating the status of request. Different types of request messages used by SIP are briefly sketched below.

INVITE: This message is used to invite clients or users to be a part of specific communication session. The content of this message gives description of specific communication session.

ACK: the client uses this message with INVITE messages to indicate confirmation of reception of response in a specific communication session.

OPTIONS: This message is to request for capabilities of the server.

BYE: This message is made sent when a particular user wants to quit a specific communication session.

CANCEL: This message is used to abandon unsettled request

REGISTER: a user or client to start specific communication session uses this message.

RTP: This Real Time Transfer Protocol establishes end-to-end transmission of data such as voice, video over services of network using unicast or multicast strategies. In the case of VoIP, the ports for the RTP communication are decided by the initial handshake done on the SIP.

SDP: The format of initialization parameters for real time transfer is given by this Session Description Protocol and is attached in the SIP message body [15].

IV. Working of VoIP

The functioning of VoIP using SIP and RTP can be worked out into 3 steps. First is the connection establishment using SIP followed by information transference using RTP and connection termination using SIP. To establish a communication as in Fig.1, client1 sends an invite request to the server to be sent to client 2. The server sends an invite request to the client 2; concurrently it sends a trying message to client 1. Once the server gets a ringing response from client 2 it will forward the same to client 1. When client 2 sends the consent it will exchange OK message between server and client1. Now client1 acknowledges the established connection by sending an ACK signal to the server and the server will send that ACK to the client2. Now the connection has been instituted. During this handshake process the invite message and the OK status message contains the SDP in the SIP body, which will provide information about the ports, and other parameters required for the RTP to perform multimedia streaming. Once the call is concluded the bye request is exchanged between client and server in subsequent steps. The second client will respond with an OK message and the connection will be terminated [3].

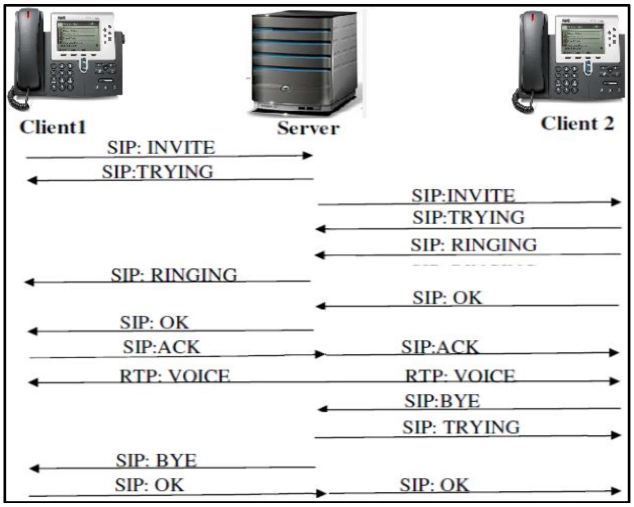


Fig. 1. Working of VoIP handshake

V. Architectural Work Flow of VoIP Forensic Analyser

The architecture of proposed VoIP forensic analyzer is shown in Fig. 2, which executes offline forensic analysis and recreates the absolute voice data between the VoIP clients. The subjected VoIP stream will be seized and observed when a particular misuse of network environment reported. The bagged network session is then undergone through an assortment of analysis and forensic processing phases by the projected architecture to trace proofs against malevolent communications. The ultimate goal is to supply adequate indications to let the offender to be impeached by respective judiciary system. This chief architectural framework hosts following major modules.

A. PCAP File Crafting Unit

This unit is used to bag the network packets though LAN, Wi-Fi interface of standalone computers or proxy servers in LAN, whichever is pertinent in network environment to create a PCAP file. Network packets are enclosed in a PCAP file with extension ".pcap". The "PCAP" file is the foremost input for projected architecture where it undergoes succession of modification throughout the forensic processes [14].

B. VoIP Session Recreating Unit

This unit is the spotlight of forensic processing by way of recreating the genuine network stream from VoIP clients. This module rebuilds the existent voice from VoIP conversation. The foremost activities accomplished at various sub stages are described below.

1) SIP Filter

SIP packets are filtered to scrutinize the initial handshake done between the client and the server to mine forensic details about VoIP initialization constraints. The default port of the SIP protocol is 5060. The filtering is done in accordance with this port. Once filtered the SIP packets from the source PCAP file is separated and saved as temporary PCAP file and is fed to the next stage.

2) Call Identifier

In this sub section, the analyzer will try to identify IP addresses of those clients who have made a call with another client in that network. From the section IV, it is seen that when the connection is established between the two clients an ACK signal is send from the client who initiates the call to the server and from there to the other client. Hence, our framework is designed in such a way that it parses the SIP packets for those IP's that send an ACK signal or receive an ACK signal (other than the server). By this strategy, VoIP analyzer collects the IP's and other forensic information of those clients who have participated in a particular VoIP conversation.

3) Port Identifier

There is no default port used for RTP traffic in any VoIP communication. To identify the port that is used for a specific communication, the initial handshake made for that communication is analyzed and separated followed by extraction of the ports.

4) Message Parser

This component traces SDP messages, which consist of INVITE and STATUS messages by filtering and identifying necessary VoIP packets through all available ports identified by port identifier. Parsed sample format of these messages are shown below.

5) INVITE message sample format.

```
INVITE sip:1006@10.100.13.139:5060 SIP/2.0

Via: SIP/2.0/UDP 10.100.12.230:56727;
branch=z9hG4bK-d8754z-ae78c05f8f58042a-1--
-d8754z-;rport
Max-Forwards: 70
Contact: <sip:1004@10.100.12.230:56727;
rinstance=a8c18539cb50ec97>
To: <sip:1006@10.100.13.139:7020>
From: <sip:1004@10.100.13.255:5060>;tag=f11afe5a
Call-ID: M2JhYjBjMGRkYzQzNTA2ZmFm
YWU4MzViN2NiOTVlMTA.
CSeq: 1 INVITE Allow: INVITE, ACK, CANCEL,
OPTIONS, BYE, REGISTER, SUBSCRIBE,
Content-Type: application/sdp
Supported: replaces
User-Agent: 3CXPhone 4.0.13679.0
Content-Length: 407
v=0o=3cxVCE 375041100 14258475
IN IP4 10.100.12.230
c=IN IP4 10.100.12.230t=0 0
```

```
m=audio 40006 RTP/AVP 0 8 3 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15a=ptime:20 a=sendrecv
m=video 40004 RTP/AVP 34
c=IN IP4 10.100.12.230
a=rtpmap:34 H263/90000
39a=fmtp:34 QCIF=1;CIF=1;SQCIF=1;CIF4=1;
a=sendrecv
```

Thus for each IP's SIP packets are parsed for INVITE and STATUS 200 OK messages. From these SIP packets, SDP parts are separated and parsed for tracing ports.

6) STATUS 200 OK message sample format

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
10.100.12.230:56727;branch=z9hG4bK-d8754z-
8f3c792019497b01-1---d8754z-
received=10.100.12.230;rport=56727
From: <sip:1004@10.100.13.255:5060>;tag=f11afe5a
To: <sip:1006@10.100.13.139:7020>;tag=as043f7e02
Call-ID: M2JhYjBjMGRkYzQzNTA2Zm
CSeq: 2 INVITE
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE,
REFER, SUBSCRIBE, NOTIFY,INFO
Contact: <sip:1006@10.100.13.139>
Content-Type: application/sdp
Content-Length: 309
v=0
c=IN IP4 10.100.13.139
t=0 0
m=audio 18638 RTP/AVP 3 0 8 101
a=rtpmap:101 telephone-event/8000
a=silenceSupp:off - - - -
a=sendrecv
m=video 0 RTP/AVP 34
```

The forensic analyzer tracks the IP addresses and ports caught up in the VoIP conversation from the above parsed messages. Here the message part "m=audio" followed by the number gives the ports assigned by that client for that specific communication. The "From" and "To" fields contain a 4-digit number followed by '@' symbol which provides the IP address of the client who is involved in that particular exchange. [3]

7) RTP Filter

The process of filtering the RTP packets begins as soon as the forensic analyzer traces the ports and IPs used for the VoIP communication. To accomplish this, the forensic analyzer filters all the UDP packets having the selected ports as the source and destination ports since most of the RTP transfer occurs through lightweight UDP packets. RTP filter collects all such UDP packets by using the port number identified port identifier.

8) RTP Rescheduler

Once RTP packets are filtered from a particular communication, the projected packet-reordering algorithm, as given in Fig.2, reorganizes RTP packets based on the sequence no and separates the data from the packets. The sequence no of the RTP packets ranges from zero to 65536. Once it reaches 65536, it again starts from zero. The marker bit M is set for the beginning of the communication. The time stamps give idea about the time at which the packet is sent. The PT (payload type) specifies the type of the codec used for the payload content. The projected VoIP forensic analyzer will extract the voice information associated with RTP packet payload and writes to PCAP file temporarily. Now packets are reordered according to the packet-reordering algorithm and are given briefly in the following section. The respective RTP packet format with its parameters are given in Fig 3. For a conceptual view.

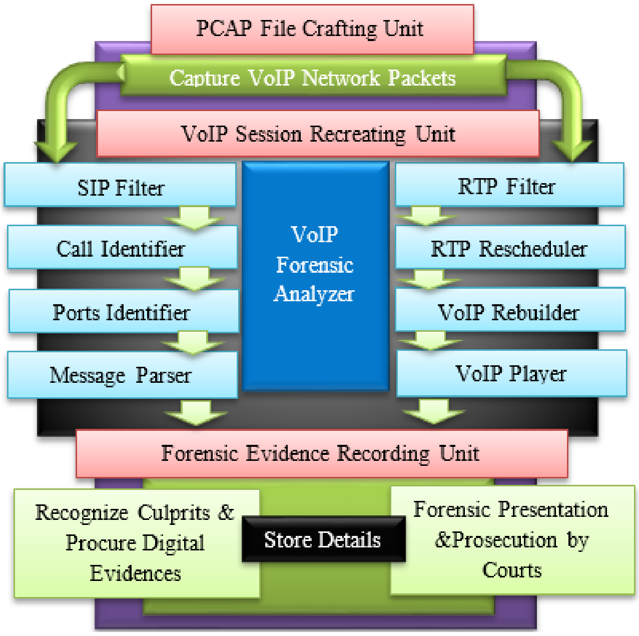

PCAP File Crafting Unit
Capture VoIP Network Packets
VoIP Session Recreating Unit
SIP Filter
Call Identifier
Ports Identifier
Message Parser
VoIP Forensic Analyzer
RTP Filter
RTP Rescheduler
VoIP Rebuilder
VoIP Player
Forensic Evidence Recording Unit
Recognize Culprits & Procure Digital Evidences
Store Details
Forensic Presentation &Prosecution by Courts

Fig. 2. Architecture of VoIP Forensic Analyzer

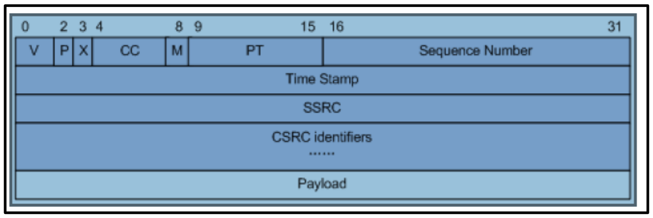


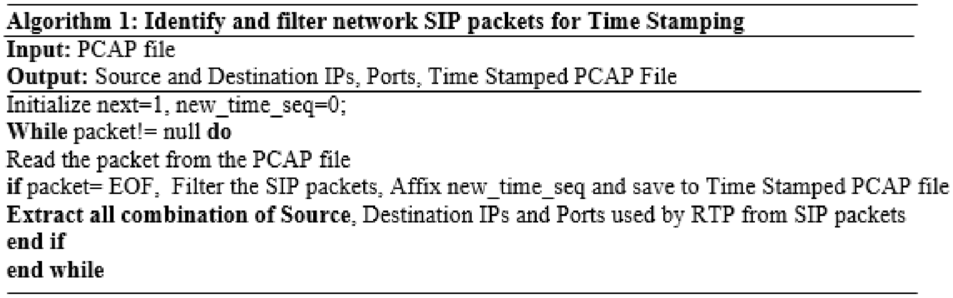
Fig. 3. RTP Packet Format

9) VoIP Stream Rebuilder

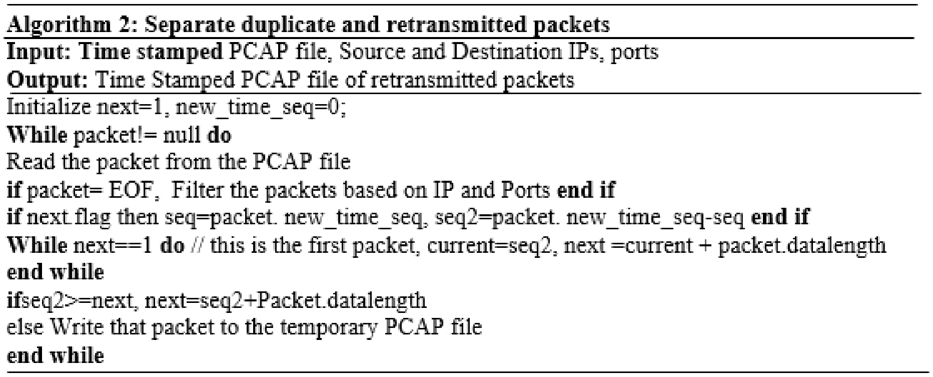
This vital component of proposed architecture rearranges the RTP communication session stream from the identified and separated combinations of IPs and Ports of involved VoIP clients. RTP packet payloads will be combined together after extracting it from header section of RTP packet followed by respective session rescheduling to reconstruct actual voice data [13, 14]. The proposed set of TCP/UDP reordering algorithms developed and given in this section would accomplish this task of rescheduling and rebuilding VoIP communication stream. Thus recreated voice stream is warehoused for upcoming forensic processing by forensic evidence recording unit of the projected VoIP analyzer. The first algorithm will identify the VoIP stream by filtering all SIP packets available in the network session followed by tracing all combinations of IPs and Ports. The second algorithm accomplishes the tracing of retransmitted or duplicate packets from VoIP communication stream. The reordering of VoIP communication stream is carried out by third algorithm. In this set of algorithms, "next" variable directs the subsequent packets. "new_time_seq" indicates regenerated timestamp for each such packets. "packet" denotes present packet for forensic processing. "EOF" designates a packet's end. "seq_relative" indicates subsequent probable time stamp of the packet. "datalength" provides packet's length. UDP packet's offset value is represented by "next.flag". We briefly outline the structure of set of algorithms in this section. During the deployment, we consider essential packet parameters and execute packet-processing functions. Thus, this set of algorithm reproduce the actual voice data present in respective VoIP communication session stream [11,14]. The set of three algorithms developed are given below.

10) VoIP Player

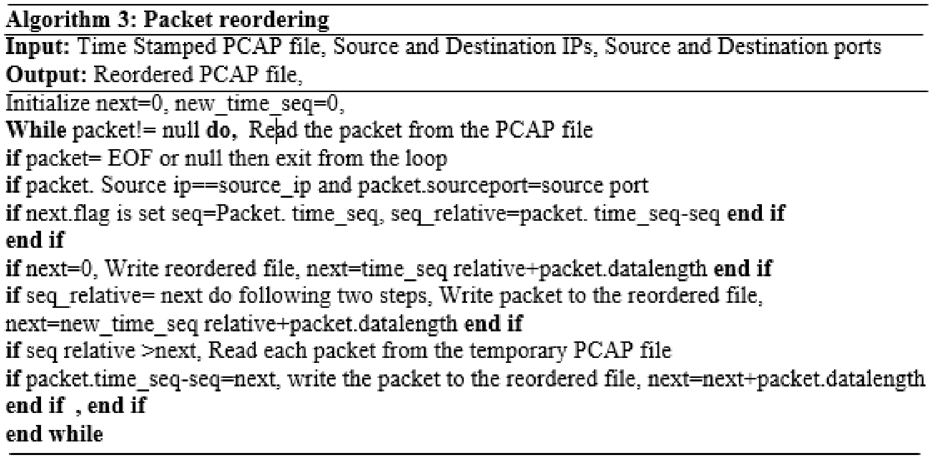
VoIP forensic player module automatically identify the codecs used by VoIP communication from the packets and decodes the processed voice data with help of SIP's codec G.729, which has already encoded using G.711.This player, plays recreated VoIP steam and voice in it perfectly.



```
Algorithm 1: Identify and filter network SIP packets for Time Stamping
Input: PCAP file
Output: Source and Destination IPs, Ports, Time Stamped PCAP File
Initialize next=1, new_time_seq=0;
While packet!= null do
Read the packet from the PCAP file
if packet= EOF,  Filter the SIP packets, Affix new_time_seq and save to Time Stamped PCAP file
Extract all combination of Source, Destination IPs and Ports used by RTP from SIP packets
end if
end while
```



```
Algorithm 2: Separate duplicate and retransmitted packets
Input: Time stamped PCAP file, Source and Destination IPs, ports
Output: Time Stamped PCAP file of retransmitted packets
Initialize next=1, new_time_seq=0;
While packet!= null do
Read the packet from the PCAP file
if packet= EOF,  Filter the packets based on IP and Ports end if
if next.flag then seq=packet. new_time_seq, seq2=packet. new_time_seq-seq end if
While next==1 do // this is the first packet, current=seq2, next =current + packet.datalength
end while
ifseq2>=next, next=seq2+Packet.datalength
else Write that packet to the temporary PCAP file
end while
```



```
Algorithm 3: Packet reordering
Input: Time Stamped PCAP file, Source and Destination IPs, Source and Destination ports
Output: Reordered PCAP file,
Initialize next=0, new_time_seq=0,
While packet!= null do,  Read the packet from the PCAP file
if packet= EOF or null then exit from the loop
if packet. Source ip==source_ip and packet.sourceport=source port
if next.flag is set seq=Packet. time_seq, seq_relative=packet. time_seq-seq end if
end if
if next=0, Write reordered file, next=time_seq relative+packet.datalength end if
if seq_relative= next do following two steps, Write packet to the reordered file,
next=new_time_seq relative+packet.datalength end if
if seq relative >next, Read each packet from the temporary PCAP file
if packet.time_seq-seq=next, write the packet to the reordered file, next=next+packet.datalength
end if  , end if
end while
```

C. Forensic Evidence Recording Unit

This component of the projected architecture will excavate forensic facts to confirm and pin down the malevolent VoIP users tangled in underhanded actions. This unit organizes the forensic information along with procuring actual evidences in form of VoIP voice data between parties and produce a detailed criminal statement against parties of a particular communication session for their subsequent investigation. The criminal statement consists of forensic particulars such as IPs, ports, credentials of VoIP clients or parties and period of malicious act along with regenerated digital proofs. This criminal statement is submitted to concerned cyber cell of a particular region to grab actual malefactors followed by their prosecution [14].

VI. Results and Discussions

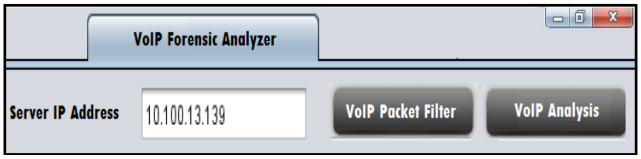
This segment furnishes the description of GUIs developed for forensic investigation of VoIP followed by the performance analysis of packet stream reassembly methodology with prevailing forensic structures. The projected innovative framework is entitled as VoIP Forensic Analyzer (VFA).

A. VOIP Control Panel

This GUI of control panel as shown in Fig 4 contains an option for loading any previous network sessions in the form of PCAP file. After stacking the PCAP file, it demands to specify the specific IP and ports used for the proxy server in VoIP analysis control panel. This data is essential in order to filter the required packets from the selected PACP file and identify the IP's and corresponding ports of the clients that have established a connection with the proxy server.

The ports used by the client and the server for communication are obtained by performing the port extraction procedure that we have explained in the VoIP session recreation unit. In selecting the "Filter Packets" button in the control panel, it starts the filtering process and displays the Client IPs and ports. Now the designed framework finds out the VoIP conversations that have been performed by the selected IPs and ports and will display the forensic results in various GUIs.

Projected framework has many GUI, which traces and displays forensic details in many steps. Here we outline and unveil GUIs that display very critical forensic detail so that the suspected VoIP communication stream can be recognized, the user can be tracked, and legal actions can be carried out with substantial evidences.

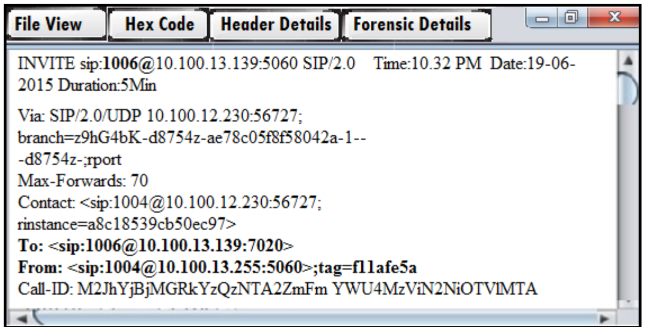


Server IP Address: 10.100.13.139

Fig. 4. GUI of VoIP Control Panel

B. Forensic Details from Header Panel

This GUI of header panel as shown in the Fig.5 displays forensic details, which gives an insight into handshake process that has been done for the available VoIP connections. This GUI encompasses the IPs and ports castoff by VoIP clients or parties along with "codec" used voice encoding. Header panel traces these forensic details from VoIP network packets collected and processed with the help of our proposed packet rescheduling and rebuilding algorithm.



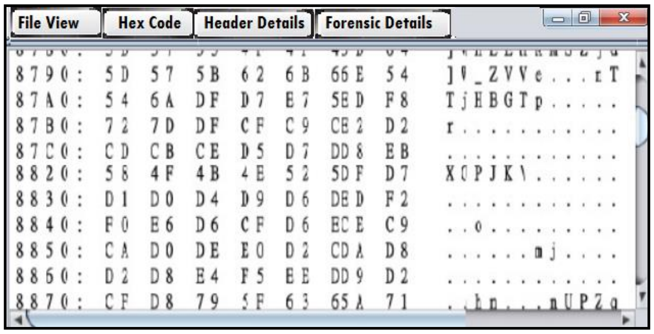
```
INVITE sip:1006@10.100.13.139:5060 SIP/2.0   Time:10.32 PM  Date:19-06-
2015 Duration:5Min

Via: SIP/2.0/UDP 10.100.12.230:56727;
branch=z9hG4bK-d8754z-ae78c05f8f58042a-1--
-d8754z-;rport
Max-Forwards: 70
Contact: <sip:1004@10.100.12.230:56727;
rinstance=a8c18539cb50ec97>
To: <sip:1006@10.100.13.139:7020>
From: <sip:1004@10.100.13.255:5060>;tag=f11afe5a
Call-ID: M2JhYjBjMGRkYzQzNTA2ZmFm YWU4MzViN2NiOTVlMTA
```

Fig. 5. GUI of Header panel

C. Hex Code Panel

This GUI of hex code panel as shown in Fig.6 provides the hex code of each byte in the data on the left side and its corresponding ASCII code on the right side for analysis.



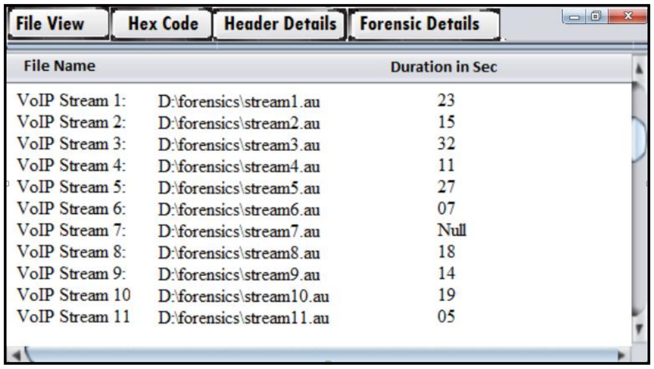
```
8790:  5D  57  5B  62  6B  66E  54   ]V_ZVVe...rT
87A0:  54  6A  DF  D7  E7  5ED  F8   TjHBGTp.....
87B0:  72  7D  DF  CF  C9  CE2  D2   r...........
87C0:  CD  CB  CE  D5  D7  DD8  EB   ............
8820:  58  4F  4B  4E  52  5DF  D7   X0PJK\......
8830:  D1  D0  D4  D9  D6  DED  F2   ............
8840:  F0  E6  D6  CF  D6  ECE  C9   ..o.........
8850:  CA  D0  DE  E0  D2  CDA  D8   .......mj....
8860:  D2  D8  E4  F5  EE  DD9  D2   ............
8870:  CF  D8  79  5F  63  65A  71   ..hn...nUPZq
```

Fig. 6. GUI of Hex Code Panel

This panel decodes the VoIP communication session and display it in hex format that characterize the pattern of loaded PCAP file. This pattern can be further used with other pattern matching vulnerability detection framework or tools.

D. File View Panel

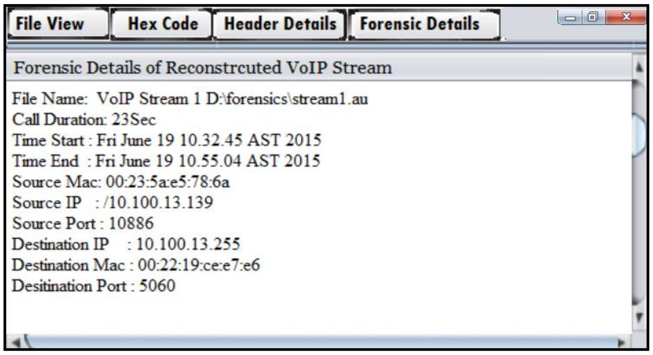
This GUI of file view panel as shown in Fig. 7 provides forensic information about the files reconstructed during the process of analysis. It also helps the investigator to know how many files (calls) are reconstructed to get an idea about the duration of each call with specific IP and port numbers



File Name		Duration in Sec
VoIP Stream 1:	D:\forensics\stream1.au	23
VoIP Stream 2:	D:\forensics\stream2.au	15
VoIP Stream 3:	D:\forensics\stream3.au	32
VoIP Stream 4:	D:\forensics\stream4.au	11
VoIP Stream 5:	D:\forensics\stream5.au	27
VoIP Stream 6:	D:\forensics\stream6.au	07
VoIP Stream 7:	D:\forensics\stream7.au	Null
VoIP Stream 8:	D:\forensics\stream8.au	18
VoIP Stream 9:	D:\forensics\stream9.au	14
VoIP Stream 10	D:\forensics\stream10.au	19
VoIP Stream 11	D:\forensics\stream11.au	05

Fig. 7. GUI of File view panel

E. Forensic panel



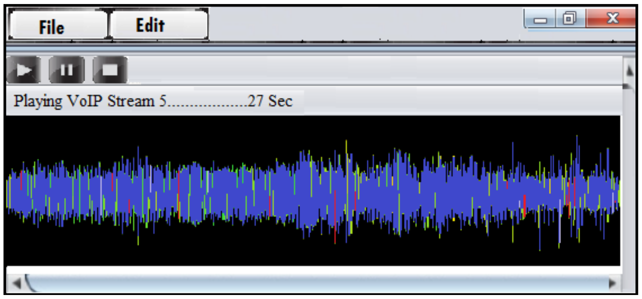
Forensic Details of Reconstrcuted VoIP Stream

File Name: VoIP Stream 1 D:\forensics\stream1.au
Call Duration: 23Sec
Time Start : Fri June 19 10.32.45 AST 2015
Time End : Fri June 19 10.55.04 AST 2015
Source Mac: 00:23:5a:e5:78:6a
Source IP : /10.100.13.139
Source Port : 10886
Destination IP : 10.100.13.255
Destination Mac : 00:22:19:ce:e7:e6
Desitination Port : 5060

Fig. 8. GUI of Forensic panel

This GUI of Forensic panel as shown in Fig.8 provides an in depth forensic details of each of the VoIP call that framework has reconstructed. It contains the details about file name of the reconstructed file. Call duration in seconds. Starting time of the call, End time of the call, IP address, port numbers and Mac Ids of VoIP clients in a particular communication stream.

F. VoIP Forensic player



Playing VoIP Stream 5..................27 Sec

Fig. 9. GUI of VoIP Forensic player

This VoIP forensic player as shown in Fig.9 plays the actual voice communication reconstructed from the suspected VoIP network stream. Once particular VoIP stream is reconstructed, VoIP forensic player decodes the processed voice data with help of SIP's codec G.729. These voice data were already encoded using G.711. This is one of the critical milestone that we have achieved in our research work which provides substantial evidences to prosecute malicious users with their conversation.

G. Performance Analysis of Projected Forensic Analyzer

The significant outcome that proves efficiency and trustworthiness of RTP rescheduling and reconstruction technique of proposed framework are sketched in this section. Performance of projected framework is systematically compared and verified against Wire shark and Ethercap, which are general-purpose free access packet analysis tools. Even if these tools collect and reorder packet, they fail in replaying the VoIP voice data after reconstruction. Here we compare rescheduling and reconstruction time taken by projected framework with respect to above-mentioned tools using PCAP file with malicious VoIP streams up to two lakh packets as shown in Fig.10. The result produces better yield in delivering commendable faster and exact reconstruction of VoIP streams. This is mainly due to managing retransmitted and duplicate packets by time stamping strategy in VoIP sessions, which fastens reconstructing and replaying VoIP sessions in various cases.

Projected architecture exhibits commendable quickness in regeneration actual VoIP communication stream from PCAP file and withstand large no of packets and its subsequent processing shows its reliability in tracing digital corruptions using VoIP platforms. We also tested the performance of proposed framework with and without using novel time stamping technique as shown in Fig. 11 with respect to the size of PCAP file against reconstruction time. When time stamping technique is devised, the session reconstruction time is drastically boosted by more than 60% of the reconstruction time taken by the technique without employing time stamping. Above saturation point (SP) as shown in graph, the performance of proposed framework provides poor reconstruction time in handling large PCAP file sizes without employing time stamping technique. Thus our novel time stamping technique critically is uplifting the performance of proposed reconstruction technique.

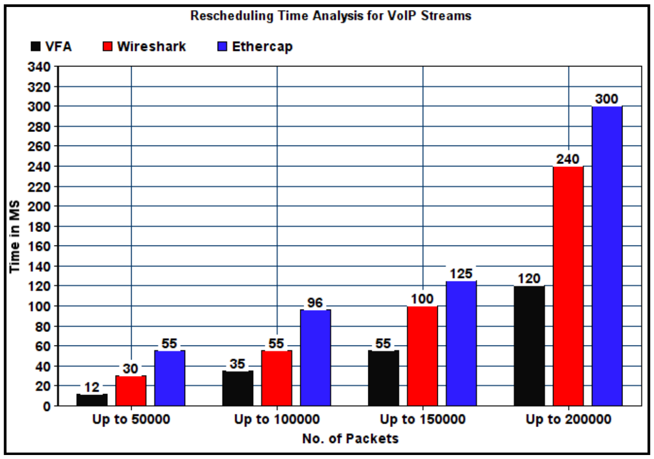


Fig. 10. Time analysis for VoIP reordering

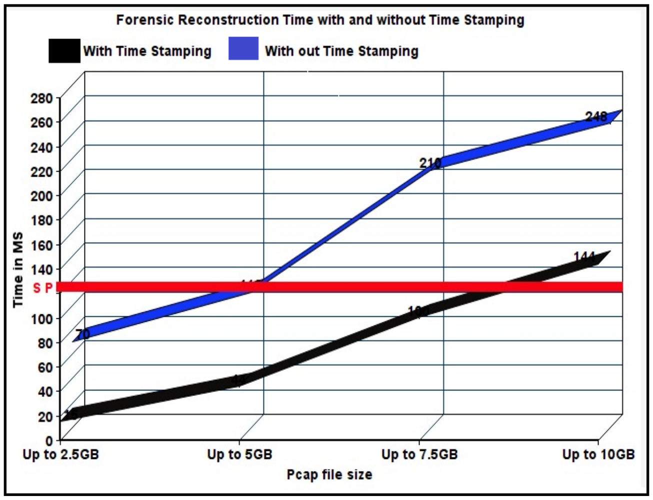


Fig. 11. Time analysis with and without time stamping technique

H. Brief Implementation details

This framework is developed with Java using JPCAP library [17] and Winpcap which are typical softwares used for data link layer processing. It permits bagging and dissecting network layer packets. Minimum configurations of 4 GB RAM and duel core level processor are preferred for processing PCAP files with 100000 packets. Since it uses Java and related subsidiaries, proposed work is thus platform independent.

VII. Conclusion

Network administrators and investigators use the projected architectural framework for retrospective forensic analysis of VoIP communication infrastructure. We have also outlined the novel technique for packet reconstruction for subsequent forensic analysis. This VoIP forensic analyzer effectively hints credentials of malicious VoIP users and craft environment for their subsequent prosecution. This software tool is easily deployable with Windows and Linux environments. While the framework outlined above is capable of detecting, identifying and reconstruction in a VoIP network, there remains a significant legal hurdle due to the cross-border nature of internet systems, especially international law needs to clear provision for the prosecution of malicious users worldwide. The proposed idea can be easily extended for forensic analysis of other internet protocols to make it support multiple protocols.

Acknowledgements

This research project was supported by the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University, KSA under research project grant no. 2014/01/2400.

References

[1] Ahmad Almulhem, (2009) “Network Forensics: Notions and Challenges” IEEE International Symposium on Signal Processing and Information Technology, pages 463-466, Dec 2009.

[2] François, J,State, R. Engel, T. Festor, O (2010)”Digital Forensics in VoIP networks” IEEE International Workshop on Information Forensics and Security (WIFS), Vol 1,pp-1-6, Dec 2010

[3] Ibrahim M, Abdullah, M.T. Dehghantanha, A “VoIP evidence model: A new forensic method for investigating VoIP malicious attacks” IEEE International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), pp 201-206, June 2012.

[4] Hofbauer, S, Quirchmayr, G., Beckers, K.” A Privacy preserving Approach to Call Detail Records Analysis in VoIP Systems” IEEE Seventh International Conference on Availability, Reliability and Security (ARES), Aug 2012

[5] Gao Hongtao “Forensic Method Analysis Involving VoIP Crime” IEEE Fourth International Symposium on Knowledge Acquisition and Modeling (KAM), pp 214-243, Oct 2011.

[6] Ahmad Azab,Paul Watters, Robert Layton “Characterising Network Traffic for Skype Forensics” IEEE Third Workshop (CTC), on Cybercrime and Trustworthy Computing, pp-19-27, Oct 2012

[7] Y.Q. Wang, M. Qi (2011), “Computer Forensics in Communication Networks”, IEEE International Communication Conference on Wireless Mobile and Computing Nov. 2011

[8] Khidir M. Ali (2012),” Digital Forensics Best Practices and Managerial Implications”, IEEE Fourth International Conference on Computational Intelligence, Communication Systems and Networks,pp-196 – 199, Jul 2012

[9] Edewede Oriwoh, Paul Sant(2013)," The Forensics Edge Management System ", IEEE 10th International Conference on Ubiquitous Intelligence & Computing and 10th International Conference on Autonomic & Trusted Computing, pp-544 - 550 Jun 2013.

[10] Eviyanti Saari, Aman Jantan (2013), " E-Cyborg: The Cybercrime Evidence Finder", 8th IEEE International Conference on Information Technology in Asia (CITA), pp-1 - 6, July 2013.

[11] Manesh T, B Brijith, Mahendra Prathap Singh, “An Improved Approach towards Network Forensic Investigation of HTTP and FTP Protocols”, International conference on Advances in Parallel Distributed Computing Communications in Computer and Information Science,Springer Berlin Heidelberg, Volume 203, 2011, pp 385-392.

[12] Manesh T, Brijith B, Bhraguram T M, R Rajaram, (2013) “Network Forensic Investigation of HTTPS Protocol ” International Journal of Modern Engineering Research, Vol. 3, Issue. 5, Sep - Oct. 2013.

[13] Manesh T, M Mohammed Sha, K Vivekanandan, (2014) “Forensic investigation framework for P2P protocol ” IEEE International conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp-256- 264, July 2014.

[14] M Mohemmed Sha, T Manesh, Saied M Abd El-atty, (2015) “Forensic Framework for Skype Communication” International conference on Advances in Intelligent Systems and Computing (ISTA-15), Springer Berlin Heidelberg, Vol. 2, pp-197- 211, July 2015.

[15] Rich Baseil, (2010) , IEEE Signal Processing Society [online] E-mail: r.baseil@ieee.org.

[16] Shaoqiang Wang, DongSheng Xu (2010). “Analysis and Application of Wireshark in TCP/IP Protocol Teaching.” International Conference on E-Health Networking, Digital Ecosystems and Technologies, pp269 - 272 Dec 2010.

[17] Shen Zihao, and Wang Hui, (2009) “Network Data Packet Capture and Protocol Analysis on Jpcap Based”. IEEE Proceedings of International Conference on Information Management, Innovation Management and Industrial Engineering, vol 3, pages 329-332, May 2009.