Extraction of Electronic Evidence from VoIP: Forensic Analysis of a Virtual Hard Disk vs RAM

David Irwin
University of South Australia, Australia
david.irwin@postgrads.unisa.edu.au

Jill Slay
University of South Australia, Australia
jill.slay@unisa.edu.au

Arek Dadej
University of South Australia, Australia
arek.dadej@unisa.edu.au

Malcolm Shore
University of Canterbury, New Zealand
malcolm.shore@canterbury.ac.nz

ABSTRACT

The popularity of Voice over the Internet Protocol (VoIP) is increasing as the cost savings and ease of use is realised by a wide range of home and corporate users. However, the technology is also attractive to criminals. This is because VoIP is a global telephony service, in which it is difficult to verify the user's identification. The security of placing such calls may also be appealing to criminals, as many implementations use strong encryption to secure both the voice payload as well as to control messages making monitoring such VoIP calls difficult since conventional methods such as wire-tapping is not applicable to VoIP calls. Therefore, other methods of recovering electronic evidence and information from VoIP are required. This research looks at what protocol evidence remains after a VoIP call has taken place examining both a virtual hard disk and the Random Access Memory (RAM).

This paper proposes a set of identifiable credentials based on packet header information contained within the VoIP protocol stack. A series of controlled tests were undertaken whereby these credentials were forensically searched for on a virtual machine which was used to make the VoIP call. This experiment was then repeated by a search for the same protocol credentials within the RAM.

Keywords: Computer forensics, digital evidence, electronic evidence, Voice over Internet Protocol, VoIP, memory forensics

1. INTRODUCTION

Voice over Internet Protocol (VoIP) technology has radically changed the way that telephony data is communicated and thus it has begun to revolutionise the Australian Telecommunications industry. With the tremendous growth in popularity and broadband access capability to Internet, such technologies have emerged that allows telephone calls to be routed over Internet infrastructure rather than the traditional Public Switched Telephone Network (PSTN) infrastructure. This technology, called VoIP, uses the Internet Protocol (IP) to route packets containing small portions of voice conversations between the callers.

The popularity of VoIP is increasing as the cost savings and ease of use is realised by a wide range of home and corporate users. However, the technology is also attractive to criminals, especially the non-carrier VoIP. This is because (1) VoIP is a global telephony service, in which it is difficult to verify the user's personal identification (2), the security of placing such calls may also be appealing to criminals, as many implementations use strong encryption to secure both the voice payload as well as to control messages, and (3) monitoring or tracing such VoIP calls is difficult since conventional methods such as wire-tapping is not applicable to VoIP calls. Therefore, other methods of recovering evidence and information from voice over IP protocol are required. It is essential that forensic computing researchers devise methods to allow law enforcement agencies to overcome some of the aspects of this method of telephony that are advantageous to criminals.

This introduction provides an overview of the VoIP transport protocols and a signalling protocol. This establishes the information from which it is possible to forensically retrieve fields from the protocol headers as well as user registration information for a VoIP call.

VoIP is not a single protocol in itself but rather a collection of a number of co-existing and competing protocols, which are used for setting up, maintaining and tearing down calls and protocols for the encapsulation and transportation of packets over the Internet. This collection of protocols referred to as the protocol stack contains the Internet Protocol (IP) (Postel, 1981a). The IP is responsible for providing the internet addresses in its internet header allowing packets to be routed from their source to a destination IP address. The IP header format is shown below in Figure1.

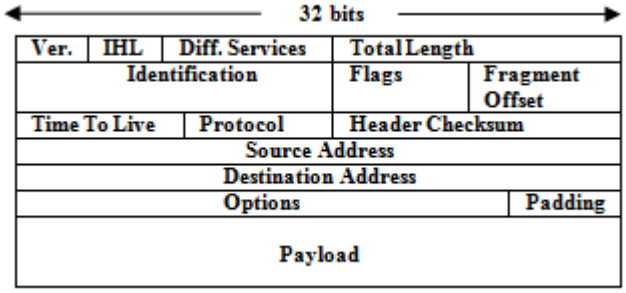


Figure 1 – IP packet header format

A reliable protocol for the transmission of packets across the Internet is the Transmission Control Protocol (TCP) (Postel, 1981b), which guarantees delivery because both lost and corrupt packets are re-transmitted. However, for real-time VoIP calls there is no purpose in re-transmitting lost packets. TCP makes use of flow control, which temporarily suspends the transmission of packets until the corrupt packet is successfully re-transmitted.

Instead, the User Datagram Protocol (UDP) (Postel, 1980), although a less reliable protocol since packet delivery is not guaranteed, is more suited to the requirements of VoIP. The selected system is thus an IP/UDP protocol stack. The IP datagram provides a source and destination address to identify the sending and receiving host but further information is required to provide an exact delivery location. This information is provided by the UDP, which adds the source and destination port numbers. The combination of an IP address and port number provides a socket. Common applications are configured to work via specific ports. The IP/UDP stack and UDP header format is shown below in Figure 2.

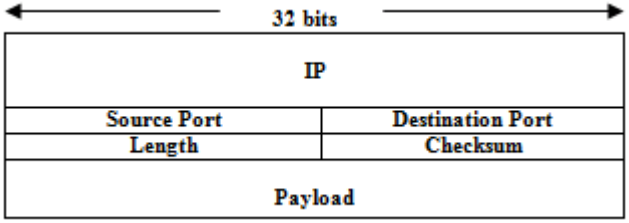


Figure 2 – IP/UDP stack showing the UDP packet header format

The reason for choosing UDP over TCP is that UDP does not guarantee delivery of packets whereas TCP does. TCP guaranteed delivery is achieved by the re-transmission of lost or out of sequence packets, which is undesirable for real-time audio.

The Real-Time Transport Protocol (RTP) (Schulzrinne et al, 2003) provides network transport for real-time applications such as transmitting audio over a packet switched network. The RTP protocol is chosen for the transmission of voice data because it adds additional information such as a sequence number to each packet to provide the application receiving the audio an opportunity to sequence packets in the correct order within a buffer and also provides a timestamp for each individual packet which allows playback of the audio at regular intervals. The protocol stack has now become an IP/UDP/RTP stack. Not that the use of RTP is not necessary to make and receive VoIP calls, it simply adds additional data to assist the receiving application with packet order and playback as well as allowing multiple users to facilitate VoIP conferencing. The IP/UDP/RTP stack and RTP packet header format is shown below in Figure 3.

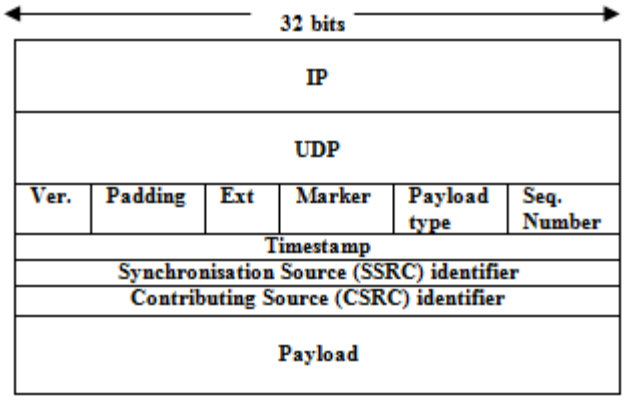


Figure 3 – IP/UDP/RTP stack showing the RTP packet header

The sequence numbers and timestamps are unique for each packet. The sequence numbers are incremented by one for each packet sent allowing the destination to re-order packets that have arrived out of sequence and detect packet loss. The timestamp conveys information relating to the sampling of data packets.

The Synchronisation Source (SSRC) field identifies the source of the synchronization e.g. computer clock whereas the Contributing Source (CSRC) field identifies the source of the individual contributions that make up the single data stream payload for the packet. It is not necessary to use the RTP to participate in a VoIP call. VoIP applications such as Skype (Download, 2009a) do not make use of the RTP whereas X-Lite (Download, 2009d) does. The experiments performed within this research were conducted on both applications Skype and X-Lite but only X-Lite results are discussed since the RTP adds an extra protocol from which fields from within the protocol header may be searched for.

An audio sample is packetised and transported across a network. If the packet is too large and needs to be fragmented or originated from multiple sources, then each fragment will have the same CSRC to identify that they have come from the same data stream but different sequence numbers, allowing them to be re-constructed in the correct sequence whereas different sources will have a unique CSRC. This is used for synchronization at the destination.

The protocols IP, UDP and RTP are essentially protocols which encapsulate and transport packet data containing the information necessary to identify the source and destination of the calling parties and deliver the payload containing the voice component. The following protocol, the Session Initiation Protocol (SIP) (Rosenberg, et al, 2002) is a signalling protocol used to establish, maintain and tear-down the call when terminated. SIP allows the calling parties called User Agents (UAs) to locate one another using a network of proxy servers, which allows UAs to be registered and invite other UAs to join in an Internet multimedia call called a session, based on HTTP requests/responses. Each transaction consists of a request that invokes a particular method and at least one response as shown in Figure 4. Jane uses her VoIP application to send an 'Invite' request to Joe. The 'Invite' request is a SIP method that specifies the action that Jane wants Joe to take, accepting the call from Jane. The 'Invite' request passes through two proxy servers to reach Joe, initiating a response from each proxy, (1) – (5). The destination 'Ringing' response from Joe is returned to the sender again passing through each proxy, (6) – (8). Confirmation of accepting the 'Invite' request is confirmed with an 'OK' response from Joe to Jane, (9) – (11). The media session can start once Jane has acknowledged the 'OK' response, (12). The session can be terminated by either party but is shown to be Joe in this instance, (13) – (14).

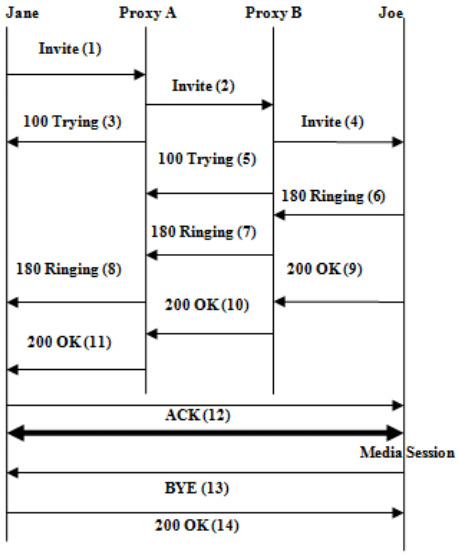


Figure 4 – SIP session between two UAs

As previously discussed, VoIP represents a packet based Internet technology for voice communication as opposed to traditional telephone calls. This will impact law enforcement as wire-tapping is no longer valid. A new architecture is required for the lawful interception of VoIP traffic as varying VoIP applications and Internet technologies means that IP packets will travel in dissimilar paths making it difficult to choose an intercept point (Karpagavinayagam et al, 2007).

Possible interception entities can be divided into two areas, those belonging to the Internet Service Provider (ISP) and those belonging to the Law Enforcement Agency (LEA), which may be located within the ISP or external to it. Entities related to the ISP include signalling, the setup, maintaining and tearing down of IP calls. The data entity is responsible for collecting the IP voice traffic which may be implemented on the same sub-network as the VoIP gateway, which connects to your existing telephone line. Entities related to the LEA may exist within the ISP architecture but remain under the control of the LEA, otherwise the gathered information requires authorisation from the ISP for access and decryption keys.

There is no single entity that VoIP traffic may be intercepted, however, the more central the components are on the signalling path makes lawful interception of VoIP more easy to perform (Seedorf 2008). Unfortunately, VoIP is characterised by a high degree of decentralisation, especially for mobile users connecting to the Internet for short periods of time. This may result in a move from nodes in a network under the authority of Law Enforcement Agencies to a situation where lawful interception is performed by requests to the Internet operators. For example, Skype uses proprietary encryption as opposed to published standards, DES (Data Encryption Standard) and AES (Advanced Encryption Standard). This requires LEAs to approach Skype for the encryption keys if they require captured VoIP traffic to be decrypted.

The VoIP application Skype is popular, but also the subject of numerous network forensics investigations because of its ability to traverse network address translation and bypass firewalls (Leung and Chan, 2009). Any Skype client, an individual with the Skype application deployed on their computer system for the purpose of making VoIP calls, may unwillingly become a super node on the Skype communication path maintain the Skype overlaying network, which may pose considerable risk to corporate businesses. Only by analysing Skype traffic, its communication network will it become possible to detect Skype UDP sockets. This will allow the blocking of Skype traffic.

Several models currently exist for investigation in digital forensics but the framework from the Digital Forensics Research Workshop (DFRWS) provides sequential steps for digital forensic analysis. These steps are shown below:

- Identifcation
- Preservattion
- Collection
- Examination
- Analysis
- Presentation

The examination and analysis of captured memory is performed in a read-only fashion, which leaves the original information un-altered.

As well as capturing the VoIP data, protocols, call configuration, raw packets with payloads, network forensic patterns (Pelaez and Fernandez, 2009) may be used to collect details about the VoIP user's activities from monitoring components in the network architecture such as incoming and outgoing numbers, geographical location, call duration with start and end times. A forensic pattern, a systematic approach to the forensic collection and analysis of data may allow for real-time analysis of VoIP captured traffic.

In previous work (Simon and Slay, 2006; Slay and Simon , 2008; Simon and Slay 2009), memory forensics have been carried out on computers (on various platforms) to extract VoIP transactions. This is a move away from VoIP network forensics to looking at specific memory storage areas within a computer to discover what artefacts of the VoIP call remain after the call has been terminated.

It should be noted that it is not the purpose of this paper to provide a framework for digital forensic collection but rather demonstrate through experimentation that more information may be gathered from the contents of RAM as opposed to hard disk storage media for digital artefacts left behind after a VoIP call. As the cost of hard disk storage reduces, it allows larger sizes of physical memory storage shipped with computer systems. The increase in the size of RAM is not so prevalent, allowing RAM to be searched much faster, gigabytes of RAM as opposed to terabytes of hard disks. The seizure of RAM contents may be accomplished by the seizure of a computer system that is still powered on or covertly over the Internet without the user's knowledge.

2. EXPERIMENT 1

The recovery and analysis of digital forensic evidence from a computer forensic examination is well documented and rigorously implemented in order to prevent ambiguous interpretation of results within a court of law. This area of memory forensics can be used to recover documents, images and emails from a target computer system.

However, there is little research in the area of VoIP forensics, the art of applying memory forensic techniques to identify VoIP packets that remain on a computer system after a VoIP call. The proposed research builds upon the previous work (Simon and Slay, 2006; Slay and Simon , 2008) of the Defence and Systems Institute (DASI) at the University of South Australia. The experimental setup is shown below in Figure 5.

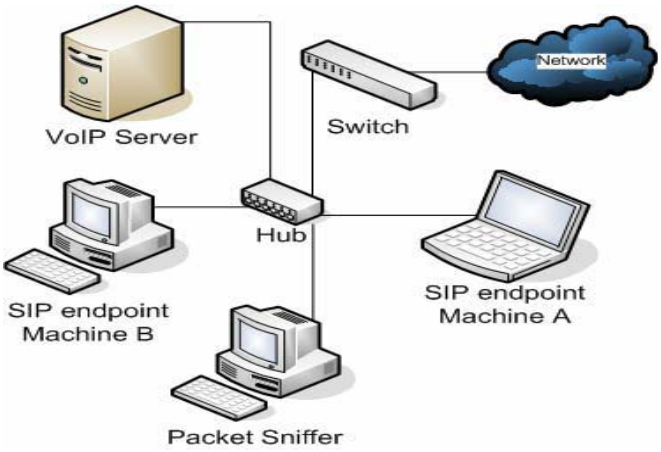


Figure 5 – VoIP call and packet detection setup

The experimental setup involved making a single VoIP call between two computers using Windows Operating System (OS) within a virtual machine (Download, 2009b). Each virtual machine was cloned from a virtual hard disk with a fresh install of Windows OS. The use of virtualisation creates a contained environment into which the X-Lite VoIP application is installed. X-Lite makes use of the SIP protocol, thus requiring a user to first register their details with a SIP server to initiate a VoIP call.

Once the call is initiated, both users may communicate with each other until one party terminates the call. Once the call is terminated the virtual machine is shut down at endpoint B and the virtual hard disk is analysed using X-Ways (Download, 2009e) forensic software.

The purpose of the packet sniffer is to monitor and collect the packets being transferred between both endpoints capturing the protocols and the payload as they pass from the source of the call to its destination and responses from the destination to the source. The packet sniffer used was Wireshark (Download, 2009c).

2.1 Target Artefacts

The target artefacts are based on protocol fields captured using Wireshark as well

as signalling information as shown in Figure 6.

Session Initiation Protocol:

- **Request-Line:**
- **Invite sip:8889215862@sip.pennytel.com**
  Method: Invite
- **Message Header**
  Contact:sip:8889215864@119.40.108.72:26610
- **To: "david"< sip:8889215862@sip.pennytel.com>**
  SIP Display info: "david"
- **SIP to address:sip:8889215862@sip.pennytel.com**
  SIP to address User Part: 8889215862
  SIP to address Host Part: sip.pennytel.com
- **From:"8889215864"sip:8889215864@sip.pennytel.com**

Figure 6 – SIP Invite request

Before a SIP call can be made, individuals or organisations are required to register with a SIP registration server, and are issued with a unique SIP address, which takes the form of users@hosts, where the user portion can be a name or telephone number and the host portion can be a domain name or network address, for example, 8889215862@sip.pennytel.com. The user part 8889215862 uniquely identifies the user agent and the host part identifies pennytel.com as the SIP service provider. This is performed using the SIP 'Register' request. Once registered, user agents can communicate with each other with the assistance of proxy servers, which can forward requests and responses to another proxy which is located closer to the actual end user agent.

Target artefacts based on SIP signalling protocol are:

user Ids 889215862 & 8889215864
host, pennytel.com (SIP service provider)
SIP server IP address
Call-ID

Figure 7 – Target SIP artefacts

Once the VoIP call was initiated, the audio file English.wav lasting 24 seconds was replayed in succession 5 times. The file English.wav consists of spoken power words "welcome, peace, love, joy, hope, compassion, mystery, unity, equality, creativity, freedom, health, spirit, inspiration, release".

The following protocol information captured from frame 797 of the VoIP call using Wireshark is shown below in figure 8. This information is used to form a byte search pattern for a manual search of the virtual hard disk using X-Ways.

Frame 797 (134 bytes on wire, 134 bytes captured)
Frame Number: 797
Frame Length: 134 bytes
Protocols in frame: eth:ip:udp:rtp
Ethernet Protocol:
Destination: Vmware_c1:09:d9 (00:0c:29:c1:09:d9)
Source: IntelCor_4a:d6:26 (00:21:6a:4a:d6:26)
Internet Protocol:
Source: 192.168.0.102
Destination: 192.168.0.105
Real-Time Transport Protocol:
Synchronisation Source ID: 0xf1fa6792 (4059719570)

Figure 8 – Wireshark capture of frame 797

The Ethernet Protocol (Hornig, 1984) provides the physical Ethernet Hardware Address (EHA) more commonly referred to as the Media Access Control (MAC) address which is a unique identifier assigned by the manufacturer of the Network Interface Card (NIC). This enables X-Ways to search for the 6 byte source/ destination MAC address, the 4 byte source/destination IP address and the 4 byte SSRC captured by Wireshark as artefacts left behind as evidence of a VoIP call.

2.2 Data Analysis and Results

Data analysis was performed using X-ways forensic software to view the virtual hard disk within which the VoIP call took place. The virtual hard disk was not imaged or copied in any way, but rather viewed in read-only mode from its original location.

Table 1 below shows the artefact results from performing a manual search and the number of hits.

Table 1 – Search for signalling protocol artefacts

Target – SIP signaling	Hits
192.168.0.102	0
192.168.0.105	0
8889215862	0
8889215864	0
Pennytel.com	0
Call-id	0

The entire duration of the VoIP call was 183 seconds, consisting of 17,704 frames, of which only 11 frames were related to the SIP at the start of the call (Invite – Trying – Ringing – Ack – Ok request/responses) and end of the call (Bye – Ok request/response). The remaining 17,693 frames were RTP. The search for packet header artefacts is shown below in Table 2.

Table 2 – Search for packet header artefacts

Target – Eth/IP/UDP/RTP	**Hits**
0x 00 21 6a 4a d6 26 (eth MAC src)	**0**
0x 00 0c 29 c1 09 d9 (eth MAC dst)	**18**
0x c0 a8 00 66 (IP src 192.168.0.102)	**1**
0x c0 a8 00 69 (IP dst 192.168.0.105)	**3**
0x f1 fa 67 92 (RTP SSRC src)	**0**
0x 22 14 ad 31 (RTP SSRC dst)	**0**

The VoIP call was received by the X-Lite application within the virtual environment. The 18 hits for the virtual machine MAC address were related to the installation of the X-Lite application within the virtual environment and not from the VoIP call. The hits for the source and destination IP addresses were not within any meaningful context i.e. a statistically occurring random hit for that 4 byte hex pattern. The search analysis shows that there are no artefacts remaining from the SIP signalling protocol or packet header information.

One may assume that the packet header information is only required to ensure that the payload reaches its destination, the virtual machine and is discarded as it is no longer required. The remaining encrypted payload will be utilised by the X-Lite application. Therefore an additional search can be performed for the encrypted payload. Subsequently, we have the original unencrypted source of the payload, English.wav, making it possible to also search for the decrypted payload at the destination virtual machine.

Is it logical to assume that if the virtual machine exchanges in excess of 17,000 RTP frames during the VoIP call that the latter frames are more likely to persist in memory? Table 3 below shows the search analysis for encrypted payload for latter occurring frames during the VoIP call.

Table 3 – Search for encrypted VoIP payload

Encrypted payload for Frame	Hit
17,704	**No**
17,600	**No**
17,400	**No**
17,200	**No**
17,000	**No**
16,500	**No**
16,000	**No**
15,000	**No**
14,000	**No**

A search for the content of the sound file English.wav also resulted in no hits. This may be due to the fact that the sound file is also contaminated with background noise when it is digitised by the VoIP application, altering its original byte content.

2.3 Conclusion: Experiment 1

The purpose of this research was to find digital evidence left behind in memory after a VoIP call had been received within a virtual environment. The resulting forensic analysis of the virtual hard disk using X-Ways has yielded no indication of packet header information from the VoIP transport protocols or the SIP signalling protocol artefacts persisting in memory once the call has been terminated.

This initial research has resulted in a change in direction from performing search analysis in physical memory (such as hard disks) to investigating the digital artefacts which temporarily exist in RAM.

If packet header information from the transport protocols can be recovered, including such fields as sequence numbers and timestamps, then this will allow partial reconstruction of the encrypted VoIP payload. However an additional question that arises is, 'Does both the encrypted payload and decrypted payload persist in RAM at the same time, thus allowing reconstruction of the audible VoIP call?', allowing for enough frames to exist in order to make human sense of the reconstruction.

Previously, 183 seconds of voice required in excess of 17,000 frames. Assuming a minimum of 20 seconds of voice in order to make sense of the communication, this would require the reconstruction of almost 2000 frames.

3. RANDOM ACCESS MEMORY

The previous research, the search for digital artefacts, digital evidence that

persists in memory after the completion of a VoIP call within a virtual environment based on VoIP protocol information provided no evidence of the call having taken place. This situation can be overcome in different ways as outlined below.

3.1 Lawful Interception

Lawful interception has existed since the dawn of the PSTN. Wire tapping, as it is traditionally known, was an easy task to perform on analogue telecommunications devices. However, the interception of traffic across the Internet (ETSI, 2001) is substantially more difficult as Internet traffic may cross geographical boundaries where no Law exists to allow legal interception of IP based traffic. The Internet provider may provide no means for lawful interception within their network, either due to lack of physical means or the right of privacy to its customers.

If a physical location within a network is chosen as the intercept point, then all traffic may be captured and not just traffic between the target machines. How does the Internet provider guarantee that non-targeted customers' information is not being intercepted even by Law Enforcement Agencies (LEAs).

3.2 Live Forensic Analysis

The distinction between live and post forensics is the nature in which digital evidence is recovered. Live forensics infers that the target machine is in use at the time potential evidence is collected. This would be a covert action, performed without the knowledge of the user. This in itself raises a number of questions:

1. The legality of using of spyware which resides within a target machine, await the initiation of a VoIP call and perform a RAM capture?
2. How is the spyware deployed?
3. Does the spyware search the RAM contents and recovers the information it requires, packet header information from the VoIP transmission protocols, user identification used to log into the VoIP application or signalling protocol information, such as SIP service provide registration details?
4. Or is the entire RAM contents processed on another machine which would require the RAM contents to be transmitted across the Internet?
5. Would the user of the target machine notice degradation in processor capability, an increase in memory use or an increase in broadband upload?

These are important considerations when constructing spyware but first the actual digital evidence which may be recovered from a RAM capture must be identified to determine the feasibility of injecting spyware onto a target machine.

3.3 Experiment 2

The RAM experimental setup differs from the previous setup in that virtualisation is no longer used. Previously VoIP calls were made within a virtual environment and the virtual hard disk searched for digital artefacts persisting in memory. RAM analysis is more complicated, in that, to perform a RAM capture, the X-Ways forensic analysis software would have to be deployed within the virtual environment.

Also the physical amount of RAM available for capture would be less than the system RAM which is hosting the virtual machine. Instead the RAM capture is performed on the system RAM of the endpoint machine. The experimental setup involved making a single VoIP call between two computers using Windows OS. Two VoIP applications were used, Skype and X-Lite for making the VoIP call at separate times.

Once the VoIP call is initiated, both users may communicate with each other until the call is terminated. During the call X-Ways RAM editor is used to expose the system RAM, and a copy of the RAM contents is saved to a file. This file can then be manually searched using the X-Ways search function for target artefacts. Once the call is terminated and the search analysis of the RAM captured file is complete, the endpoint machine is shutdown to allow the contents of the RAM to dissipate. The contents of RAM are not permanent and will slowly disappear over time as the voltage supply decays to the cells holding the binary information (zeros and ones). After a significant amount of time the endpoint machine is re-started and the process is repeated using a different VoIP application.

3.4 Target Artefacts

The target artefacts are the same for experiment 1, again based on protocol fields captured using Wireshark as well as signalling information. Once the VoIP call was initiated, the audio file Highway Blues.wma lasting 93 seconds was replayed in succession twice. This file was chosen because of its longer duration compared to the previous audio file English.wav.

The following protocol information captured from frame 753 of the VoIP call using Wireshark is shown below. This information is used to form a byte search pattern for a manual search of the RAM file produced by X-Ways as shown in Figure 9 below.

Frame 753 (214 bytes on wire, 214 bytes captured)
Frame Number: 753
Frame Length: 214 bytes
Protocols in frame: eth:ip:udp:rtp
Ethernet Protocol:
Source: Dell_a2:dd:cb (00:24:e8:a2:dd:cb)
Destination: IntelCor_4a:d6:26 (00:21:6a:4a:d6:26)
Internet Protocol:

Source: 192.168.0.102
Destination: 192.168.0.100
Real-Time Transport Protocol:
Synchronisation Source ID: 0x542ec3f5 (1412350965)

Figure 9 – Wireshark capture of frame 753

This enables X-Ways to search for the 6 byte source/destination MAC address, the 4 byte source/destination IP address and the 4 byte SSRC captured by Wireshark as artefacts left behind as evidence of a VoIP call.

3.5 Data Analysis and Results

Data analysis was performed using X-ways forensic software to view the RAM in editor mode allowing textual and hexadecimal searches to be performed. In addition, packet reconstruction software (Simon, 2008) was also used to identify packet information from the RAM captured file. This software retrieves all protocols involved in a VoIP call and displays them in a format that allows Wireshark to display the retrieved packets.

Table 4 below shows the protocol information recovered by the packet reconstruction software.

Table 4 – VoIP protocols recovered from software

Protocol	Hits
SIP	**3**
ETH/IP/UDP/RTP	**454**

Analysis of the RAM capture file with the packet reconstruction software recovered 454 packets. The recovered packets are not in sequence order. They are in the order they were recovered from the RAM file. This is correct, as in reality, pages of memory in RAM would not be sequential as objects and processes may be moved in and out of RAM as they are required. Table 5 below shows the X-Ways search for digital artefacts.

Table 5 – Search for packet header artefacts

Target – SIP/Eth/IP/UDP/RTP	Hits
8889215862 (SIP user)	**280**
8889215864 (SIP user)	**149**
Pennytel (SIP host)	**921**
119.40.108.72 (Server IP address)	**81**
SIP Call-ID (MGFKNjg…)	**24**
0x 00 24 e8 a2 dd cb (eth MAC src)	**2402**
0x 00 21 6a 4a d6 26 (eth MAC dst)	**1487**
0x c0 a8 00 66 (IP src 192.168.0.102)	**1613**
0x c0 a8 00 64 (IP dst 192.168.0.100)	**3112**
0x e4 bd 7b 03 (RTP SSRC src)	**744**
0x 54 2e c3 f5 (RTP SSRC dst)	**48**

The size of the RAM captured file was 3,620,888 KB. The most accurate identification for the existence of digital artefacts is the RTP synchronisation source identifier, in total 792 (744 + 48) packets were detected. This is considerably more that that detected by the packet reconstruction prototype software which still requires fine tuning. Alternatively, the 792 RTP packets may be fragmented, just partial packets containing the correct SSRC but missing information. The packet reconstruction software detected 454 full packets which could be viewed extensively in Wireshark.

It would be a worthwhile exercise to examine closely the packets reconstructed by the software. To sequence them by their respective RTP sequence number to find several consecutive frames then search for them using X-Ways to determine their physical location in RAM. Are they also consecutive or appear to be random? This is shown below in Table 6.

Table 6 – Byte offset for packets located in RAM

IP Seq. No.	RTP Timestamp	RAM Offset (bytes)	Difference (bytes)
60514	**0x 00 31 d9 cc**	**0160378936**	
60515	**0x 00 31 da 6c**	**0160370744**	**8192**
60516	**0x 00 31 db 0c**	**0160362552**	**8192**
60517	**0x 00 31 db ac**	**0160346168**	**16384**
60518	**0x 00 31 dc 4c**	**0160329784**	**16384**

Although each packet is only 214 bytes in size, consecutive packets appear to be separated by a multiple of 8192 bytes (8KB). This may be related to the size of the page files in memory.

3.6 Conclusion: Experiment 2

The purpose of this research was to find digital evidence left behind in RAM after a VoIP call had been received between two SIP endpoint machines.

The resulting forensic analysis of RAM captured using X-Ways provided an abundance of digital artefacts from Ethernet and IP addresses, IP sequence numbers, RTP timestamps and SSRCs, all part of the VoIP transmission protocol packet header information, Also recovered was SIP signalling information identifying users, SIP provider(s) and SIP server IP address and call-ID(s).

In addition, the use of packet reconstruction software applied to the RAM captured file also extracted a large number of packets.

4. PARADIGM SHIFT – HARD DISK TO RAM FORENSICS

The simplified experiments discussed previously provided an insight not previously investigated, the contents of RAM for VoIP artefacts left behind after a VoIP call. To investigate this further, the following experiments provide a definitive analysis of RAM captures for both X-Lite and Skype calls using packet searching software.

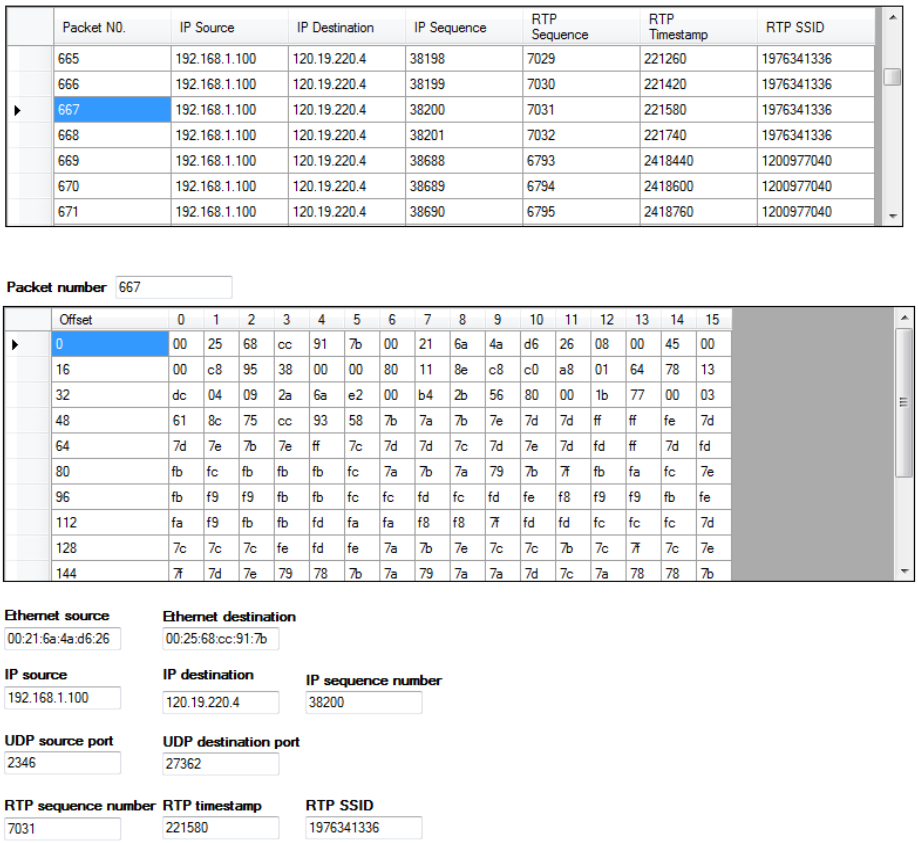
4.1 Skype and X-Lite Call Analysis

The following results are the analysis of a 4.0GB RAM capture performed after a single Skype call lasting 3 minutes on a laptop. The laptop was the powered down to allow the RAM contents to dissipate. A second RAM capture was performed the following day after 3 successive X-Lite calls, the first lasting 30 seconds, the second lasting 30 seconds and the third lasting 3 minutes. The remnants of calls that can be recovered in the RAM capture are compared against the Wireshark capture of the VoIP calls to identify the number of packets recovered versus packets from the VoIP call. The analysis is shown below in table 7.

Table 7 – Skype and X-Lite call analysis

VoIP Application	Call durataion	Packet count from Wireshark	RAM packets recovered (total)	RAM packets recovered (unique)	% of call recovered
Skype	**180 seconds**	**18,701**	**41,959**	**18,208**	**97.4%**
X-Lite	**30 seconds**	**3,097**	**4,759**	**3,093**	**99.9%**
X-Lite	**30 seconds**	**3,290**	**5,488**	**3,274**	**99.5%**
X-Lite	**180 seconds**	**9,089**	**17,695**	**9,063**	**99.7%**

Note that the total amount of packets recovered from the RAM capture exceeds the original number of packets in the call. It was found that in some instances, packets existed in up to 6 different memory locations in RAM. These were filtered out to examine the number of unique packets recovered. Figure 6 is used to demonstrate how multiple X-Lite calls are separated from each other within a single RAM capture by observing the SSID (synchronisation source ID), unique to each call stream within the call.



Packet N0.	IP Source	IP Destination	IP Sequence	RTP Sequence	RTP Timestamp	RTP SSID
665	192.168.1.100	120.19.220.4	38198	7029	221260	1976341336
666	192.168.1.100	120.19.220.4	38199	7030	221420	1976341336
667	192.168.1.100	120.19.220.4	38200	7031	221580	1976341336
668	192.168.1.100	120.19.220.4	38201	7032	221740	1976341336
669	192.168.1.100	120.19.220.4	38688	6793	2418440	1200977040
670	192.168.1.100	120.19.220.4	38689	6794	2418600	1200977040
671	192.168.1.100	120.19.220.4	38690	6795	2418760	1200977040

Packet number 667

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	00	25	68	cc	91	7b	00	21	6a	4a	d6	26	08	00	45	00
16	00	c8	95	38	00	00	80	11	8e	c8	c0	a8	01	64	78	13
32	dc	04	09	2a	6a	e2	00	b4	2b	56	80	00	1b	77	00	03
48	61	8c	75	cc	93	58	7b	7a	7b	7e	7d	7d	ff	ff	fe	7d
64	7d	7e	7b	7e	ff	7c	7d	7d	7c	7d	7e	7d	fd	ff	7d	fd
80	fb	fc	fb	fb	fb	fc	7a	7b	7a	79	7b	7f	fb	fa	fc	7e
96	fb	f9	f9	fb	fb	fc	fc	fd	fc	fd	fe	f8	f9	f9	fb	fe
112	fa	f9	fb	fb	fd	fa	fa	f8	f8	7f	fd	fd	fc	fc	fc	7d
128	7c	7c	7c	fe	fd	fe	7a	7b	7e	7c	7c	7b	7c	7f	7c	7e
144	7f	7d	7e	79	78	7b	7a	79	7a	7a	7d	7c	7a	78	78	7b

Ethernet source 00:21:6a:4a:d6:26

Ethernet destination 00:25:68:cc:91:7b

IP source 192.168.1.100

IP destination 120.19.220.4

IP sequence number 38200

UDP source port 2346

UDP destination port 27362

RTP sequence number 7031

RTP timestamp 221580

RTP SSID 1976341336

Figure 6 – X-Lite packet analysis

4.2 X-Lite Call Registration and Setup

The X-Lite VoIP application is chosen for detailed analysis of call registration and setup because this information is unencrypted and transported in packets in plain text. The X-Lite call id is a lengthy and unique character string.

A search of the RAM capture using X-Ways forensic tool for the SIP registration call id, OTCWOTjimzI2OTK0NZRimGR4NJQ4ZDY1ZGE5Y2Q0ODC yields the X-Lite subscribe request for SIP pennytel identification 8889215864

(username). Each X-Lite call has a different call id for its setup, allowing each call id to be searched for and subsequently found in the RAM capture, for example the call 1 setup exchange captured by Wireshark is shown below in Figure 7.

Invite request SIP/SDP 8889215862 (number being called)
Invite request SIP/SDP
100 Trying
401
401 Unauthorized
Ack
401 unauthorized
Ack request
SIP/SDP Invite
SIP/SDP Invite
SIP/SDP Invite
100 Trying
180 Ringing

Figure 7 – X-Lite control signal during call setup

Searching for call 1 id MjQ3ZGM2MDA3ZDI3NjI2yZNiMzcIYTM3NWE5MTQONTE using X-ways discovered more than 80 occurrences of the call id in the RAM capture, including Invite requests, Ack and also Bye, which identifies the call is terminated, 'user hung up'. Armed with this information allows Law Enforcement to approach the identified SIP registrar, in this case pennytel, with the call id and the recovered sequence of packets for the X-Lite call with a request for the packet contents to be decrypted. It is not the purpose of this research to attempt packet payload decryption.

5. FUTURE WORK

The lack of VoIP protocol artefacts left behind on a virtual hard disk after a VoIP call resulted in a shift from disk forensics to RAM forensics. RAM forensics has successfully demonstrated the ability to recover VoIP protocol artefacts left behind in RAM after a VoIP call has taken place.

This research is initially motivated to recover packet sequence information to allow VoIP payloads to be reconstructed correctly. However continuing upon this will be the identification of the end-user(s), using a track and trace capability. If a user wishes to remain anonymous, they will most likely spoof IP and MAC addresses or be hidden behind firewalls.

The immediate foreseeable research is:

- The further development of a software tool to recover packet information in sequence and extract the encrypted payload
- identify the VoIP application from signalling and call setup packets

- perform a track and trace of VoIP end-users
- understand the paging of virtual memory in RAM to identify and recover the decrypted audio component

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the Australian Research Council in this work via Linkage Grant LP0989890 and additional scholarship contributions from the Australian Federal Police.

7. REFERENCES

Download (2009a). Skype Downloaded July 20, 2009 at www.skype.com.

Download (2009b). VM Workstation Downloaded July 15, 2009 at www.vmware.com.

Download (2009c). Wireshark Downloaded July 20, 2009 at www.wireshark.org,

Download (2009d). X-Lite Downloaded July 24, 2009 at www.counterpath.com.

Download (2009e). X-Ways Downloaded July 18, 2009 at www.x-ways.net,

ETSI TR 101 944 V1.1.2 (2001). Telecommunication Security - Lawful Interception - Issues on IP Interception ETSI TR 101 944 V1.1.2.

IETF RFC 768 (1980). User Datagram Protocol, Postel, J.

IETF RFC 791 (1981a). Internet Protocol, Postel, J.

IETF RFC 793 (1981b). Transmission Control Protocol, Postel, J.

IETF RFC 894 (1984). A Standard for the Transmission of IP Datagrams over Ethernet Networks, Hornig, C.

IETF RFC 3261 (2002). SIP: Session Initiation Protocol, Rosenberg, J. Et al.

IETF RFC 3550 (2003). RTP: A Transport Protocol for Real-Time Applications, Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V.

Karpagavinayagam, B., State, R. And Festor, O. (2007). Monitoring Architecture for Lawful Interception in VoIP Networks, Second International Conference on Internet Monitoring and Protection.

Leung, C.M., Chan, Y.Y. (2007). Network Forensic on Encrypted Peer-to-Peer VoIP Traffics and the Detection, Blocking, and Prioritization of Skype Traffics, 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaboration Enterprises.

Pelaez, J.C., Fernandez, E.B. (2009). VoIP Network Forensic patterns, 2009 Fourth International Multi Conference on Computing in the Global Information Technologies.

Seedorf, J. (2008). Principles, Systems and Applications of IP Telecommunications, Services and Security for Next Generation Networks, Second International Conference, IPTComm 2008.

Simon, M. (2008). Packet reconstruction software: Defence and Systems Institute (DASI) at the University of South Australia.

Simon, M. and Slay, J. (2006). Voice over IP: Forensic Computing Implications, 4th Australian Digital Forensics Conference, Edith Cowan University, School of Computer and Information Science, December 4, 2006.

Simon, M.and Slay, J. (2009). Enhancement of Forensic Computing Investigations through Memory Forensic Techniques. 2009 International Conference on Availability, Reliability and Security. Fukuoka Institute of Technology, Fukuoka, Japan pp.995-1000.

Slay, J. and Simon, M. (2008). Voice over IP Forensics. e-forensics 08: Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering.