CERIAS Tech Report 2016-7

BUILDING A DIGITAL FORENSIC INVESTIGATION TECHNIQUE FOR FORENSICALLY SOUND ANALYSIS OF COVERT CHANNELS IN IPV6 AND ICMPV6, USING CUSTOM IDS SIGNATURES AND FIREWALL SYSTEM LOGS

by Lourdes Gino Dominic Savio, Marcus K. Rogers, Raymond A. Hansen, Baijian Yang
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Graduate School Form
30 Updated 12/26/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Lourdes Gino Dominic Savio

Entitled

BUILDING A DIGITAL FORENSIC INVESTIGATION TECHNIQUE FOR FORENSICALLY SOUND ANALYSIS OF COVERT CHANNELS IN IPV6 AND ICMPV6, USING CUSTOM IDS SIGNATURES AND FIREWALL SYSTEM LOGS

For the degree of Master of Science

Is approved by the final examining committee:

Marcus K. Rogers
Chair

Baijian Yang

Raymond A. Hansen

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Marcus K. Rogers

Approved by: Jeffrey L. Whitten 7/19/2016

Head of the Departmental Graduate Program Date

BUILDING A DIGITAL FORENSIC INVESTIGATION TECHNIQUE FOR FORENSICALLY SOUND ANALYSIS OF COVERT CHANNELS IN IPV6 AND ICMPV6, USING CUSTOM IDS SIGNATURES AND FIREWALL SYSTEM LOGS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Lourdes Gino Dominic Savio

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2016

Purdue University

West Lafayette, Indiana

To my parents, Dominic and Sheela for giving me the wealth of knowledge and support in all my endeavors.

To my fiancée, Ramya who was the sole purpose and driving force for my Masters.

To the almighty God, without whose blessings, I wouldn't be where I am.

ACKNOWLEDGMENTS

This research would not have been possible without the guidance and continuous support of my committee members: Dr. Marcus Rogers, Prof. Raymond Hansen, and Prof. Baijian Yang. Thanks for enabling me from time to time and showing me the right path.

TABLE OF CONTENTS

Page

LIST OF FIGURES

GLOSSARY

Covert Channels	Any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy (Latham, 1986, p. 87).
Digital Forensics	The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations (Palmer, 2001, p. 16).
False Negatives	A False Negative is an incorrect identification of a malicious activity as being benign (Scarfone & Mell, 2007, p. 2-3).
False Positives	A False Positive is an incorrect identification of a benign activity as being malicious (Scarfone & Mell, 2007, p. 2-3).
Firewall	Firewall is a computer system that protects the network from network-based threats and attacks, and provides a single choke point where security and audit can be imposed. A firewall builds a blockade between an internal network that is assumed to be secure and trusted, and another network, usually an external (inter)network, such as the Internet, that is not assumed to be secure and trusted (Oppliger, 1997).

Firewall System Logs	Firewall System Logs or Audit Logs are a security-relevant chronological record, set of records, and/or destination and source of records that provide documentary evidence of the sequence of activities that have affected at any time a specific operation, procedure, or event (Schaeffer, 2010).
Forensic Model	A standard structured process that provides a suitable mechanism to be followed by the computer forensic investigators while conducting a computer forensic investigation, as all of their actions are subjected to scrutiny by the judiciary should the case be presented in the court (Yusoff, Ismail, & Hassan, 2011, p. 17).
ICMPv6	Internet Control Message Protocol version 6 (ICMPv6) is a protocol used by IPv6 nodes to report errors encountered in processing packets, and to perform other internet-layer functions, such as diagnostics (ICMPv6 "ping"). ICMPv6 is an integral part of IPv6, and the base protocol is required to be fully implemented by every IPv6 node (Conta & Gupta, 2006, p. 1).
IDS	Intrusion detection System (IDS) is a software that automates the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices (Scarfone & Mell, 2007, p. 2-1).
IDS Signature	A signature is a pattern that corresponds to a known threat. Signature-based detection is the process of comparing signatures against observed events to identify possible incidents (Scarfone & Mell, 2007, p. 2-4).

IPv6	Internet Protocol version 6 (IPv6) is a newer version of the Internet Protocol, designed as the successor to IP version 4 (IPv4). The changes from IPv4 to IPv6 fall primarily into the following categories: expanded addressing capabilities, header format simplification, improved support for extensions and options, flow labeling capability, and authentication and privacy capabilities (Hinden & Deering, 1998, p. 1).
Network Forensics	Network forensics is a sub-branch of digital forensics relating to the monitoring and analysis of computer network traffic for the purposes of information gathering, legal evidence, or intrusion detection. Unlike some other areas of digital forensics, network investigations deal with volatile and dynamic information. Network traffic is transmitted and then lost, so network forensics is often a pro-active investigation (Casey, 2011; Latham, 1986).
True Negatives	A True Negative is a correct identification of a benign activity as being benign (Scarfone & Mell, 2007, p. 2-3).
True Positives	A True Positive is a correct identification of a malicious activity as being malicious (Scarfone & Mell, 2007, p. 2-3).

ABSTRACT

Dominic Savio, Lourdes Gino M.S., Purdue University, August 2016. Building a digital forensic investigation technique for forensically sound analysis of covert channels in IPv6 and ICMPv6, using custom IDS signatures and firewall system logs. Major Professor: Dr. Marcus K. Rogers.

Covert Channels are communication channels used for information transfer, and created by violating the security policies of a system (Latham, 1986, p. 80). Research in the field has shown that, like many communication channels, IPv4 and the TCP/IP protocol suite has features, functionality and options which could be exploited by cyber criminals to leak data or for anonymous communications, through covert channels. With the advent of IPv6, researchers are on the lookout for covert channels in IPv6 and one of them demonstrated a proof of concept in 2006. Nine years hence, IPv6 and its related protocols have undergone major changes, which introduced a need to reevaluate the current situation of IPv6. The current research is a continuation of our (author of this thesis - Lourdes, and committee member - Prof. Hansen) previous studies (Lourdes & Hansen, 2015, 2016), which demonstrated the corroboration of covert channels in IPv6 and ICMPv6 by building a software for the same and testing against a simulated enterprise network. Our study had also explained how some of the enterprise firewalls and Intrusion Detection Systems (IDS) do not currently detect such covert channels, and how they could be tuned to detect them. The current research aimed at understanding if these detection mechanisms (IDS signatures) of IPv6 and ICMPv6 covert channels are forensically sound, and at exploring if the system logs left by such covert channels in the firewall could provide forensically sound evidence. The current research showed that the IDS signatures that detected certain covert channels in IPv6 and ICMPv6, conformed to the forensic soundness criteria of

'validity of the scientific method', and 'known/potential error rates'. The current research also showed that the firewall system logs potentially detected certain covert channels in IPv6 and ICMPv6 and also conformed to the forensic soundness criteria of 'validity of the scientific method'. Thus the current study showed that these could be used as digital forensic investigation techniques for network forensics of certain types of covert channels in IPv6 and ICMPv6.

CHAPTER 1. INTRODUCTION

A covert channel is "a method of communication that is not part of an actual computer system design, but can be used to transfer information to users or system processes that normally would not be allowed access to the information" (Rowland, 1997, p. 1). In order to understand the covert channels better, figure 1.1 shows an example of Alice and Bob, who would like to communicate their meeting time, secretively and decide to use the telephone ring as a communication mechanism. Alice, wanting to meet Bob at 1:23am, calls Bob on a public telephone near him, gives a single ring and cuts the call. Alice waits for a minute, calls Bob again, but this time gives two rings and cuts the call. Alice then waits for one more minute, calls Bob again, and gives three rings now and cuts the call. At this point Bob understands that it is Alice who is trying to communicate the message, "meet at 1:23am". Also, a 3rd person observing this would not understand this communication and would discard it as a prank call. Although the true purpose of the telephone ring is so that the receiver could pick the call to speak, Alice and Bob successfully created a covert means of communication, which went undetected.

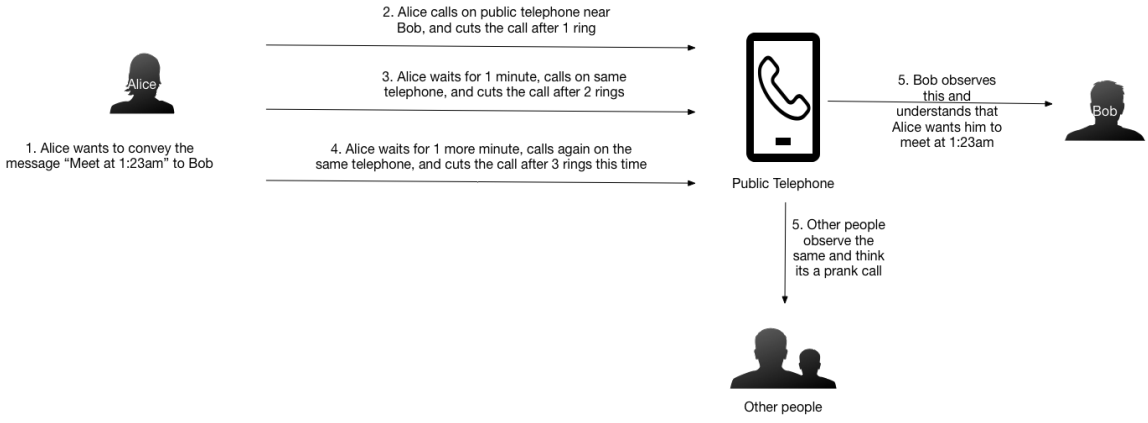

Figure 1.1.: Example of a covert channel

While potentially disruptive and effective, covert channels are often underrated and not much importance is given to them. According to Shah, Molina, Blaze, et al. (2006), this behavior can be explained as effect of covert channels having low bandwidths and requiring a compromise of a system in the first place. While the compromise itself would prove to be more fatal to the system, the covert channel created would have a lesser impact compared to the vulnerability that allowed the exploit and creation of the covert channel (p. 1). Since covert channels are hard to detect, there arises a possibility that cyber criminals might exploit those for covert communication. Hence analyzing and detecting covert channels could help digital forensic investigators while performing network forensics.

Digital forensic models provide a digital forensic investigator with a formalized process and set of procedures to be carried out when performing a digital forensic analysis. The process and procedure used in digital forensic analysis has a direct impact on the outcome of the digital forensic investigation. Hence it is crucial for digital forensic investigators to follow a standard structured process (Yusoffet al., 2011). Although various digital forensic models exist, which address a wide variety of digital crime scenarios, the center of these models is the analysis phase, where various investigative techniques are applied to the evidence collected, to identify the perpetrator of the crime or the crime itself (Yusoff et al., 2011).

The current research was an exploratory study which aimed to create a digital forensic investigation technique using custom IDS signatures and firewall system logs, to help in analyzing certain specific covert channel in IPv6 and ICMPv6, in a forensically sound manner.

1.1 Scope

Covert channels and various studies on the subject have been in existence for more than four decades. In 1973, Lampson explained a method of exploiting the communication channels which are “not intended for information transfer at all,

such as the service program's effect on the system load" and termed it "covert channels" as opposed to the "legitimate channels" of communication (p. 614). Many research studies showed that the weaknesses in the TCP/IP design and implementation could be exploited, which paved a way for new covert channels in IPv4 and IPv6. Lourdes and Hansen (2015, 2016) demonstrated the corroboration of such covert channels in IPv6 and ICMPv6 by building a software for the same and testing against a simulated enterprise network.

Rowland (1997) explained that these covert channels posed a major security threat as they could be exploited "in the areas of data smuggling and anonymous communication" (p. 1). He also noted that the detection of such channels could be difficult, mainly if they were encrypted. This might pose a major challenge in the field of digital forensics, mainly network forensics. The current research attempts to test the forensic soundness of two evidences, custom IDS signatures of Snort and Suricata IDS, and system logs of pfSense firewall, thereby creating a network forensic investigation technique, that standardizes the process of a network forensic analysis of the following covert channel in IPv6:

1. IPv6 Flow Label based covert channel
2. IPv6 Experimental Extension Header based covert channels
3. ICMPv6 based covert channels

1.2 Significance

According to McCusker (2006), cybercrime is a major part of transnational threat landscape owing to the increasingly complex online activity, that is could be classified as an organized crime. He also argues that the future of cybercrime resides in the digital environment that was primarily built to enable business and social relationships, although susceptible to malicious activity (p. 257). The increase in cybercrime demands better digital forensics to bring the criminals to justice.

According to Marcella Jr and Greenfield (2002), one of the major challenges faced by digital forensic investigators is the technical challenges that obstruct law enforcement's ability to find and prosecute cyber criminals (p. 198). Network forensics, a sub-branch of digital forensics is no exception to this challenge. Covert channels are a bigger menace owing to their stealthy nature. Various research studies attempted to detect covert channels in IPv4 and some were even successful. But the rapid growth of IPv6 adoption, from 0.14% in January 2009 to 11.23% in July 2016, is demanding the organizations and businesses to migrate to IPv6 (Google, 2016). This could potentially lead to cybercrime using covert channels in IPv6 and ICMPv6. Thus the current research was an exploratory study aimed at solving one of the technical challenges faced when performing forensics of certain covert channels in IPv6 and ICMPv6, a digital forensic investigation technique.

1.3 Research Question

Can 'custom IDS signatures' and 'firewall system logs' help in analyzing the following IPv6 and ICMPv6 based covert channels, and can they be tested for forensic soundness using the criteria of validity of the scientific method, and potential error rates:

1. IPv6 Flow Label based covert channel, where the Flow Label field in the IPv6 header is used to send covert data.

2. IPv6 Extension Header based covert channels, where the covert data is sent as the payload of experimental extension header types 253 and 254.

3. ICMPv6 based covert channels, where the covert data is sent as payloads of ICMPv6 headers type 1, 2, 3, 4, 128, 129, 253 and 254.

1.3.1 Statement of the Problem

With increasing number of devices connecting to the Internet everyday, there is a major paradigm shift from IPv4 to IPv6 addressing, as the Internet began to run out of IPv4 addresses (Blanchet, 2009). The implementation of IPv6 has security loopholes that could be exploited to perform covert communication (Lucena, Lewandowski, & Chapin, 2006). Using such covert channels, it is possible for cyber criminals, terrorists, whistle-blowers, anti-government groups and insiders to communicate stealthily. The lack of reliable detection mechanisms of IPv6 and ICMPv6 covert channels pose a major security threat. Thus the current study aimed at analyzing the techniques of detection of the following covert channels in IPv6 and ICMPv6 using custom IDS signatures and firewall system logs:

1. IPv6 Flow Label based covert channel
2. IPv6 Experimental Extension Header based covert channels
3. ICMPv6 based covert channels

The study also aimed at testing the forensic soundness of the techniques. Forensic soundness of a digital forensic process helps preserve the integrity of the evidence and could be tested by various criteria. Thus the current research aimed at testing two of the common criteria between McKemmish (2008) and *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993): the validity of the scientific method and the known/potential error rates.

1.4 Assumptions

The assumptions of this study were:

1. The network protocol stack exploited in the study is standard, conforming to their respective RFCs.

2. The design of the covert channel tool used in the research, is assumed to work in most of the networking environment. Although, the design of certain networking environments may prevent the operation of the software.

1.5 Limitations

The limitations of this study were:

1. The study does not focus on IPv4 based covert channels.
2. The study does not focus on covert channels created by TCP or UDP header exploits.
3. The study does not focus on encrypted covert channels.
4. The study does not take into consideration the network components and the network topologies that were not part of the design in the methodology.
5. The study does not test all the forensic soundness criteria for the IDS signatures and firewall system logs.
6. The study does not take into consideration all possible IDS signatures and firewall system logs.
7. The study being exploratory, does not measure reliability of the digital forensic investigation technique.
8. The data set generated and used by the study does not represent the real life traffic. Nor can it be extrapolated to represent real traffic. Moreover the study does not represent the network environment outside of the Covertv6 tool.

1.6 Delimitations

The delimitations of this study were:

1. The study focuses only on covert channels created by the Covertv6 tool, which exploits IPv6 flow label, IPv6 experimental extension headers and ICMPv6 header based covert channels.
2. The study focuses only on unencrypted covert channels.
3. The study includes only the specifications in the RFCs present at the time of beginning this research.
4. The study was restricted to only pfSense firewall, Snort IDS and Suricata IDS, and the network topology discussed in the methodology.
5. The study tests the forensic soundness criteria of validity of the scientific method, and known/potential error rates, only.
6. The study considers only the custom signatures described by Lourdes and Hansen (2016) and the firewall system logs of pfSense firewall.
7. The dataset generated and used by the study represented the normal ICMPv6 ping packets only. Moreover the study is confined to the network environment of the Covertv6 tool.

1.7 Summary

This chapter provided the scope, significance, research question, assumptions, limitations, delimitations, and definitions for the current research. The next chapter provides a review of the literature relevant to the research.

CHAPTER 2. REVIEW OF RELEVANT LITERATURE

2.1 Covert Channels

Covert channels and various research on the subject have been in existence for more than four decades. In 1973, Lampson (1973) explained a method of exploiting the communication channels whose purpose was not transfer of information, and termed it "Covert Channels" as opposed to the "legitimate channels" of communication (p. 614). In 1986, the Department of Defense not only gave a formal definition for covert channels but also classified them into "storage channels" and "timing channels" based on their method of operation (Latham, 1986, p. 45). While a covert storage channel involves direct/indirect reading/writing of processes in a covert manner, a covert timing channel involves signaling mechanisms that modifies the response time, often used to convey the data (Lucena et al., 2006, p. 148). Since a covert channel exploits a weakness in the system by hiding as a legitimate channel, it is different from a side channel which exploits the application's implementation to leak information (Shah et al., 2006, p. 1). Many research studies showed that the weaknesses in the TCP/IP design and implementation could be exploited, which paved a way for new covert channels, mainly in IPv4. With the migration of IPv4 to IPv6, many researchers have turned their focus on covert channels in IPv6 (Lucena et al., 2006).

2.2 Evaluating Covert Channels

In 1997, Rowland (1997) showed that IPv4 (mainly TCP in IPv4) has vulnerabilities that allowed him to exploit these in creating covert channels in his

implementation, called a “covert TCP program”, thereby leaking vital data through a network. He specifically used the IP identification field, TCP initial sequence number field, and TCP acknowledgment number field to show the exploit. He also explained how these posed a major security threat and how these could be exploited for smuggling data and communicating anonymously (p. 1). He also noted that the detection of such channels could be difficult, mainly if they were encrypted or if packets were mirrored off of a server, making them seem legitimate (p. 1).

Ahsan (2002) furthered the research by Rowland (1997) and presented his thesis on covert channel analysis and data hiding in IPv4. He not only provided a compendium of prior research in the field, but also explained various exploits that allowed the creation of covert channels. He explained that covert channel based on packet header manipulation were not restricted to only a TCP header, but also IGMP and ICMP headers by the use of various encoding mechanisms. He also discussed another method of creation of covert channels, “data hiding through packet sorting” (p. 53). This method worked based on various algorithms that could be used to sort and resort packets. These packets could then be encrypted using the IPSec architecture, thereby providing confidentiality through the network and avoiding detection of the packet header modifications by powerful firewall or IDS. Interestingly, he did not discuss the negative implications of covert channels such as data leakage, rather presented it as a means of improving the network security. Finally, he noted covert channels on IPv6 as future work, as early as 2002. Unfortunately, there is no published work of his, on the same.

Llamas, Allison, and Miller (2005) reviewed various research studies on covert channels at the time and showed the existence of numerous other exploits than the usual packet header manipulations discussed by the previous researchers, that allowed the creation of covert channels. They included, ‘Covert Messaging through TCP Timestamps’, ‘IP Checksum Covert Channels and Selected Hash Collision’, ‘Malicious ICMP Tunneling’, and ‘Exploiting authorized data streams over the HTTP protocol’ (p. 3). They also described various implementations of

covert channels such as, 'Data Hiding in TCP/IP with HTTP Reverse Proxy Servers', and 'IP Covert Timing Channels' (p. 4). Like, Ahsan (2002), Llamas et al. (2005) provided a glimpse of IPv6 covert channel research at that time. They also showed how Graf (2003) exploited the IPv6 Destination option to create a covert messaging tool. Even though IPv6 deployments were uncommon even as late as 2008, the research on covert channels in IPv6 existed as early as 2003 (Colitti & Gunderson, 2008).

2.3 Covert Channels in IPv6

One among the two important research studies, that were the basis for the current research is, Lucena et al. (2006), while the other one being a proof of concept by Murphy (2006). To be precise, Lucena et al. (2006) did a specification-based analysis, to identify flaws and ambiguities in protocols, which could be exploited to create covert channels, and provided a comprehensive list of 22 covert channels in IPv6 (p. 147). Some of their findings included:

1. The number of research studies on network storage covert channels were more than the number of research studies on network timing covert channels owing to the synchronization issues and lower bandwidth of the latter. This potentially eliminates the need for further study on network timing covert channels.

2. The most effective defense mechanisms of the time against IPv4 covert channels were protocol scrubbers, traffic normalizers and active wardens. They did not explain if these defense mechanisms hold good against IPv6 covert channels, which paves a way for a study on how these mechanism fare against IPv6 covert channels.

3. There were at least six IPv6 covert channels, created by exploiting six fields in the IPv6 header such as, Traffic Class, Flow Label, Payload Length, Next

Header (adding various extension headers to it), Hop Limit and Source Address.

4. Other potential IPv6 covert channels could be created by exploiting the extension headers such as the Hop-by-Hop Options Header, Routing Header, Fragment Header, Destination Options Header, Authentication Header, Encapsulation Security Payload Header.

5. More covert channels could be created by tunneling traffic such as IPv6 in IPv4, IPv6 in IPv6 and IPv4 in IPv6.

They finally noted covert channels in ICMPv6 as their future work. But, in the same year, Murphy (2006) showed this to be possible.

In 2006, Murphy (2006) demonstrated a proof of concept tool called "V00d00n3t", which exploited the ICMPv6 echo-reply payload to create a covert data channel. He showed that he was able to transfer data over the Internet in a network infrastructure which was designed using a tunnel broker and a set of routers and servers, as explained in the topology by Murphy (2006). Although the design validated that the packets would survive in a 'slick' 6 network and wild uncontrolled environments, he noted that the limitation was the survival of the covert data in a production environment with firewall, IPS/IDS etc. This motivated the introduction of firewall and IDS in the current research.

2.4 Understanding the Exploitable Fields of IPv6 and ICMPv6

RFC 2460 (Hinden & Deering, 1998) defines the specifications of an IPv6 packet. Figure 2.1 shows the various field of the IPv6 header. Version is a 4-bit field indicating the version of IP protocol (4 or 6). Traffic class is a 8-bit field indicating the various classification of traffic for the purpose of Quality of Service (QoS). Flow label is a 20-bit field used to identify unique flows. RFC 6437 (Rajahalme, Amante, Jiang, & Carpenter, 2011) explains that IPv6 flow label could be used along with

source address and destination address to efficiently enable classification of IPv6 flows in case of fixed IPv6 main headers. RFC 6437 mandates that the flow label not be modified by any networking devices en-route from source to the destination. This presents a possibility of utilizing this 20-bits for transmitting covert data.

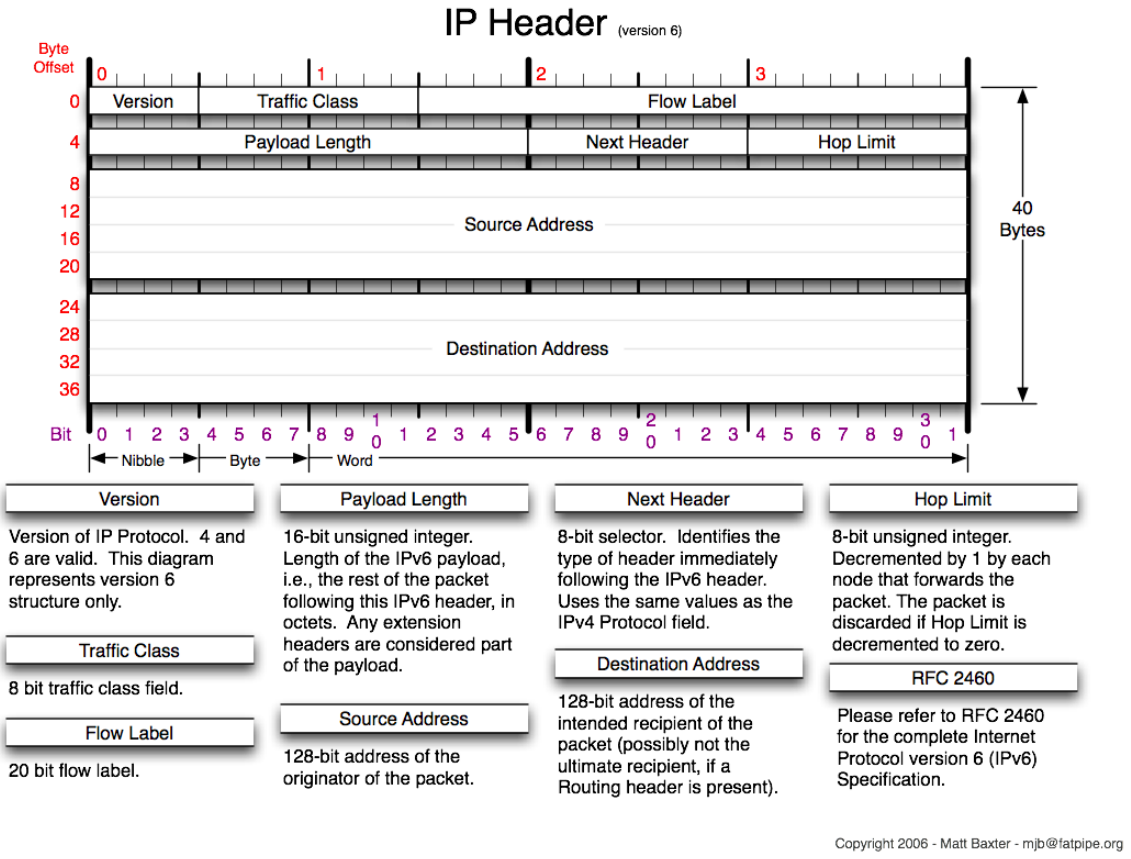

Figure 2.1.: IPv6 header (Baxter, 2016)

The payload length is a 16-bit field that identifies the length of the payload data. The next header field is a 8-bit field that identifies the subsequent header of the IPv6 header. According to Hinden and Deering (1998), an IPv6 packet has the ability to place certain option information as headers, in between the IPv6 header and the upper-layer header. These are the extension headers. Figure 2.2 shows how the extension headers fit in the IPv6 header. Various types of IPv6 extension headers include, Hop-by-Hop options extension header, Routing extension header,

Fragment extension header, Destination Options extension header, Authentication extension header, and Encapsulating Security Payload extension header. In addition RFC 4727 Fenner (2006) defines two experimental header types where the next header value is 253 and 254. This presents a potential to exploit these two experimental headers to create covert channels.

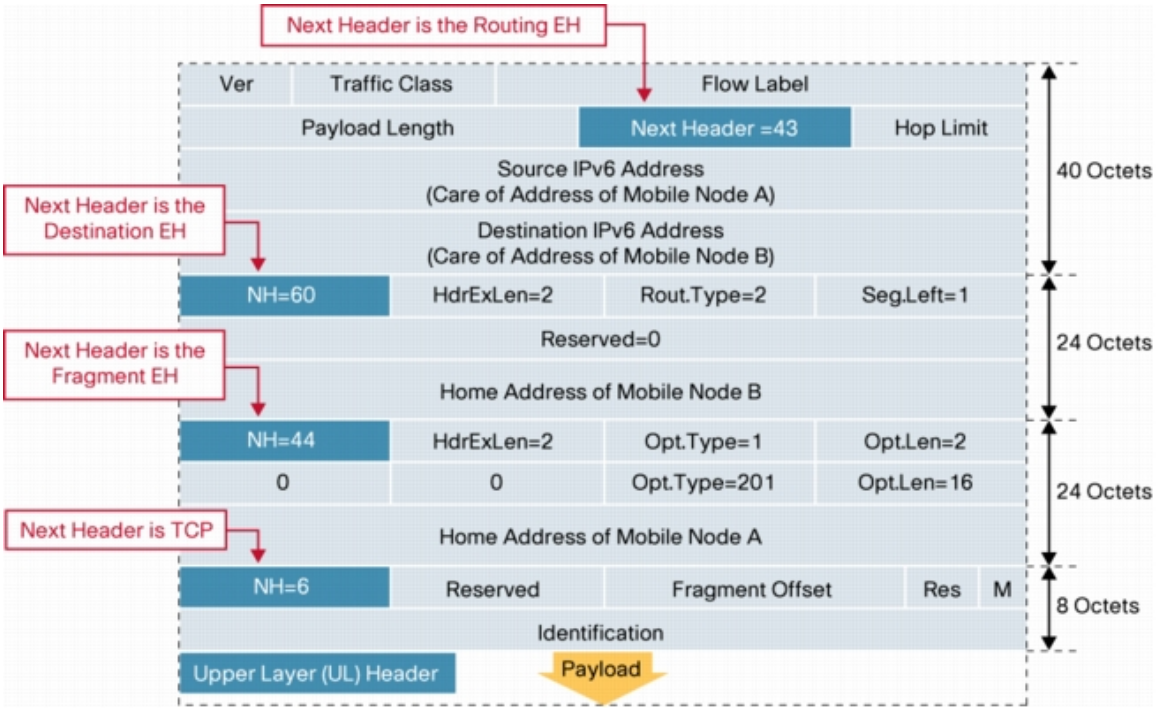

Figure 2.2.: IPv6 extension header (Cisco, 2006)

The other fields in the IPv6 header are, hop limit, which identifies the time-to-live of the packet, the source IPv6 address and the Destination IPv6 address.

RFC 4443 (Conta & Gupta, 2006) defines the specification for ICMPv6 protocol. Figure 2.3 shows an ICMPv6 header along with various ICMPv6 message types. The type field identifies the type of message, the code field identifies a sub-type of the message and the checksum field contains a CRC of the ICMPv6 header. The data field contains the payload of the message.

A ICMPv6 destination unreachable packet is sent by any networking device which does not have the route details to reach the destination, or by the destination

Generic ICMPv6 packet format:

IPv6 Header (40 bytes)		
Type	Code	Checksum
Data		

ICMPv6 principal types and codes :

	Type	Meaning	Code	Code explanation
Errors	1	Destination Unreachable	0	No route to destination
			1	Administratively prohibited
			2	Out of scope
			3	Address unreachable
			4	Port unreachable
	2	Packet Too Big		
	3	Time Exceeded	0	Hop limit exceeded
			1	Fragment reassembly time exceeded
	4	Parameter Problem	0	Erroneous header field
			1	Unrecognized next header type
			2	Unrecognized IPv6 option
Informational	128	Echo Request		
	129	Echo Reply		
	130	Multicast Listener Query		
	131	Multicast Listener Report		
	132	Multicast Listener Done		
Neighbor Discovery	133	Router Solicitation		
	134	Router Advertisement		
	135	Neighbor Solicitation		
	136	Neighbor Advertisement		
	137	Redirect		

Figure 2.3.: ICMPv6 header with message types (NDHU, 2003)

itself, when a particular service is unavailable. A ICMPv6 packet too big message is sent by any networking device who is unable to process the packet owing to its large size. A ICMPv6 time exceeded message is sent by a networking device which is unable to forward the packet since the time-to-live of the packet expired. A ICMPv6 parameter problem packet is sent by a networking device when it faces any problem in processing the packet. A ICMPv6 echo request is often used by the ping program to test the reachability of a destination from the source. A ICMPv6 echo response is the response to a ICMPv6 echo request packet. (Conta & Gupta, 2006) mandate that the ICMPv6 packet be processed at the destination of the packet and

not be modified by any forwarding device, unless for security purposes. This presents a potential for creation of covert channels where the covert data is sent as the payload of the above message types. In addition, Fenner (2006) adds two experimental message types, 253 and 254, which could also be potentially used to transfer covert data.

Although various other fields and headers could be potentially exploited to create covert channels, the current research is restricted to those created by the above methods.

2.5 Reevaluating Covert Channels

Although Lucena et al. (2006) discussed the various covert channels in IPv6 and Murphy (2006) showed a proof of concept of the IPv6 covert channels, major changes have been introduced in IPv6 since 2006, making the research studies potentially obsolete and introducing the need to understand the current scenario. Both Lucena et al. (2006) and Murphy (2006) worked on the basis of the three IPv6-related RFCs that existed in 2006, RFC 2460 (Hinden & Deering, 1998), RFC 3697 (Rajahalme, Conta, & Carpenter, 2004), and RFC 4443 (Conta & Gupta, 2006). According to Lourdes and Hansen (2015), the following are the list of RFC that underwent major updates:

1. RFC 4294 "IPv6 Node Requirements", April 2006 (Jankiewicz, Loughney, & Narten, 2011).

2. RFC 4727 "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", November 2006 (Fenner, 2006)

3. RFC 4884 "Extended ICMP to Support Multi-Part Messages", April 2007 (Bonica, Gan, Tappan, & Pignataro, 2007).

4. RFC 5095 "Deprecation of Type 0 Routing Headers in IPv6", December 2007 (Abley, Savola, & Neville-Neil, 2007; Kaeo, 2007)

5. RFC 5722 “Handling of Overlapping IPv6 Fragments”, December 2009 (Krishnan, 2009).

6. RFC 5871 “IANA Allocation Guidelines for the IPv6 Routing Header”, May 2010 (Arkko & Bradner, 2010).

7. RFC 6437 “IPv6 Flow Label Specification”, November 2011 (Rajahalme et al., 2011).

8. RFC 6564 “A Uniform Format for IPv6 Extension Headers”, April 2012 (Krishnan, Woodyatt, Kline, Hoagland, & Bhatia, 2012).

9. RFC 6935 “IPv6 and UDP Checksums for Tunneled Packets”, April 2013 (Chimento, Eubanks, & Westerlund, 2013).

10. RFC 6946 “Processing of IPv6 Atomic Fragments”, May 2013 (Gont, 2013).

11. RFC 7045 “Transmission and Processing of IPv6 Extension Headers”, December 2013 (Jiang & Carpenter, 2013).

12. RFC 7112 “Implications of Over-sized IPv6 Header Chains”, January 2014 (Bonica, Manral, & Gont, 2014)

Taking into consideration the above changes, the author of this thesis, in an earlier study (Lourdes & Hansen, 2015) aimed at understanding and validating the presence of covert channels in the present day IPv6 implementations. In addition, most of the previous research studies were theoretical, without a proper proof of concept to validate the theory. Thus, this required building a tool that could exploit the flaws in design and implementation of IPv6 and related protocols to create a covert channel, thereby providing proof of concept. We demonstrated the corroboration of such covert channels in IPv6 by building a software called ‘Covertv6’ for the same, and testing against a simulated enterprise network.

2.6 Detecting Covert Channels

We furthered our research towards evaluating the capabilities of various enterprise network security solutions in detecting such covert channels in IPv6, in a new study (Lourdes & Hansen, 2016). We established the following:

1. Certain enterprise firewall and IDS do not detect the tested covert channels in IPv6, out-of-the-box.

2. The same firewall and IDS do not detect the tested covert channels in IPv6, even after enabling all possible features and signatures available at the time of testing.

3. Custom signatures can be created for possibly detecting certain types of covert channels (mainly ICMPv6) by white-listing the most common implementations.

We provided a list of custom signatures for detection of ICMPv6 based covert channels based on:

1. Non-standard payload length of the ICMPv6 echo request/reply packet: Most of the ICMPv6 implementation have a payload length less than or equal to 56bytes. Any value above that could be a possible covert channel.

2. Non-standard payload of the ICMPv6 packet: In the case of the ICMPv6 error messages, the payload is parts of the invoking packet, which would contain most of the IPv6 header. In the case of ICMPv6 echo request/reply, which are most commonly used by the ping6 program, the payload is the time-stamp appended with the alphabets in order or the numerals in increasing order. Any deviations from these patterns could be a possible covert channel.

3. Non-standard sequence of ICMPv6 packets: Most implementations increment the sequence number of the consequent ICMPv6 packets. Also, most common implementation of ICMPv6 echo request/reply packets, ping6, sends 1 packet

per second. Any deviation from this behavior could be a possible covert channel.

We also discussed other detection mechanisms specific to our implementation of the covert channel tool such as:

1. Signature for detecting the magic numbers of most archive file formats such as zip, tar, bzip, gzip, in addition to detection of most common file types such as exe/dll, doc, txt etc., in the payload of a packet.
2. Signature for detecting beacons to/from unknown servers.
3. Signature for detecting random flow labels between a set of communicating hosts.
4. Signature for detecting experimental extension headers as possible covert channels.

Although these detection mechanisms provided a possible means of detecting certain covert channels, there is an imminent need to understand the forensic-soundness of these evidence.

2.7 Defining Network Forensics

Network forensics is the use of various tools, techniques and procedures to capture, record and analyze network traffic after a network attack. It can be considered as an extension of network security and involves collection of data after a network incident, that could be used to investigate more about the attack or that can be used as evidence against the attacker in the court of law. It is different from network security in several different ways. While network security concentrates more on protecting a system against an attack, network forensics is a postmortem investigation of an attack. While a incident response process does a postmortem investigation as well, a network forensics process deals with the admissibility of the

evidence in the court of law (Kent, Chevalier, Grance, & Dang, 2006). This could provide one with valuable information, like the identity/location of the attacker, nature and duration of the attack, the vulnerability that made the attack possible along with several other factors. Apart from helping to narrow in on the attacker, such valuable information would also help prevent similar attacks in the future.

There have been a number of definitions for network forensics. Palmer (2001) defines it as:

> The use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities. (p. 27)

Although, Ranum (1997) was the first to coin the term network forensics and he defined it as, "the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents" (p. 1).

2.8 Need for Forensics

Despite the security measures like firewall, intrusion detection systems and intrusion prevention systems, eventually an attack happens. Given that an attack has happened, the above said systems are not sufficient to penalize the adversary according to law. A set of procedures and tools that could relate the attack to the adversary and collect evidences that could be presented in the court of law are required. Network forensics is the branch of cyber forensics that does just this. Given that a security breach in a company would subvert its reputation and trust in the minds of customers, the companies are working aggressively towards implementing network security and network forensic measures. The ever increasing

volume of attacks and the companiesâĂŹ ever growing concern about security are the main motivation factors behind Network Forensics (Meghanathan, Allam, & Moore, 2010).

2.9 Challenges in Network Forensics

There are several challenges in carrying out network forensics. For instance, the network needs to have the infrastructure and the other components required to conduct a forensic analysis. In many legacy systems, it is harder to incorporate hardware and software required for the same. Almulhem (2009) analyses some of the most prevalent challenges. One of the most important challenges is to ensure data integrity of the collected data. As ensuring the collected evidence's integrity is of prime importance in any forensic procedure, it no different for networks. While ensuring integrity during creating a bit stream image of a hard drive could be easily achieved, it is more complicated in case of computer networks. Another such issue noted by Almulhem (2009) is, ensuring the privacy of the users. While a network is analyzed, it is critical to ensure the privacy of the other users who are connected to the network but are not associated with the investigation. It is also important to keep a track of the various data sources in the network. In cases of huge networks, there might be a huge number of data sources, some of this might also include raw data. While analyzing all the data sources may be desirable, it is not always practically possible. The huge volume of data also creates issues while analyzing. Highly sophisticated tools are required to deal with such huge volume of data (Almulhem, 2009).

Another important challenge as pointed out by Nehinbe (2011) is the intrusion redundancy problem. This problem is also referred to as attack swamping, wherein the attackers cause alarms to go off for no reason and exhaust the capability of the network analysts to carry out an examination and at times, this causes the important threats to go unnoticed. As Nehinbe (2011) points out, apart

from the practical difficulties that arise when trying to implement network forensics, there are also challenges in conducting research in network forensics. As it is the case with any research, better results can be obtained only when there is sufficient and appropriate data to analyze. A network forensic research would usually involve analyzing various attacks and finding ways to find evidence and techniques for bringing the perpetrator to the law. But nowadays, data about more sophisticated attacks are hard to obtain. Victims of such large scale attacks are usually enterprises from whom data cannot be easily obtained for security reasons. A solution for this would be to replicate sophisticated network attacks and conduct research on the data produced during these attacks. But then, replicating these attacks would require huge amount of resources and it is hard for the researchers to think from the perspective of the attackers. Even if the companies agree to provide details regarding the attack, gaining an overall picture from a few attacks is not easy (Nehinbe, 2011). Thus, the current research was done as a continuation of (Lourdes & Hansen, 2015, 2016), where the covert channel replication was done by the former and a detection mechanism was tested by the latter research, due to lack of data in the field of covert channels.

2.10 Forensic Soundness of Evidence

In the attempt to identify forensically-sound evidence to detect covert channels in IPv6, it is important to understand the meaning of forensic soundness of the evidence. One of the profound works in this field is attributed to McKemmish (2008). He defines forensic soundness as: "the application of a transparent digital forensic process that preserves the original meaning of the data for production in a court of law" (p. 10). In his opinion, the forensic soundness of the evidence is attributed to the forensic process used. Thus he proposed four major criteria for assessing the forensic soundness of the process and hence the evidence:

1. Unaffectedness of the meaning and interpretation of the electronic evidence.

2. Identifiability and explanation of all errors.

3. Independent examination and verifiability of the process.

4. Experience of the digital forensic examiner.

In addition, McKemmish (2008) clearly explains that although 'forensically sound' describes the whole forensic process, two objectives have to be met: preserving the state of the data that was acquired and analyzed, and preserving the evidentiary value of the digital data. Thus to satisfy these objectives, he suggested expressing the term 'forensically sound' as a series of steps/procedures and explained that this approach was logical but non-uniformity would lessen its value. Thus, it becomes important to understand that, while a network forensic investigation technique for analyzing covert channels in IPv6/ICMPv6 could be arrived at, by the end of the research, the applicability of this model towards various similar covert channels would still be an uncertainty, as explained by McKemmish (2008). While McKemmish (2008) suggested that evidence needs to preserved as-is, Casey (2007) suggested that this is often not possible and hence defines the forensic soundness in a way that "the acquisition process should change the original evidence as little as possible and any changes should be documented and assessed in the context of the final analytical results" (p. 50).

Although, the previous research studies explain the meaning of forensic soundness, the most often used criteria for evaluation of the admissibility or forensic soundness of an evidence, is the Daubert standard (*Daubert v. Merrell Dow Pharmaceuticals, Inc.*, 1993). According to the Daubert standards, four main factors need to be considered for the admissibility of scientific evidence as follows:

1. Whether the scientific method for obtaining the evidence is testable.

2. Whether the scientific method is peer reviewed and published.

3. The known or potential error rates of the scientific method.

4. Whether the scientific method is generally accepted in the relevant scientific community.

The Mckemmish and Daubert criteria formed the basis for the current research, which aimed at creating a scientific method (digital forensic investigation technique) for analyzing certain covert channels in IPv6/ICMPv6, which is admissible in the court of law, in the United States of America, since the Daubert criteria and Mckemmish criteria hold good in the United States (*Daubert v. Merrell Dow Pharmaceuticals, Inc.*, 1993; McKemmish, 2008). Although, the current research tried to test only two of the common criteria between both the research: the validity of the scientific method, and the known/potential error rates of the methods.

2.11 Digital Forensic Process Models

Digital forensic process models provide a standard structured process which defines a suitable mechanism to be followed by the computer forensic investigators (Yusoff et al., 2011). Since the current research strived to create a digital forensic investigation technique and test it against the forensic soundness criteria, it becomes important to understand the aspects of various digital forensic process models, mainly the network forensic process models. While there are many process models proposed by numerous researchers, Yusoff et al. (2011) provided a detailed comprehensive study of most of them, thereby helping in understanding their commonalities and differences. They analyzed the following digital forensic process models:

1. Computer Forensic Investigative Process (Noblett, Pollitt, & Presley, 2000)
2. DFRWS Investigative Model (Palmer, 2001)
3. Abstract Digital Forensics Model (ADFM) (Reith, Carr, & Gunsch, 2002)
4. Integrated Digital Investigation Process (IDIP) (Carrier, Spafford, et al., 2003)

5. Enhanced Digital Investigation Process Model (EDIP) (Baryamureeba & Tushabe, 2004)
6. Computer Forensics Field Triage Process Model (CFFTPM) (Rogers, Goldman, Mislan, Wedge, & Debrota, 2006)
7. Digital Forensic Model based on Malaysian Investigation Process (DFMMIP) (Perumal, 2009)
8. Scientific Crime Scene Investigation Model (Ciardhuáin, 2004)
9. End to End Digital Investigation (Stephenson, 2003)
10. Extended Model of Cybercrime Investigation (Ciardhuáin, 2004)
11. A Hierarchical, Objective-Based Framework for the Digital Investigations Process (Beebe & Clark, 2005)
12. Framework for a Digital Forensic Investigation (Köhn, Olivier, & Eloff, 2006)
13. Common Process Model for Incident and Computer Forensics (Freiling & Schwittay, 2007)
14. Dual Data Analysis Process (Bem & Huebner, 2007)
15. Network Forensic Generic Process Model (Pilli, Joshi, & Niyogi, 2010)

Upon analysis of all these digital forensic process models, they summarized their findings as below:

1. Five generic phases were common between most of these process models. They are, pre-process, acquisition & preservation, analysis, presentation, and post-process.
2. They proposed a new common process model called, the Generic Computer Forensic Investigation Model (GCFIM).

3. The differences in these process models were in the content of each phase and how detailed they were.

Since the current research strived to create a digital forensic investigation technique, it falls under the analysis phase, which includes performing various techniques of analysis on the acquired data to find the source of crime and the cyber criminal (Yusoff et al., 2011, p. 29).

While there are a number of forensic process models to perform disk level forensics, there was an increased need for process models specific to networks, since the devices were becoming more and more inter-networked. The first ever Digital Forensics Research Workshop was held in 2001 and it came up with a model consisting of the various phases like identification, preservation, collection, examination, analysis, presentation and decision. This model was considered abstract and too generalized. Reith et al. (2002) improvised this model by introducing an abstract model that could be applied to any cyber-crime in general. The concept of incident response was not introduced until Prosise, Mandia, and Pepe (2003). A few other notable ones include Carrier et al. (2003), Ciardhuáin (2004) and Pollitt (2007). Ren (2006) was by far a model that is the most directed towards network forensics. This has the following steps: capture, copy, transfer, analysis, investigation and presentation. Their work also included methodologies for implementation this process model for various types of networks. It should be noted that these models yielded best results when they were able collaborate well with the other security measures like Intrusion Detection Systems.

Thus the current research aimed at building a digital forensic investigation technique using custom IDS signatures and firewall system logs to detect and analyze some of the IPv6 and ICMPv6 covert channels, and test the forensic soundness of the methods using the *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993); McKemmish (2008) criteria of validity and error rates.

2.12 Summary

This chapter provided a detailed review of literature relevant to the current research. It explained what a covert channel is, various research studies on covert channels in IPv4 and IPv6, a clear understanding of the IPv6 and ICMPv6 packets which would be used to create the covert channels, what network forensics is, how to establish forensic soundness of evidence, and some of the existing network forensic models. The next chapter explains framework and methodology of the research.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

3.1 Research Goal

The goal of the research was to understand if certain IDS signatures and firewall system logs, provide forensically sound evidence for detection and analysis of a subset of IPv6 and ICMPv6 covert channels. Hence the digital forensic investigation techniques that were tested for forensic soundness were:

1. Detection of certain covert channels in IPv6 and ICMPv6 using custom IDS signatures.

2. Detection of certain covert channels in IPv6 and ICMPv6 using firewall system logs.

3.2 Research Framework

3.2.1 Overview

The research framework consisted of a software tool that could covertly transfer data through an IPv6 network as described in the network topology in the figure 3.1. This topology was designed similar to the one tested by Murphy (2006) in his proof of concept, as an uncontrolled network environment and proved to be working. The local site designed with a Ubuntu PC (Linux kernel 3.19.0-26) acted as a client that leaks information. A Cisco router (IOS 15.5(1)T) and a pfSense firewall (v2.2.4) with Snort (2.9.7.5 pkg v3.2.7) and Suricata (2.0.8 RELEASE pkg v2.1.6) packages installed, simulated the network on the local site. The pfSense

firewall connected on its WAN side to the IPv4 Internet, created an IPv6 over IPv4 tunnel to a tunnel broker maintained by Hurricane Electric Electric (2015). The remote site consisted of a Ubuntu server (Linux kernel 3.19.0-26) that received the leaked data, and was behind a Cisco router (IOS 15.5(1)T). The Cisco router connected on its WAN side to the IPv4 Internet, created an IPv6 over IPv4 tunnel to Hurricane Electric.

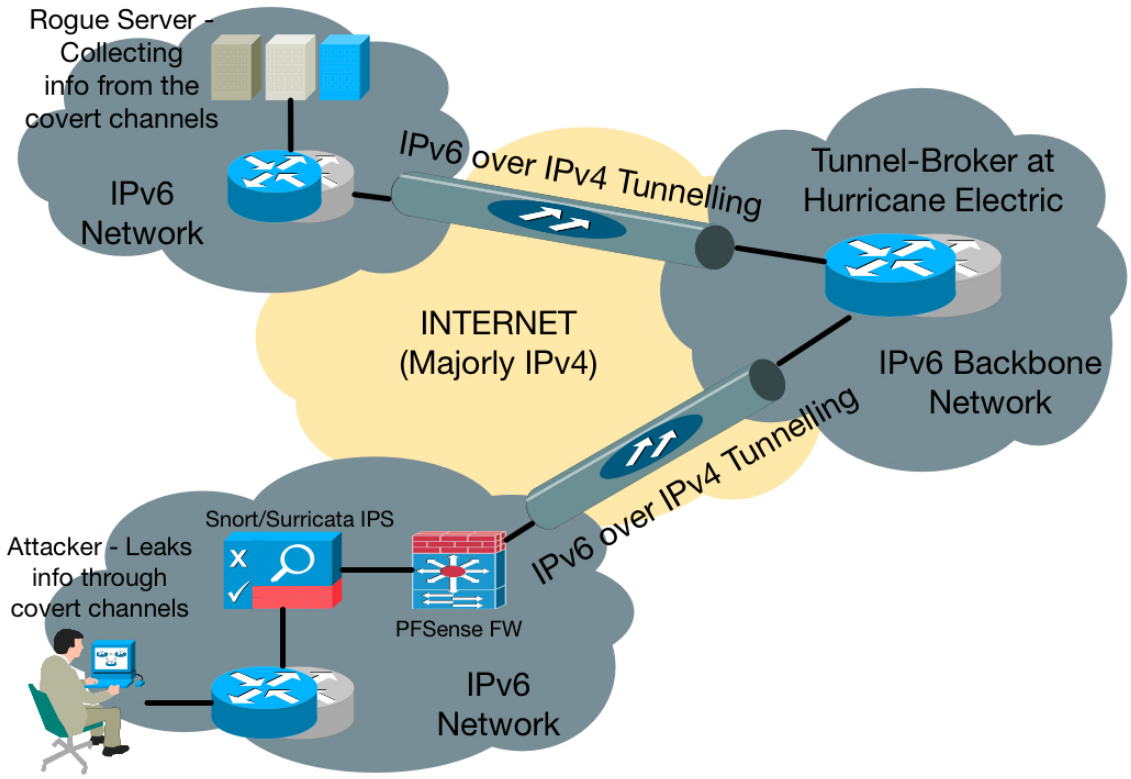

Figure 3.1.: Network topology of the simulation

3.2.2 Configuration of the Firewall and IDS

1. The pfSense firewall was configured with the default access control and deep packet inspection of known protocols. All system logging were enabled with detailed raw logging capability.

2. The Snort IDS, which is a proprietary enterprise IDS solution, was configured with the subscription ruleset, which is "the same Snort ruleset developed for the NGIPS" and used by businesses, according to (Snort, 2015, p. 1). The Snort IDS was configured to use all categories of signatures published in the ruleset 'Talos Rules 2016-06-07' (Snort, 2016). In addition, the signatures described by Lourdes and Hansen (2016) were enabled.

3. The Suricata IDS, "a high performance open-source Network IDS, IPS and Network Security Monitoring engine" (Suricata, 2015b, p. 1) was configured with all features described in Suricata (2015a). The Suricata IDS was configured to use all categories of signatures published in the ruleset along with Snort subscription ruleset 'Talos Rules 2016-06-07' (Snort, 2016). In addition, the signatures described by Lourdes and Hansen (2016) were enabled.

3.2.3 Covertv6 Tool

The client-server model software called "Covertv6", developed by Lourdes and Hansen (2015), using the Scapy library in Python, was used create the covert channels. The client and server initially send specific beacons indicating liveliness and start of data transfer. Then the client archives the file/folder in '.zip' format, to preserve the integrity of the files contained, throughout the binary data transfer process. The client then reads the bitstream of the zipped data and packs into a buffer set, which is then encoded differently based on various exploits as follows:

1. IPv6 Flow Label: The client encodes the 20-bit flow label value in a ICMPv6 echo request packet with the buffered data and sends it to the server.

2. IPv6 Experimental Extension Headers: The client encodes the buffered data into the payload/value field of the experimental extension header with type 253 and 254.

3. ICMPv6: ICMPv6 based exploit is done as below:

(a) Destination Unreachable - The client sends the buffered data as payload in the ‘destination unreachable’ message with ICMPv6 type 1 code 0

(b) Packet Too Big - The client sends the buffered data as payload in the ICMPv6 type 2 code 0 message

(c) Time Exceeded - The client sends the buffered data as payload in the ICMPv6 type 3 code 0 message

(d) Parameter Problem - The client sends the buffered data as payload in the ICMPv6 type 4 code 0 message, with the pointer set to ‘0xFF’

(e) Echo Request - The client sends the buffered data as payload in the ICMPv6 type 128 code 0 message

(f) Echo Reply - The client sends the buffered data as payload in the ICMPv6 type 129 code 0 message

(g) Experimental ICMPv6 - The client sends the buffered data as payload in the ICMPv6 type 253, 254 code 0 message

The server receives the packets and decodes the exploited field and adds it to its buffer for data reconstruction. As soon as all the data is transferred, the client and server send end beacons to signal the end of transfer. The server reconstructs the ‘.zip’ file from the bitstream and extracts the archive to get the transferred data.

3.2.4 IDS Signatures

Both Snort and Suricata IDS were configured with the subscription ruleset and all categories of the signatures were enabled. As shown by Lourdes and Hansen (2016), these signatures were not sufficient to detecting these covert channels.

As described by Lourdes and Hansen (2016), the following list of signatures were created and tested against their corresponding covert channels:

1. Signature 1: Alert if ICMPv6 payload length is greater than 32 Bytes. This was in accordance to most ICMPv6 based tools sending data less that 32 Bytes.
2. Signature 2: Alert if ICMPv6 payload of length 32 Bytes but does not contain the pattern 'abcdefghij'. This was in accordance to most ICMPv6 based tools sending data that contained that pattern.
3. Signature 3: Alert if ICMPv6 echo request or reply contains other extension headers. This was in accordance to most ICMPv6 based tools not using extension headers.
4. Signature 4: Alert if a sequence of ICMPv6 echo request or reply packets between two hosts, within a particular time interval, do not have incremental sequence numbers. This was in accordance to most ICMPv6 based tools sending ICMPv6 packets whose sequence number increments.
5. Signature 5: Alert if a sequence of ICMPv6 echo request or reply packets between two hosts, arrive at a rate not equal to 1 pkt/sec, within a particular time interval. This was in accordance to most ICMPv6 based tools sending data at the rate of 1 pkt/sec.
6. Signature 6: Alert if a beacon (containing start or end of leak) is sent or received to/from external networks. This was in accordance to the architecture of the Covertv6 tool and how it works. It sends a start and end beacon during a covert communication.
7. Signature 7: Alert if flow labels are different within a particular time interval, in a communication between two hosts. This was in accordance to the violation of RFC 3697, which mandates a source and destination to identify its individual flows by a single flow label.

8. Signature 8: Alert if the IPv6 or ICMPv6 packets had experimental extension header values. This was in accordance to most IPv6 and ICMPv6 based implementations not sending experimental headers.
9. Signature 9: Alert if ICMPv6 packets contain the magic numbers of most common file types such as exe/dll, doc, txt etc., in the payload. This was in accordance to the architecture of the Covertv6 tool and how it works.
10. Signature 10: Alert if ICMPv6 packets contain the magic numbers of archive and compression file-formats such as zip, tar, bzip, gzip etc., in the payload. This was in accordance to the architecture of the Covertv6 tool and how it works.

The IDSs were configured to produce alert for these signatures per every flow. A flow is defined as a sequence of packets transferred from a source to a destination, which the source likes to call as a flow. A flow could be all packets in that connection or a stream of media, but may not necessarily be mapped one-to-one (Rajahalme et al., 2004, p. 1). So the IDS produced alerts in the following manner:

1. In the case of signatures that detect IPv6 flow label based covert channels, a single alert was produced as soon as the corresponding signature was fired.
2. In the case of signatures that detect IPv6 experimental extension header based covert channels, a single alert was produced for every packet encountered with the experimental extension header. Since this would overflow the logging system, the IDS were configured to produce just one alert for every 100 packets encountered. So, as soon as the first covert packet with experimental extension header hits the IDS, the corresponding signature was fired and an alert was produced. Then the IDS waited for 99 more packets to produce the second alert.

3. In the case of signatures that detect ICMPv6 based covert channels, the signatures 1, 2, 3, 6 & 8 produced alerts on a per-packet or summary basis, while the signatures 4, 5, 7, 9 & 10 produced alerts on a per-flow basis as described above.

While, not all signatures were fired for every covert channel, the current research considered the custom IDS signatures as a whole system. Hence a hit on one signature was considered a hit for the entire system. Hence the study on which signatures were hit more or less, was out of scope of the research.

3.3 Analytical Procedure

The Covertv6 tool was run to create the above specified covert channels in IPv6 and ICMPv6, under the following four conditions:

1. Condition 1: Varying the size of the leaked data in the covert packets by transferring an ISO file of size 1MB and then 2MB. Time interval between transfer was set to a default value of 1 sec. No noise of overt traffic was introduced. Since the current research did not attempt to test the performance of the system at different covert packet sizes, the values were set to 1MB and 2MB. A higher value is out of scope of the study.

2. Condition 2: Varying the time interval of transfer between each covert packets. The time intervals were, 1 sec, 5 sec, 10 sec and 20 sec. The 1MB ISO file was transferred. No noise of overt traffic was introduced.

3. Condition 3: Varying the time randomness of transfer of the covert packets by randomly generated time intervals between transfer. The 1MB ISO file was transferred. No noise of overt traffic was introduced.

4. Condition 4: Varying the noise level of overt data traffic, by generating ICMPv6 ping packets between the client and server, at the following data

rates: 1Mbps, 10Mbps and 100Mbps. The 1MB ISO file was transferred covertly, with time interval set to 1 sec.

The noise of the overt traffic was used to simulate a normal ICMPv6 flow between two endpoints. This noise was introduced in order to study the error rates of the custom IDS signatures. By no means was this overt traffic representative of a real life data traffic, and the tests were confined to the environment of the Covertv6 tool. 10 iterations of each of above tests were run, in order to establish the validity of the scientific method. The scientific method would be deemed valid if it produced consistent results in all the 10 iterations.

3.4 Data Collection and Analysis

3.4.1 Measuring Error Rates of Custom Signatures

False positive rate and false negative rate of each of the custom signatures enabled above was measured in order to establish the forensic soundness based on the criteria proposed by *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993); McKemmish (2008). It was calculated as follows:

$$FPR = FP/(FP + TN) \tag{3.1}$$

$$FNR = FN/(FN + TP) \tag{3.2}$$

where, FPR is the False Positive Rate, FP is the number of False Positives, TN is the number of True Negatives, FNR is the False Negative Rate, FN is the number of False Negatives and TP is the number of True Positives. Among the four conditions tested above, only condition 4 provided the details needed for the above calculation, since the error rates had to be calculated when overt traffic was flowing along with covert traffic. A regular ICMPv6 ping flow was used as overt data to test each of the 10 signatures since, the covert data is a variation of the overt data. So, the following steps were carried out:

1. Step 1: The overt data containing ICMPv6 ping packets were first run through the IDS with the custom signatures enabled. The number of false positives, false negatives, true positives, and true negatives, were calculated.

2. Step 2: The same overt data was run through along with the covert data, and the same numbers were calculated.

3.4.2 Exploring Evidences Provided by Firewall System Logs

The firewall system logs from the pfSense firewall was carefully explored to observe any potential markers that identify the covert channel exploited. The entire process was recorded and the results of the analysis is presented.

3.5 Threats to Validity

1. The hardware and software capability of the network components pose a threat to validity since the components might behave differently under stress.

2. Not all exploits have to necessarily be implemented the way it is done in the Covertv6 tool. A different method of the same exploit might produce a different result.

3.6 Summary

This chapter provided the framework and methodology to be used in the research study.

CHAPTER 4. RESULTS

All the tests were run successfully as per the methodology and the results were collected and analyzed and presented below. The figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, and 4.7 show the running of the Covertv6 tool, a sample capture of the exploit, configurations of the IDS signatures (including custom signatures), configuration of the firewall system logs, and sample firewall logs, taken during the tests.

```
lginod@ubuntu:~/Desktop$
lginod@ubuntu:~/Desktop$ sudo python covertv6.py
[sudo] password for lginod:
usage: covertv6.py [-h] [-l] [-x] [-i ICMP_OPT] [-e] [-s SRC] -d DST [-n SIZE]
                   [-t TIME_INTERVAL] -o OPMODE -m MODE -f FILENAME
covertv6.py: error: argument -d/--dst is required
lginod@ubuntu:~/Desktop$
lginod@ubuntu:~/Desktop$ sudo python covertv6.py -h
usage: covertv6.py [-h] [-l] [-x] [-i ICMP_OPT] [-e] [-s SRC] -d DST [-n SIZE]
                   [-t TIME_INTERVAL] -o OPMODE -m MODE -f FILENAME

optional arguments:
  -h, --help            show this help message and exit
  -l, --flow_label      Exploit using IPv6 Flow label
  -x, --ext_hdr         Exploit using IPv6 Extension headers
  -i ICMP_OPT, --icmp_opt ICMP_OPT
                        Exploit using ICMPv6 Options: 1. Destination
                        Unreachable 2. Packet too big 3. Time Exceeded 4.
                        Parameter Problem 128. Echo Request 129. Echo Reply
  -e, --encryption      Encryption On/Off
  -s SRC, --src SRC     Source IPv6 address
  -d DST, --dst DST     Destination IPv6 address
  -n SIZE, --size SIZE  Size of the payload (only if -i is set, default
                        1400Bytes)
  -t TIME_INTERVAL, --time_interval TIME_INTERVAL
                        Time interval
  -o OPMODE, --opMode OPMODE
                        Mode of Operation: 0. Client 1. Server 2. Forwarder
  -m MODE, --mode MODE  Mode of Transfer: 0. Transfer 1. Beacon 2. Forward
  -f FILENAME, --filename FILENAME
                        Path of the file to be transferred
lginod@ubuntu:~/Desktop$
lginod@ubuntu:~/Desktop$ sudo python covertv6.py -d 2004::128 -o 0 -m 0 -f doc/ -i 128 -t 1
.
Sent 1 packets.
Exploiting using ICMPv6:
Using ICMPv6 Echo Request
```

Figure 4.1.: Running the Covertv6 tool

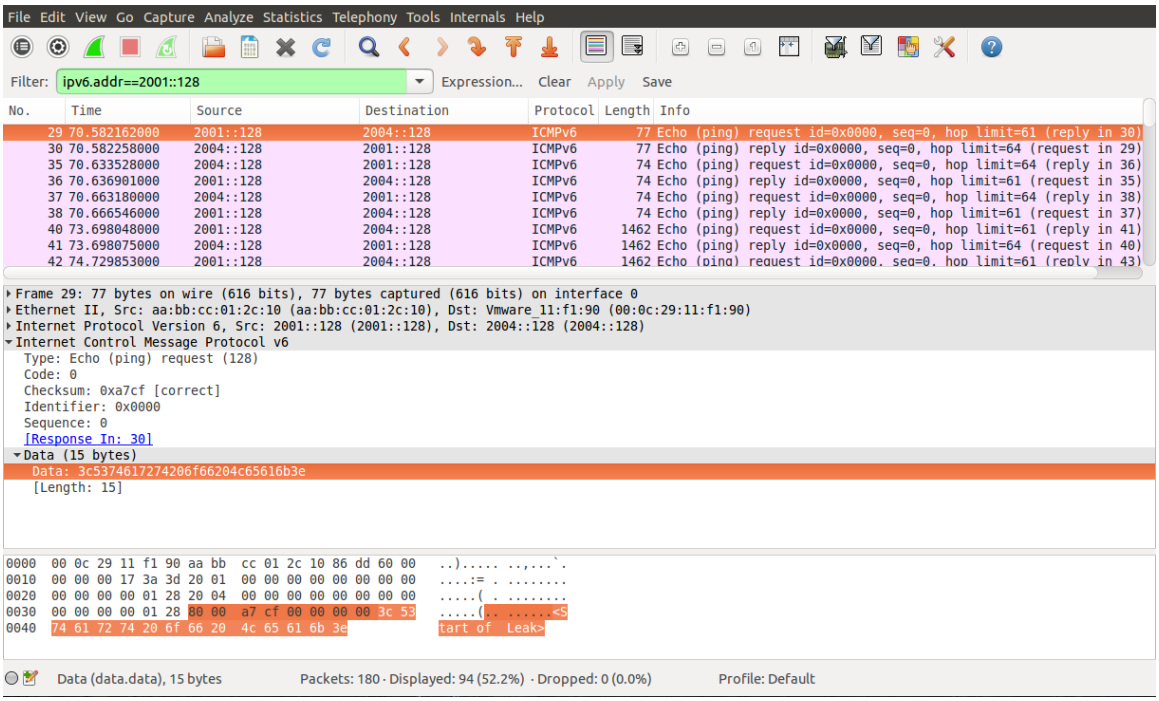

Figure 4.2.: Sample capture of covert data

4.1 Validity of the Scientific Method

4.1.1 Custom IDS Signatures

It was observed that in each of the 10 iterations of the four test conditions, totaling 100 test cases (2x10, 4x10, 1x10, 3x10 tests for conditions 1, 2, 3, and 4 respectively), the corresponding covert channels detected by each of the 10 custom signatures was consistent at a 100% detection rate.

4.1.2 Firewall System Logs

pfSense (2016) clearly explains the various system logs a pfSense firewall generates for the traffic that passes through it. Among these system logs, the ones which were of interest to the current research were the following:

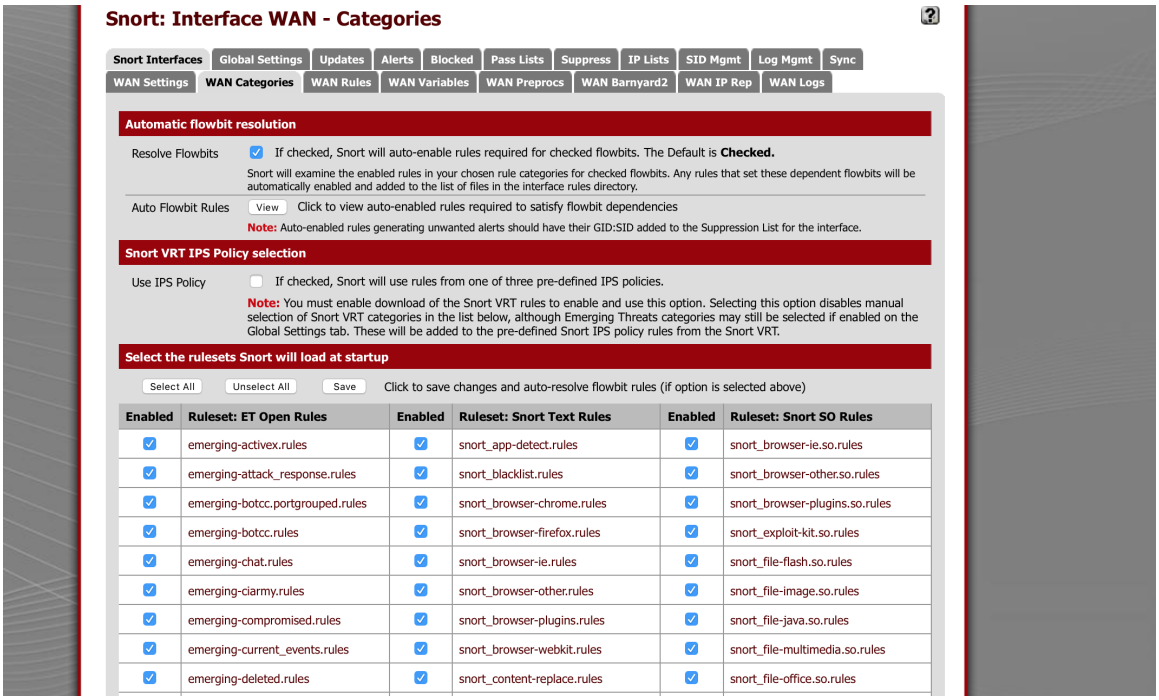

Figure 4.3.: Enabling all signature categories in Snort

1. Log Type 1: <Timestamp> <Hostname> filterlog: <rule_number> <sub_rule_number> <anchor> <tracker> <real_interface> <reason> <action> <direction> <ip_version = 6> <class> <flow_label != 0> <hop_limit> <protocol> <protocol_id> <length> <source_ip> <destination_ip>

2. Log Type 2: <Timestamp> <Hostname> filterlog: <rule_number> <sub_rule_number> <anchor> <tracker> <real_interface> <reason> <action> <direction> <ip_version = 6> <class> <flow_label> <hop_limit> <protocol = other> <protocol_id = 253 or 254> <length> <source_ip> <destination_ip>

3. Log Type 3: <Timestamp> <Hostname> filterlog: <rule_number> <sub_rule_number> <anchor> <tracker> <real_interface> <reason> <action> <direction> <ip_version = 6> <class> <flow_label>

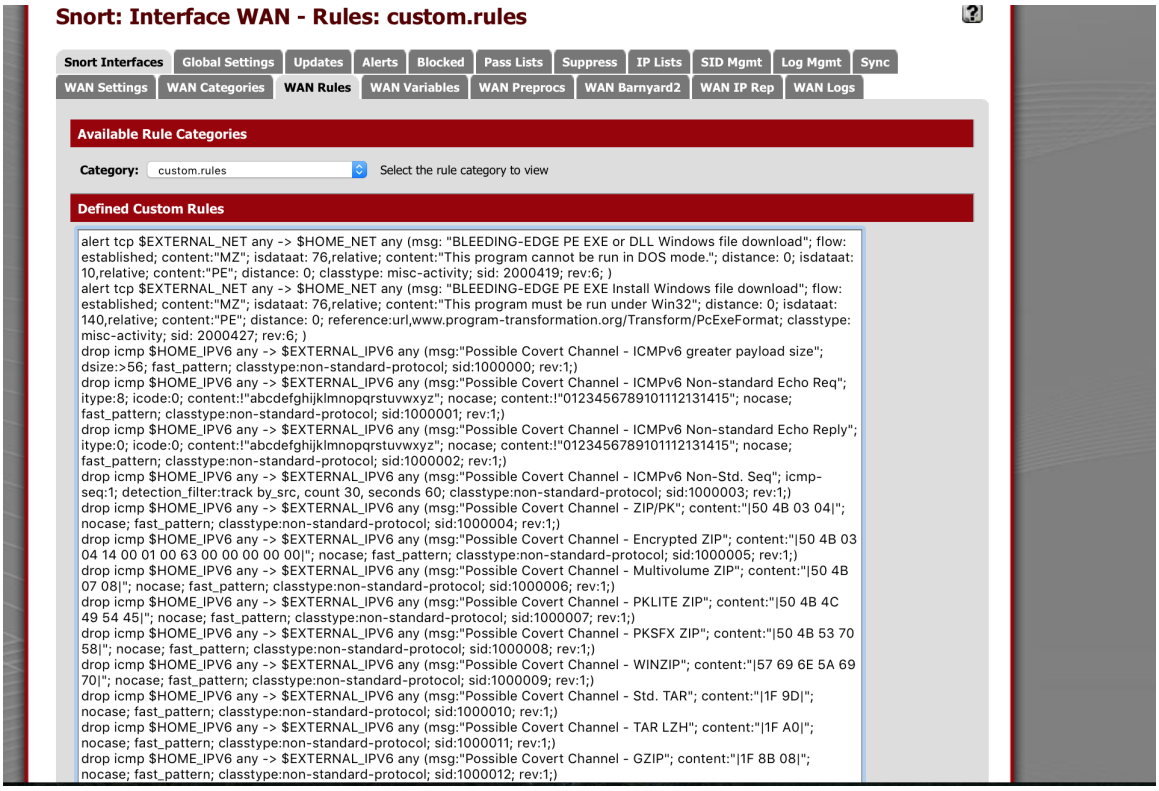

Figure 4.4.: Enabling custom signatures in Snort

<hop_limit> <protocol = ICMPv6> <protocol_id = 58> <length> <source_ip> <destination_ip> <icmp_type = 1> <icmp_dest_id = random_string> <icmp_protocol_id = random_string>

4. Log Type 4: <Timestamp> <Hostname> filterlog: <rule_number> <sub_rule_number> <anchor> <tracker> <real_interface> <reason> <action> <direction> <ip_version = 6> <class> <flow_label> <hop_limit> <protocol = ICMPv6> <protocol_id = 58> <length> <source_ip> <destination_ip> <icmp_type = 253 or 254>

5. Log Type 5: <Timestamp> <Hostname> filterlog: <rule_number> <sub_rule_number> <anchor> <tracker> <real_interface> <reason> <action> <direction> <ip_version = 6> <class> <flow_label>

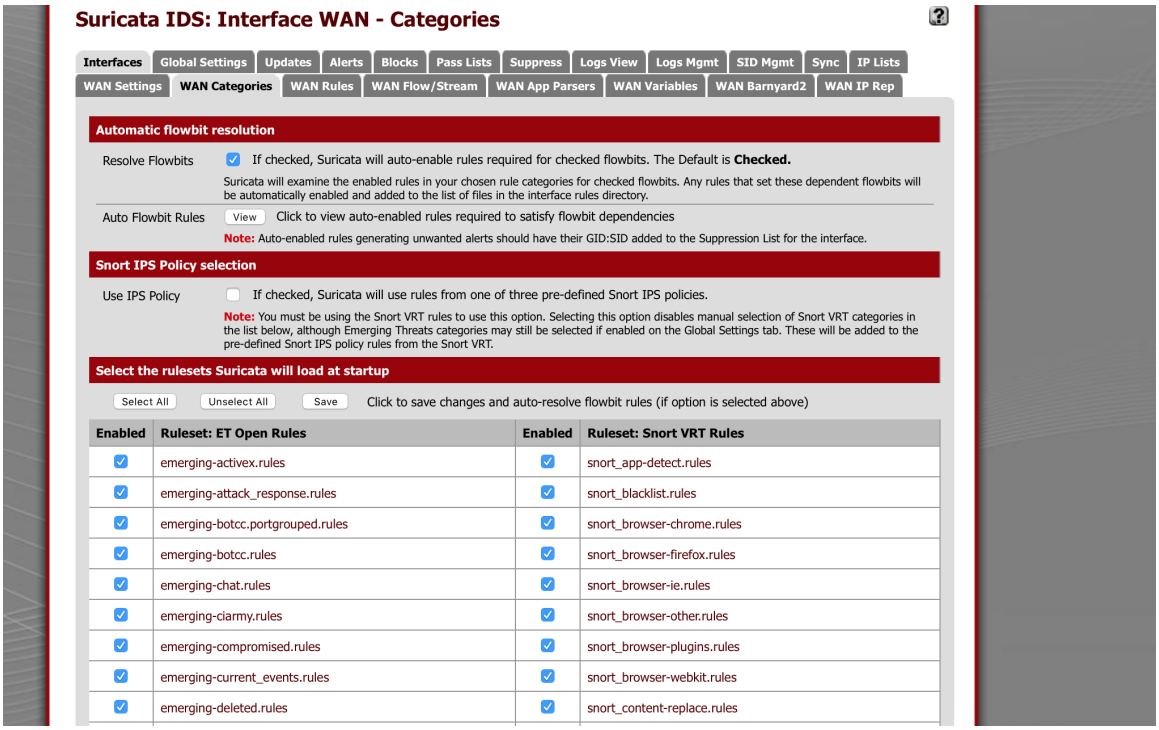

Figure 4.5.: Enabling all signature categories in Suricata

<hop_limit> <protocol = ICMPv6> <protocol_id = 58> <length> <source_ip> <destination_ip> <icmp_type>

The above logs signify the following events:

1. Log type 1 identifies an IPv6 packet with non-zero flow label.
2. Log type 2 identifies an IPv6 packet with experimental extension header.
3. Log type 3 identifies an ICMPv6 destination unreachable packet whose payload is non-standard.
4. Log type 4 identifies an ICMPv6 packet with experimental type value.
5. Log type 5 identifies a generic ICMPv6 packet.

It was observed that in each of the 10 iterations of the four test conditions, totaling 100 test cases, one of the above five log types was produced. Although, log

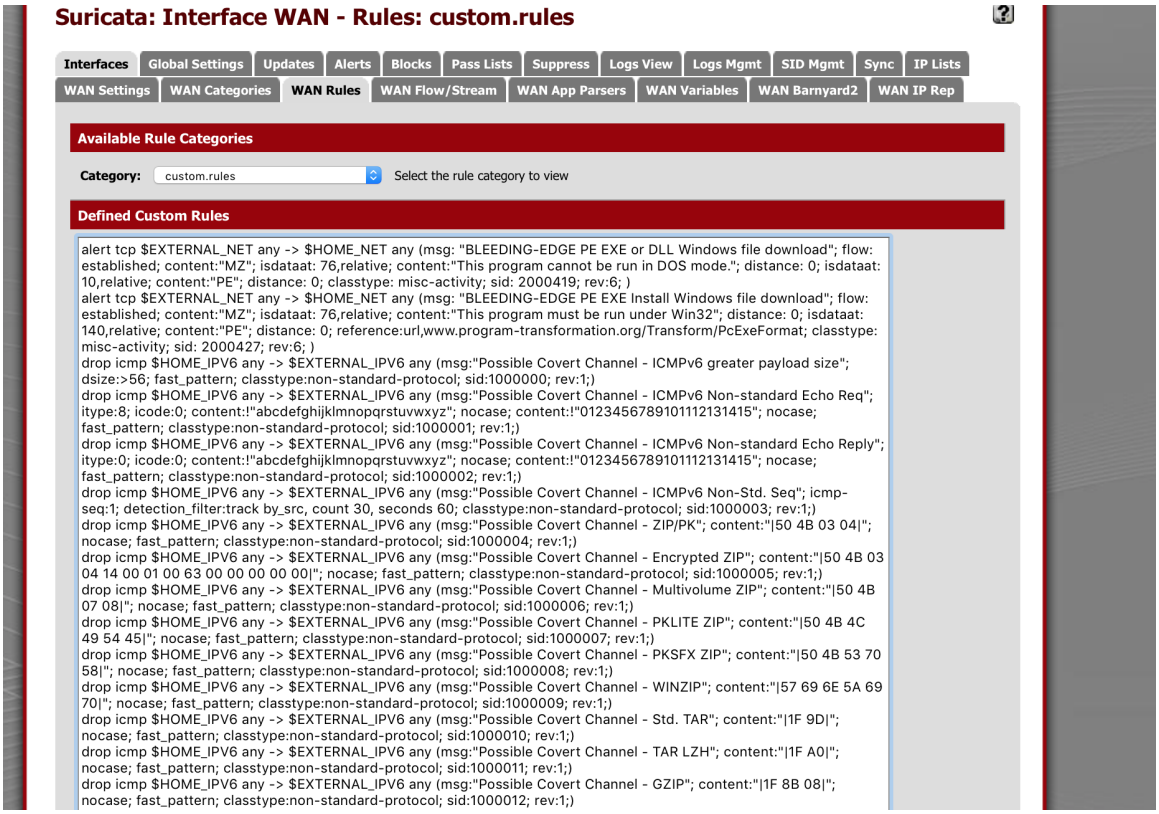

Figure 4.6.: Enabling custom signatures in Suricata

type 5 did not identify any particular covert channel, thereby deeming a detection rate of 80%.

4.2 Error Rates

4.2.1 Custom IDS Signatures

In each of the 10 iterations of the condition 4 tests, totaling 30 (3x10) test cases, the following was observed:

1. Step 1: FP, TP, FN and TN of the overt data was calculated and the values were,

$$FP = 0; TP = 0; FN = 0; TN = 0; \tag{4.1}$$

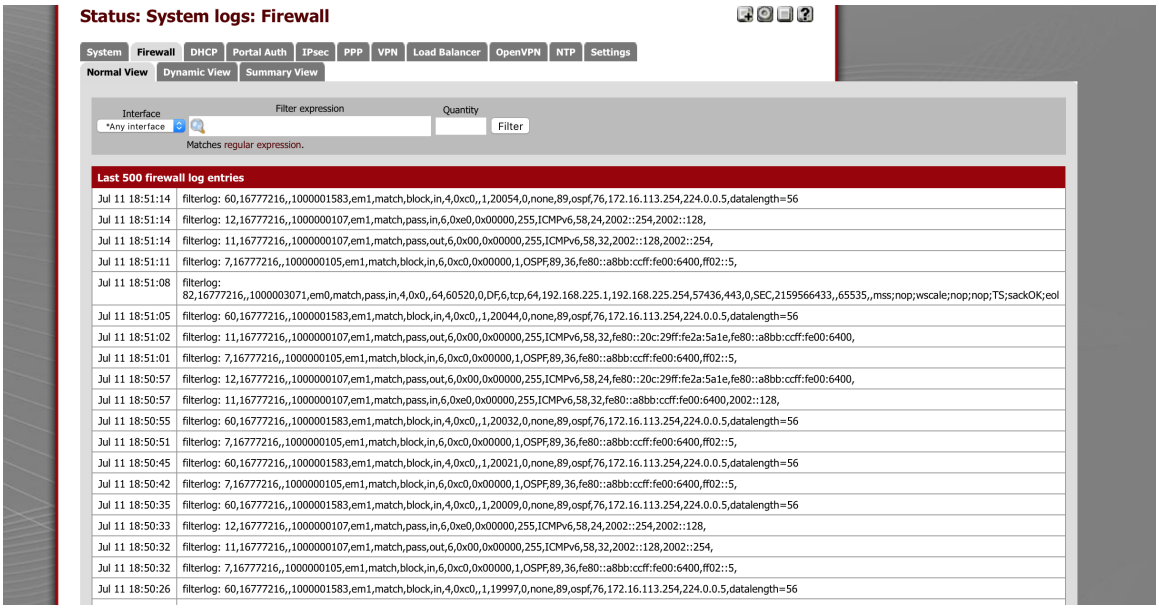

Figure 4.7.: Sample Firewall system logs in pfSense

2. Step 2: FP, TP, FN and TN of the mixed data was calculated and the values were:

$$FP = 0; TP = 0; FN = 0; TN = 0; \tag{4.2}$$

It was observed that the corresponding covert channels detected by each of the 10 custom signatures did not generate any false positives or false negatives.

$$FPR = 0/(0 + TN) = 0\% \tag{4.3}$$

$$FNR = 0/(0 + TP) = 0\% \tag{4.4}$$

Hence, the false positive rate and false negative rates of these custom signatures were 0%.

4.3 Exploratory Analysis of Firewall System Logs

Owing to the exploratory nature of the study with respect to the firewall system logs, the error rates of the detectability of covert channels using the firewall

system logs were not calculated. But, the system logs were analyzed for potential markers for identifying covert channels. The analysis is as below:

1. All the data flow through the pfSense firewall produced detailed system logs as described in pfSense (2016).

2. Log type 1 had a marker with flow label not equal to 0. A log analysis tabling source IP, destination IP and flow label fields was able to identify a flow with varying flow label field between the same source and destination, which could potentially be used to analyze the IPv6 flow label based covert channel.

3. Log type 2 had a marker with protocol id set to 253 or 254 denoting the presence of experimental extension headers in the IPv6 packet. This could potentially be used to analyze the IPv6 experimental extension header based covert channel.

4. Log type 3 had a marker with ICMP destination and protocol set to random values, indicating that the ICMPv6 destination unreachable packet was in non-conformance to the RFC. This could potentially be used to analyze the ICMPv6 destination unreachable based covert channel.

5. Log type 4 had a marker with ICMPv6 type set to 253 or 254 denoting the presence of experimental ICMPv6 types in the ICMPv6 packet. This could potentially be used to analyze the ICMPv6 experimental type based covert channel.

6. Log type 5 did not have any markers. All of the remaining covert channel tested in the study, produced this log type. The absence of markers could be attributed to the design and inability of the firewall to produced deep-packet detail logs.

4.4 Summary

This chapter summarized the results and provided an analysis of the results.

CHAPTER 5. DISCUSSION

The current research successfully created two digital forensic investigation techniques: 'analyzing certain IPv6 and ICMPv6 covert channels using custom IDS signatures' and 'analyzing certain IPv6 and ICMPv6 covert channels using firewall system logs'. The research also successfully tested some of the forensic soundness criteria: the validity of the scientific method, and known/possible error rates, against these two techniques. The research thoroughly studied the results and a clear understanding of the results is presented below.

Each of the 10 iterations of the tests conducted with the custom IDS signatures produced consistent results with respect to the detectability of the covert channels discussed. It was also observed that the detection rate was 100%. This provides evidence of the validity of the scientific method employed, ie., custom IDS signatures. This potentially substantiates the validity of the custom IDS signatures created by Lourdes and Hansen (2016), in identifying and analyzing the IPv6/ICMPv6 covert channels discussed in the study. Although the custom IDS signatures were tested under 4 different operating conditions, the measurement of error rates such as false positive rate and false negative rates could only be calculated when there was a presence of overt traffic along with the covert traffic tested (condition 4). The results showed that in each of the 10 iterations of the tests performed under condition 4, the false positive rate was 0% and the false negative rate was 0%. This could be attributed to the custom tailoring of the signatures to match the exact covert channel or covert channel creation mechanism. Hence, it could be inferred that an understanding of the covert channels could potentially help in identifying patterns that could be used to detect them. Thus the results provide a calculated error rate of 0%, while employing the forensic investigation

technique of custom IDS signatures created by Lourdes and Hansen (2016), in identifying and analyzing the IPv6/ICMPv6 covert channels discussed in the study.

Thus, the custom IDS signatures created by Lourdes and Hansen (2016) was tested against the two common digital forensic soundness criteria between *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993) and McKemmish (2008). The other two criteria of McKemmish (2008), 'reliability' and 'expertise of the forensic examiner' could not be tested as a part of this research. While 'reliability' could only be tested by running the tests multiple times by independent researchers, the 'expertise of the forensic examiner' is purely subjective to the forensic examiner and is independent of the method used. Hence these two criteria were not tested by the current research study. Similarly, the other two criteria of *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993), 'peer review' and 'acceptance in scientific community' could not be tested as a part of this research. Like in the case of McKemmish (2008), *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993) criteria of 'peer review' requires publishing the techniques and a thorough review by peers and experts, and the criteria of 'acceptance in scientific community' would be the result of it. Hence the current research did not attempt to test those criteria. But, once these criteria have been tested, these custom IDS signatures created by Lourdes and Hansen (2016) could potentially be used as digital forensic investigation technique in the digital forensic investigation process.

Each of the 10 iterations of the tests conducted with the pfSense firewall system logs produced consistent results with respect to the detectability of the covert channels discussed. It was also observed that the detection rate was 80%. This provides evidence of the testability of verifiability of the scientific method employed, ie., pfSense firewall system logs. This potentially substantiates the validity of the pfSense firewall system, in identifying and analyzing the IPv6/ICMPv6 covert channels discussed in the study. Although log types 1 through 4 provided potential evidence for the presence of the covert channels tested, log type 5 was a more generic log type and did not provide markers for detection of the

covert channels tested. A detailed study of the log type 5 showed that these markers were not present because the firewall was logging only the headers of IPv6/ICMPv6 packets and not the payload of the IPv6/ICMPv6 packets, which would have provided more insight of the data being carried. In addition, log types 1 through 4, potentially identified the presence of an anomaly while compared with normal traffic. But by no means could they successfully validate the anomaly to be a covert channel. This was one of the major challenges faced by the research while trying to identify the detectability of the firewall system logs.

In order to understand the error rates of the firewall system logs method in detecting the covert channels tested, the research has to be carried out under various test conditions which were out of the scope of this research, owing to the exploratory nature of the research with respect to the firewall system logs. Thus the current research tested only one of the two common criteria between *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993) and McKemmish (2008), 'the validity of the scientific method'. The current research was unable to test the other three criteria of *Daubert v. Merrell Dow Pharmaceuticals, Inc.* (1993) and of McKemmish (2008). Once these criteria have been tested, the firewall system logs could potentially be used as digital forensic investigation technique in the digital forensic investigation process.

Thus, while the current research successfully created two digital forensic investigation techniques for forensically sound analysis of certain covert channels in IPv6 and ICMPv6, the custom IDS signatures and the firewall system logs, the techniques still require additional forensic soundness tests such as peer review of the methods and wide acceptance of the techniques.

5.1 Challenges Faced

Some of the main challenges faced during the study were:

1. Constantly changing network infrastructure due to reliance on the internet to route the traffic from local site to remote site. This was mitigated by creating static infrastructure.

2. Constantly changing software due to upgrades by the product vendors. This was mitigated by fixing the software version to specific versions as discussed in the methodology.

3. Constantly changing configuration, mainly the IDS signatures. This was mitigated by fixing the IDS rule set version to a specific version as discussed in the methodology.

5.2 Future Work

The future work on the research will be a study on encrypted covert channels. The future work will also attempt to test forensic soundness of other network forensic evidences such as flow data, packet captures, and network device databases, towards analyzing the covert channels in IPv6 and ICMPv6. Another potential future work would be to measure the error rates of the firewall system logs while detecting the covert channels in IPv6.

5.3 Summary

This chapter provided a detailed discussion of the research, explaining the challenges faced during the study. This chapter also discussed possible future work.

LIST OF REFERENCES

LIST OF REFERENCES

Abley, J., Savola, P., & Neville-Neil, G. (2007). *RFC 5095 Deprecation of Type 0 Routing Headers in IPv6* (Tech. Rep.). Retrieved from http://tools.ietf.org/html/rfc5095.

Ahsan, K. (2002). *Covert Channel Analysis and Data Hiding in TCP/IP.* Unpublished doctoral dissertation, University of Toronto.

Almulhem, A. (2009). Network Forensics: Notions and Challenges. In *Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on* (pp. 463–466).

Arkko, J., & Bradner, S. (2010). RFC 5871 IANA Allocation Guidelines for the IPv6 Routing Header. Retrieved from http://tools.ietf.org/html/rfc5871.

Baryamureeba, V., & Tushabe, F. (2004). The Enhanced Digital Investigation Process Model. In *Proceedings of the Fourth Digital Forensic Research Workshop* (pp. 1–9).

Baxter, M. (2016, jul). *IPv6 Header.* Retrieved from http://fatpipe.org.

Beebe, N. L., & Clark, J. G. (2005). A Hierarchical, Objectives-based Framework for the Digital Investigations Process. *Digital Investigation*, *2*(2), 147–167.

Bem, D., & Huebner, E. (2007). Computer Forensic Analysis in a Virtual Environment. *International Journal of Digital Evidence*, *6*(2), 1–13.

Blanchet, M. (2009). *Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks.* John Wiley and Sons.

Bonica, R., Gan, D., Tappan, D., & Pignataro, C. (2007). *RFC 4884 Extended ICMP to Support Multi-Part Messages* (Tech. Rep.). Retrieved from http://tools.ietf.org/html/rfc4884.

Bonica, R., Manral, V., & Gont, F. (2014). RFC 7112 Implications of Oversized IPv6 Header Chains. Retrieved from http://tools.ietf.org/html/rfc7112.

Carrier, B., Spafford, E. H., et al. (2003). Getting Physical with the Digital Investigation Process. *International Journal of Digital Evidence*, *2*(2), 1–20.

Casey, E. (2007). What Does "Forensically Sound" Really Mean? *Digital Investigation*, *4*(2), 49–50.

Casey, E. (2011). *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet.* Academic press.

Chimento, P., Eubanks, M., & Westerlund, M. (2013). RFC 6935 IPv6 and UDP Checksums for Tunneled Packets. Retrieved from http://tools.ietf.org/html/rfc6935.

Ciardhuáin, S. Ó. (2004). An Extended Model of Cybercrime Investigations. *International Journal of Digital Evidence*, *3*(1), 1–22.

Cisco. (2006, oct). *IPv6 Extension Headers Review and Considerations.* Retrieved from http://www.cisco.com/en/US/technologies/tk648/tk872 /technologies_white_paper0900aecd8054d37d.html.

Colitti, L., & Gunderson, S. H. (2008). Global IPv6 Statistics-Measuring the Current State of IPv6 for Ordinary Users. In *RIPE 57 Meeting.*

Conta, A., & Gupta, M. (2006). RFC 4443 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. Retrieved from http://tools.ietf.org/html/rfc4443.

Daubert v. Merrell Dow Pharmaceuticals, Inc. (Vol. 509) (No. 92-102). (1993). Supreme Court.

Electric, H. (2015, aug). *Hurricane Electric Tunnel Broker.* Retrieved from https://tunnelbroker.net.

Fenner, B. (2006). RFC 4727 Experimental Values in IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers. Retrieved from http://tools.ietf.org/html/rfc4727.

Freiling, F., & Schwittay, B. (2007). A Common Process Model for Incident Response and Digital Forensics. *Proceedings of the IMF2007*.

Gont, F. (2013). RFC 6946 Processing of IPv6 Atomic Fragments. Retrieved from http://tools.ietf.org/html/rfc6946.

Google. (2016, jul). *Google IPv6 Statistics.* Retrieved from https://www.google.com/intl/en/ipv6/statistics.html.

Graf, T. (2003). *Messaging Over IPv6 Destination Options.* Retrieved from http://gray-world.net/papers/messip6.txt.

Hinden, R., & Deering, S. (1998). RFC 2460 Internet Protocol, Version 6 (IPv6) Specification. Retrieved from http://tools.ietf.org/html/rfc2460.

Jankiewicz, E., Loughney, J., & Narten, T. (2011). RFC 6434 IPv6 Node Requirements. Retrieved from http://tools.ietf.org/html/rfc6434.

Jiang, S., & Carpenter, B. (2013). RFC 7045 Transmission and Processing of IPv6 Extension Headers. Retrieved from http://tools.ietf.org/html/rfc7045.

Kaeo, M. (2007, may). *IPv6 Routing Header Security - RIPE54 - Tallinn, Estonia.* Retrieved from http://meetings.ripe.net/ripe-54/presentations/IPv6_Routing_Header.pdf.

Kent, K., Chevalier, S., Grance, T., & Dang, H. (2006). Guide to Integrating Forensic Techniques Into Incident Response. *NIST Special Publication*, 800–86.

Köhn, M., Olivier, M. S., & Eloff, J. H. (2006). Framework for a Digital Forensic Investigation. In *ISSA* (pp. 1–7).

Krishnan, S. (2009). RFC 5722 Handling of Overlapping IPv6 fragments. Retrieved from http://tools.ietf.org/html/rfc5722.

Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., & Bhatia, M. (2012). *RFC 6564 A Uniform Format for IPv6 Extension Headers* (Tech. Rep.). Retrieved from http://tools.ietf.org/html/rfc6564.

Lampson, B. W. (1973). A Note on the Confinement Problem. *Communications of the ACM*, *16*(10), 613–615.

Latham, D. C. (1986). Department of Defense Trusted Computer System Evaluation Criteria. *Department of Defense*.

Llamas, D., Allison, C., & Miller, A. (2005). Covert Channels in Internet Protocols: A Survey. In *Proceedings of the 6th Annual Postgraduate Symposium about the Convergence of Telecommunications, Networking and Broadcasting, PGNET* (Vol. 2005).

Lourdes, G. D., & Hansen, R. A. (2015). Network Forensics of Covert Channels in IPv6. In *Proceedings of the 16th Annual Information Security Symposium* (p. 26).

Lourdes, G. D., & Hansen, R. A. (2016, mar). *Capability Analysis & Signature-based Detection of IPv6 Covert Channels.* (This research was done as a part of an Independent Study (CNIT 590) under Prof. Raymond Hansen, at Purdue University, IN, USA.)

Lucena, N. B., Lewandowski, G., & Chapin, S. J. (2006). Covert Channels in IPv6. In *Privacy Enhancing Technologies* (pp. 147–166).

Marcella Jr, A., & Greenfield, R. S. (2002). *Cyber Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes*. CRC Press.

McCusker, R. (2006). Transnational Organised Cyber Crime: Distinguishing Threat from Reality. *Crime, Law and Social Change*, *46*(4-5), 257–273.

McKemmish, R. (2008). When is Digital Evidence Forensically Sound? In *IFIP International Conference on Digital Forensics* (pp. 3–15).

Meghanathan, N., Allam, S. R., & Moore, L. A. (2010). Tools and Techniques for Network Forensics. *arXiv preprint arXiv:1004.0570*.

Murphy, R. (2006). *IPv6/ICMPv6 Covert Channels.* DEFCON.

NDHU. (2003, jan). *ICMPv6 Message Types.* Retrieved from https://6book.ndhu.edu.tw/lab_html/packet_capture.htm.

Nehinbe, J. O. (2011). Emerging Threats, Risks and Mitigation Strategies in Network Forensics. In *Electrical and Computer Engineering (CCECE), 2011 24th canadian conference on* (pp. 001228–001232).

Noblett, M. G., Pollitt, M. M., & Presley, L. A. (2000). Recovering and Examining Computer Forensic Evidence. *Forensic Science Communications*, *2*(4), 1–13.

Oppliger, R. (1997). Internet Security: Firewalls and Beyond. *Communications of the ACM*, *40*(5), 92–102.

Palmer, G. (2001). A Road Map for Digital Forensic Research. In *First Digital Forensic Research Workshop, Utica, New York* (pp. 27–30).

Perumal, S. (2009). Digital Forensic Model Based on Malaysian Investigation Process. *International Journal of Computer Science and Network Security*, *9*(8), 38–44.

pfSense. (2016, jul). *Filter Log Format for pfSense 2.2.* Retrieved from https://doc.pfsense.org/index.php/Filter_Log_Format_for_pfSense_2.2.

Pilli, E. S., Joshi, R. C., & Niyogi, R. (2010). Network Forensic Frameworks: Survey and Research Challenges. *Digital Investigation*, *7*(1), 14–27.

Pollitt, M. M. (2007). An Ad-hoc Review of Digital Forensic Models. In *Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Second International Workshop on* (pp. 43–54).

Prosise, C., Mandia, K., & Pepe, M. (2003). *Incident Response & Computer Forensics*. McGraw-Hill/Osborne.

Rajahalme, J., Amante, S., Jiang, S., & Carpenter, B. (2011). RFC 6437 IPv6 Flow Label Specification. Retrieved from http://tools.ietf.org/html/rfc6437.

Rajahalme, J., Conta, A., & Carpenter, B. (2004). RFC 3697 IPv6 Flow Label Specification. Retrieved from http://tools.ietf.org/html/rfc3697.

Ranum, M. J. (1997). Network Forensics and Traffic Monitoring. *COMPUT SECUR J*, *13*(2), 35–39.

Reith, M., Carr, C., & Gunsch, G. (2002). An Examination of Digital Forensic Models. *International Journal of Digital Evidence*, *1*(3), 1–12.

Ren, W. (2006). Modeling Network Forensics Behavior. *Journal of Digital Forensic Practice*, *1*(1), 57–65.

Rogers, M. K., Goldman, J., Mislan, R., Wedge, T., & Debrota, S. (2006). Computer Forensics Field Triage Process Model. In *Proceedings of the Conference on Digital Forensics, Security and Law* (p. 27).

Rowland, C. H. (1997). Covert Channels in the TCP/IP Protocol Suite. *First Monday*, *2*(5).

Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). *NIST Special Publication*, *800*(2007), 94.

Schaeffer, R. (2010). *National Information Assurance (IA) Glossary.* CNSS Secretariat, NSA, Ft. Meade.

Shah, G., Molina, A., Blaze, M., et al. (2006). Keyboards and Covert Channels. In *Usenix Security* (Vol. 6, pp. 59–75).

Snort. (2015, aug). *Snort - Explanation of Rules.* Retrieved from https://www.snort.org/rules_explanation.

Snort. (2016, jun). *Talos Rules 2016-06-07.* Retrieved from https://www.snort.org/advisories/talos-rules-2016-06-07.

Stephenson, P. (2003). A Comprehensive Approach to Digital Incident Investigation. *Information Security Technical Report*, *8*(2), 42–54.

Suricata. (2015a, aug). *Complete List of Suricata Features.* Retrieved from http://suricata-ids.org/features/all-features/.

Suricata. (2015b, aug). *Suricata IDS.* Retrieved from http://suricata-ids.org.

Yusoff, Y., Ismail, R., & Hassan, Z. (2011). Common Phases of Computer Forensics Investigation Models. *International Journal of Computer Science & Information Technology*, *3*(3), 17–31.