ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/337304290

Covert Channel and Data Hiding in TCP/IP

Technical Report · November 2019

CITATION
1

READS
4,638

1 author:

Roshan Poudél

6 PUBLICATIONS 5 CITATIONS

SEE PROFILE

All content following this page was uploaded by Roshan Poudél on 09 August 2023.

The user has requested enhancement of the downloaded file.

Covert Channel and Data Hiding in TCP/IP

Author: Roshan Poudel

1.0 Abstract/Summary:

Confidentiality, Integrity and Availability are the primary objective of information Security. However, over the years information security researchers has found several techniques of data hiding on various systems that are widely used. One technique of data hiding covert channel, which emphasizes to hide information on a system, processes, **TCP/IP** Networking protocols which is against the computer policies and rules regulation. **TCP/IP** a protocol suite for specifying the standards for data transmission and communication between computers was found to be vulnerable to covert channel attack. These covert channels often utilize standard protocol loopholes often called as Network steganography. Further, I have developed a custom python script for **POC** purpose that can send, parse and detect covert channel which can be handy weapon for every forensic investigator. Besides the report, Practical approach has been taken into consideration for which a section for Proof of Concept is allocated at the end of this paper where covert channel over **DNS** as well as mitigating measure is demonstrated practically.

Key Words:

TCP/IP, Data hiding, covert channel, protocols, network tunneling

Abbreviation:

POC	Proof of Concept
TCP/IP:	Transport Control Protocol/Internet Protocol
DNS:	Domain Name System
HTTP	Hyper Text Transport Protocol
ICMP	Internet Control Management Protocol
LAN	Local Area Network
ARP	Address Resolution Protocol

Table of Contents:

Chapter 1: Introduction 6

1.0 Introduction 1

1.2 Aims and Objectives: 2

__1.2.1 Objectives: 2

1.3 Scope and Deliveries of report 3

Chapter 2: Background and Literature Review 4

2.1 Background 5

2.2 Previous Researches on Covert Channel and TCP/IP: 5

2.3 Contribution on Topic by Previous Researchers: 6

2.4 Case Study 7

2.4.1 Red Team's Best Friend: Covert Channel 7

2.4.2 Review 7

2.5 Data Hiding and TCP/IP: 8

2.5.1 Network Steganography 8

2.5.1.1 Covert Channel……………..…………………...…………………………………….8

2.5.2 TCP/IP Based Data Hiding 9

2.5.3 Proof of Concept on DNS Protocol of Application Layer 11

2.5.4 Detecting Covert Channel 15

Chapter 3:Conclusion And Ethics 16

3.1 Conclusion and Further Work 17

3.2 Ethical and Social Issues: 17

Chapter 4: Reference... 18

4.1 References 18

Chapter 5: Appendix... 20

5.1 POC Codes: 23

Table of Figures:

Figure 1: Overt Channel & Operation of Covert Channel..2

Figure 2: Basic TCP/IP header structure (src: Rowland)... 6

Figure 3: Network Steganography (source: dark-reading,2016)..8

Figure 4: Covert Channel Operation (source: Kamran Ashan, Toronto University 2002).............9

Figure 5: TCP/IP Stack and Protocols (source: Oracle crop, 2019)...9

Figure 7: Scenario for POC of DNS Covert Channel..12

Figure 8: Creating a DNS Packet with Python Scripting.. 13

Figure 9: Secret Message sniffed from Wireshark in Receiver..14

Chapter 1: Introduction

1.0 Introduction

Information Security has now become everyone’s prerequisite, either directly or indirectly connected with network environment. The recent research and development have provided us sophisticated computer networks, systems and complex software. With the development of sophisticated systems continue the question for the security of those system arises. As news of data breach come in daily basis, Security has become the hot topic in information security.

One of the modern developments in computer networks have provided us with **TCP/IP** stack which specifies the procedures for communication between the computer networks. Virtually all the large networks and protocols, including internet are built based on **TCP/IP**protocol suite. Protocols such as IPsec, SSH, SSL, TLS are used to provide confidentiality and integrity between network communications. With the research on **TCP/IP**it was found that TCP/IP was vulnerable to various types of attacks like **SYN**Flood attack, Sniffing, Session Hijacking, ping of death, IP spoofing, Data hiding and many more. Data hiding in TCP/IP is possible by creating covert channel in protocols to send confidential information. Data hiding in various protocols is possible due to the loopholes present in their design architecture. The previous researches from various sources on this topic is consulted and data hiding via **DNS**protocol is shown practically to demonstrate how these widely used systems can be used by red team for Penetration Testing as well as also by cybercriminals to transfer the data in unauthorized way resulting the era of digital Crime Evolution.

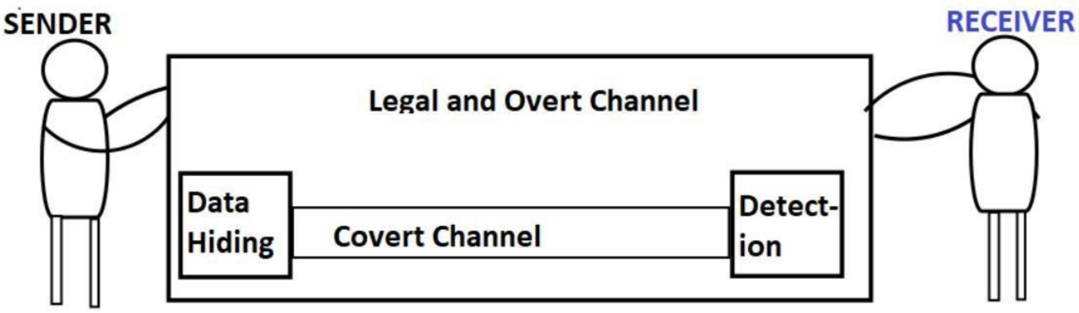


Figure 1: Overt Channel & Operation of Covert Channel

1.2 Aims and Objectives:

Our research paper focuses to elaborate network covert channels that arise in distributed **TCP/IP** systems even when the transmission lines between network nodes are controlled.

1.2.1 Objectives:

- Extensive study and research about the **TCP/IP** stack with the help of research papers, journals, book from various sources like **IEEE**, **SANS**, research gate.
- Evaluation, analysis and contribution by several researchers on the same topic domain.
- Adequate explanation about the layers of **TCP/IP** model and loopholes present on various protocols like **ICMP**, **DNS**, **TCP** which made them victim of covert channel attack.
- Depth explanation on mitigating measures of covert channel attack that has become a crucial part of every digital forensics investigator.
- Elaborate the real time case study on topic domain to prove how covert channel can be used for Penetration Testing
- Practical, demonstration of Covert Channel as Proof of Concept on widely used protocol to explain how widely used systems can be used by to transfer secret information.
- Application of Covert Channel for Red Team during Penetration Testing with the help of Case Study

1.3 Scope and Deliveries of report

Network Security is the indisputable part of computer Networks. Network Security operates in the principle of computer networks and TCP/IP protocol stack. According to Internet World Stat, the global digital crime is increasing by 5% every year. Covert Channel channel is the major factor contributing in this number. Forensic investigators are struggling to tackle the attack which provides report has excellent target audience among security researchers and forensic investigators. Hence, this paper delivers critical analysis of data hiding in TCP/IP and detecting measures.

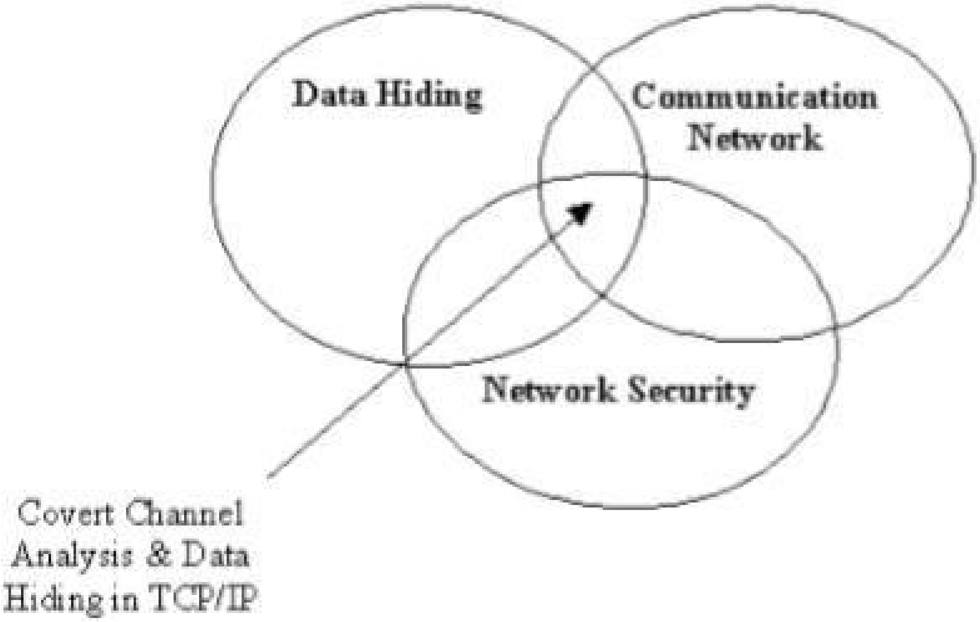


Figure 2: Scope of covert channel and data hiding in TCP/IP (Kamran Ahsan, 2007

Chapter 2: Background and Literature Review

2.1 Background

Data Hiding techniques have been used for ages and they date back to ancient Greece. The aim of Data Hiding communication back then and now, in modern applications, is the same: to hide secret data in an innocently looking cover and send it to the proper recipient who is aware of the information hiding procedure. In an ideal situation the existence of hidden communication cannot be detected by third parties. The way that people communicate evolved over ages and so did stenographic methods. At the same time, the general principles remained unchanged.

Covert Channels have been defined for the first time by Lampson in 1973 as a communication channel not designed for any in 1984 Simmons introduces the concept of subliminal channel through steganography. This concept, applied to the case of two prisoners exchanging sensitive information about how to escape through an open channel.

(source: stegano, 2018)

2.2 Previous Researches on Covert Channel and TCP/IP:

Katzenbeisser and Petitcolas, have also observed the potential for data hiding in the TCP/IP protocol suite. The significance of using of TCP/IP stems from the sheer volume of secret communication that can be realized since TCP/IP packets are used to transport thousands of Internet packets with each overt communication link. Katzenbeisser and Petitcolasuse the term Internet steganography for this potential scenario and indicate that the ongoing research work includes the embedding, recovering and detecting information in TCP/IP packet headers.

The results offered by Wolf (2008), can be observed as a logical extension of but used with LAN protocols. Wolf institutes the point that encryption, which is used for LAN security, cannot safeguard the suitable blocking of unlawful info through the covert channels. The thesis denotes that in each system where shared resources are used the existence of covert channels can be expected. Author highlights the association between protocol format and covert storage channels as well as the relationship between protocol technique elements and covert timing channels by considering frame layouts of the protocols. Padding field, the reserved fields and unused fields of the frame are used by the Covert storage channels.

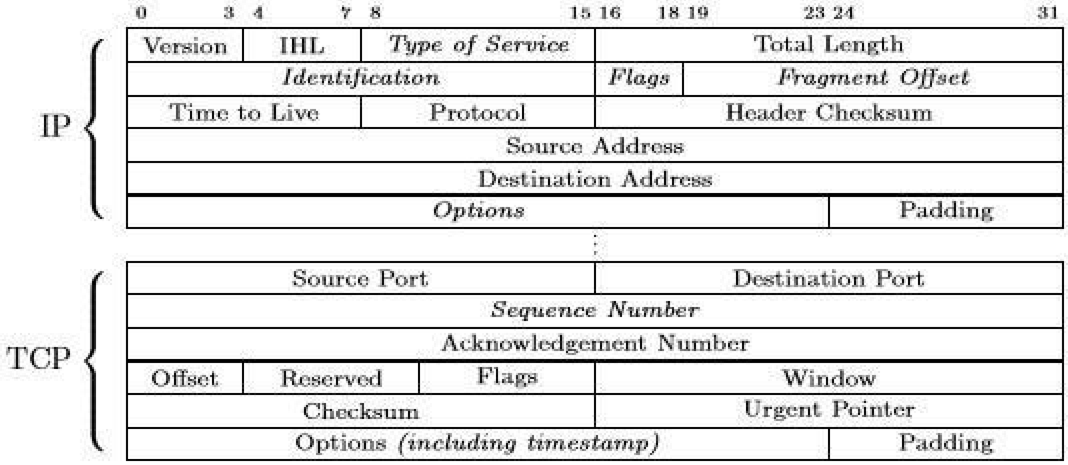


Figure 3: Basic TCP/IP header structure (src: Rowland)

A more specific approach is adopted by Rowland (2010). Rowland developed suitable encrypting and decrypting techniques by using the fields such as the **TCP** initial sequence number, IP identification field, and acknowledgement field, sequence number fields. Rowland delivered an idea of the presence as well as the manipulation of covert channels in **TCP/IP** protocol suite. Moreover, TCP/IP have various vulnerabilities like IP Spoofing, SYN Flood, Session Hijacking, Data Hiding etc.

2.3 Contribution on Topic by Previous Researchers:

- Identification of Covert Channel on protocols like **TCP, ICMP, DNS** working on transport and network layer of TCP/IP by Serdar Cabuk from Purdue University.
- The various protocols in **TCP/IP** are vulnerable to attacks like IP Spoofing, Session Hijacking, Data Hiding, SYN Flood attacks etc. concluded from research conducted by Rowland.

- After the research Ser concluded that covert channels are then used to leak sensitive information to unauthorized parties.

2.4 Case Study

A real time incident from IBM X-Force Red Team is collected to show how Covert Channel can help red team during Penetration Testing of an Organization.

2.4.1 Red Team's Best Friend: Covert Channel

Once **IBM** X-Force Red Team were tasked with delivering a malicious payload to network users without setting off security controls or alerting the defensive team. From our red team's perspective, Firefox File Sharing was a good fit for sending encrypted files. Firefox File Sharing allows for large file sizes up to 1 GB, which is large enough an allowance to both send a payload and exfiltrate data. This answered our need for a siloed, covert channel into the organization. It would encrypt and decrypt the payload for us with an AES-GCM algorithm directly in the internet browser, yet we won't have to deal with any key generation or distribution. The payload would evade the inspection of intercepting proxies that can unwrap Transport Layer Security (TLS), and would remain private and won't be shared with any party along the way, including Mozilla.

(source: IBM Security Intelligence, 2018)

2.4.2 Review

The IBM X Force Team take the advantage of Covert Channel during Penetration Testing by delivering encrypted file without upsetting the Blue Team. The red team successfully delivered the file with the advantage of covert channel to port 80. This concludes that Red teams have the advantage of only needing to find one way in, while blue teams are tasked with securing all ways in and out. This one-sided advantage means that defenders have to keep a close eye on attack tactics, techniques and procedures and expect encryption before covert channels are used by cyber criminals conduct crime through it.

2.5 Data Hiding and TCP/IP:

2.5.1 Network Steganography

Network Steganography is art of Hiding secret data inside the digital media like image, vi *many more*(InfoSec, 2017). More specifically, Network Steganography emphasizes in hiding data, information inside networking protocols, processes and systems. Network Steganography is often used during Penetration Testing as well as by Cyber Criminals for transmitting Un-authorized data.

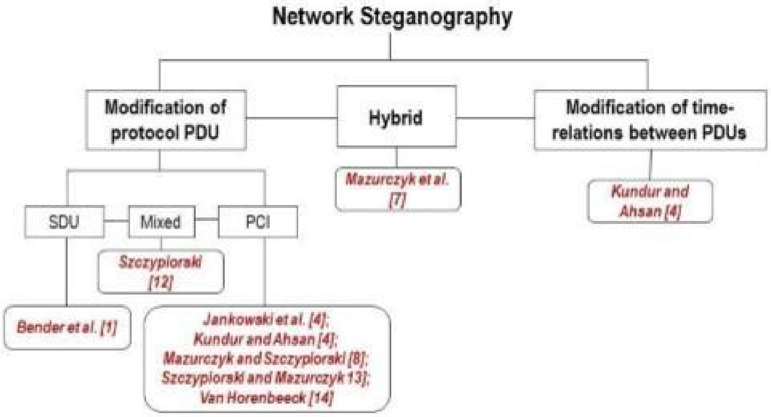

Network Steganography
Modification of protocol PDU
Hybrid
Modification of time-relations between PDUs
SDU
Mixed
PCI
Mazurczyk et al. [7]
Kundur and Ahsan [4]
Szczypiorski [12]
Bender et al. [1]
Jankowski et al. [4]; Kundur and Ahsan [4]; Mazurczyk and Szczypiorski [8]; Szczypiorski and Mazurczyk 13]; Van Horenbeeck [14]

Figure 4: Network Steganography (source: dark-reading,2016)

2.5.1.1 Covert Channel

Covert channel is one of the main sub-disciplines of data hiding. Covert channel tools are not attack tools; they are not used to break into systems. In some scenarios, they can be used alongside an exploit tool to information off a system or to secretly deliver attacks to a system. There are many existing techniques available for development of covert channels by manipulating certain fields in the protocols such as HTTP, ICMP, TCP, etc.

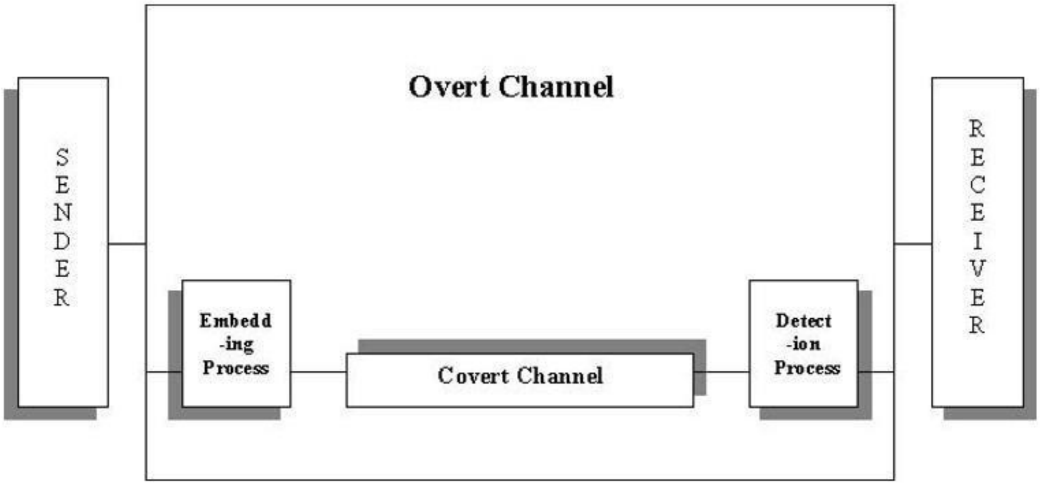
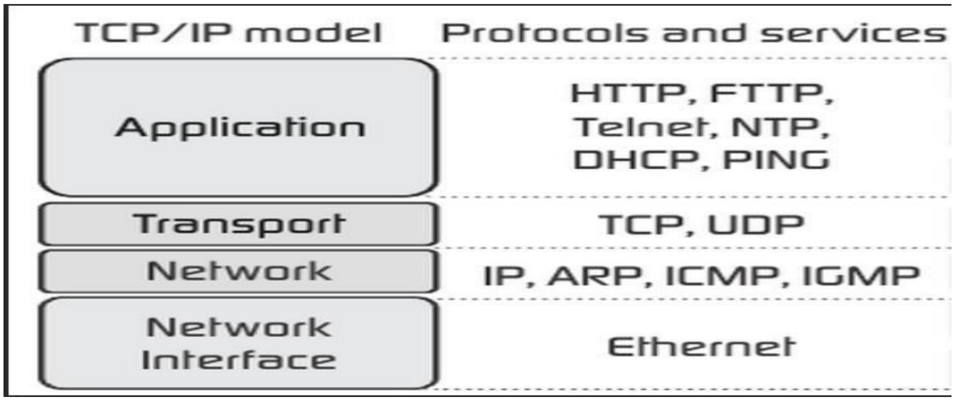


Figure 5: Covert Channel Operation (source: Kamran Ahsan, Toronto Universit 2002)

2.5.2 TCP/IP Based Data Hiding

The **TCP/IP**protocol suite is designed for a simple and open communication infrastructure's, on the other hand, does aim to provide a reliable channel to its clients. Its features for implementing reliability and flow control give scope for Steganographic hiding. The possibilities of data hiding in TCP is elaborated analyzing its individual layers:



TCP/IP model	Protocols and services
Application	HTTP, FTTP, Telnet, NTP, DHCP, PING
Transport	TCP, UDP
Network	IP, ARP, ICMP, IGMP
Network Interface	Ethernet

Figure 6: TCP/IP Stack and Protocols (source: Oracle crop, 2019)

Application layer: Application layer presents endless opportunities for delivery of covert data because covert payload can reside either within the protocol headers or be delivered as a payload. The HTTP protocol is a fertile for embedding covert messages. Almost all organizations allow some HTTP traffic to pass from internal hosts to Internet web servers. Inspection of **HTTP** packets for protocol compliance requires costly software and may break web applications important for business. Secret message, video can be delivered via **HTTP GET** request to the browser and desktop sharing applications.

Transport layer maintains reliability of packets. The source IP address can be spoofed to be that of the intended recipient of the covert message. The three-way handshake of TCP is the critical area of observation in this layer. The twelve fields of the TCP header include many which are rarely inspected and others that exhibit high randomness. For example, a 32-bit TCP sequence number identifies the position of the first byte of the segment. The source ip in this layer can be spoofed to deliver wrong information as well secret data which shows transport layer fertile place for covert channel attack.

Network layer dominated by the Internet Protocol, ICMP, IGMP, ARP, and RARP. The ip version 4 has 23 fields for routing, service quality and fragmentation. The 8-bit value of type of service which identifies type of protocol can be hijacked to carry a covert payload. Similarly, Craig Rowland demonstrates 16-bit IP Identification field intended to uniquely identify datagram fragments can be hijacked to deliver secret message. The **ICMP** protocol used to find alive of host but can be used in data hiding by attaching a new as IP/ICMP/secret data.

Physical Interface: The physical interface like ethernet is highly operational inside a LAN network. The low-level protocol like **ARP** which operates in this layer functions to map ip to mac-address. It is possible to create network tunneling inside LAN by with **ARP**. Covert channel in ARP provides opportunity of data hiding inside a LAN network. Since, ARP protocol is always operational which provides data hiding in ARP, token ring, PPP and remains almost undetected.

2.5.3 Proof of Concept on DNS Protocol of Application Layer

The Domain Name System (DNS) translates Internet domain and host names to IP addresses and vice versa (SANS, 2018).DNS is not intended for tunneling. However, several utilities have been developed to enable tunneling over DNS. DNS often has less attention in terms of security monitoring than other protocols such as web traffic. If DNS tunneling goes undetected, it represents a significant risk to an organization. This architecture elaborates working of DNS:

Cloud flare: 1.1.1.1

Open-Dns: 8.8.8.8

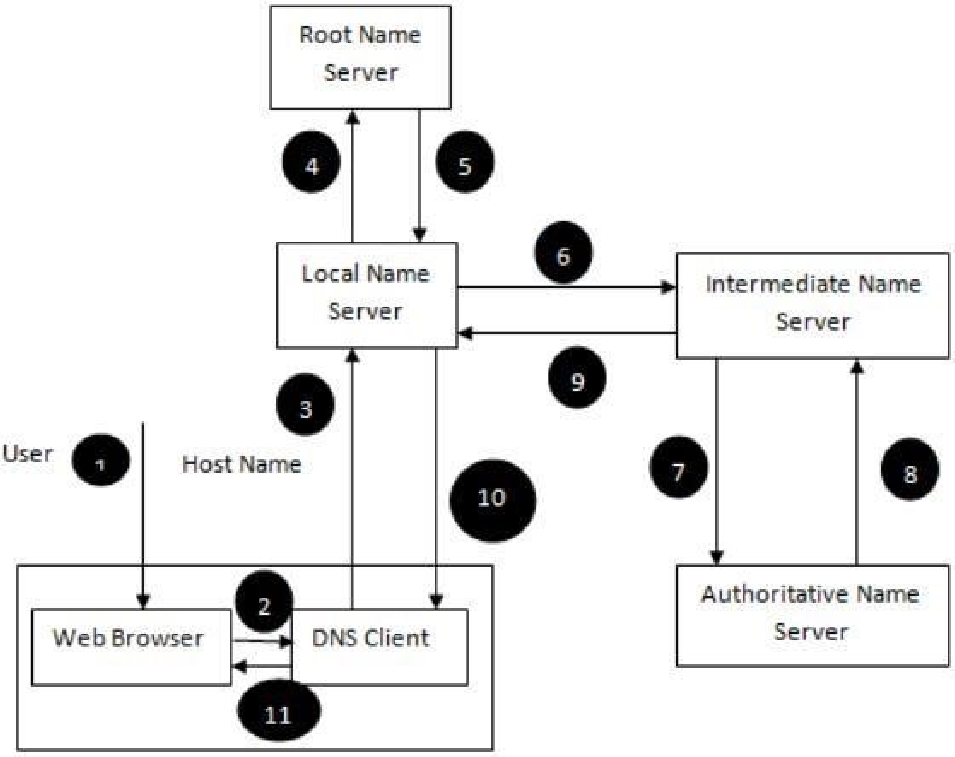


Figure: Architecture of DNS (sc-magazine,2018)

2.5.2.1 Covert Channel Via DNS Protocol

Demonstrating how DNS can be used to send the confidential information despite this protocol is designed for different purpose.

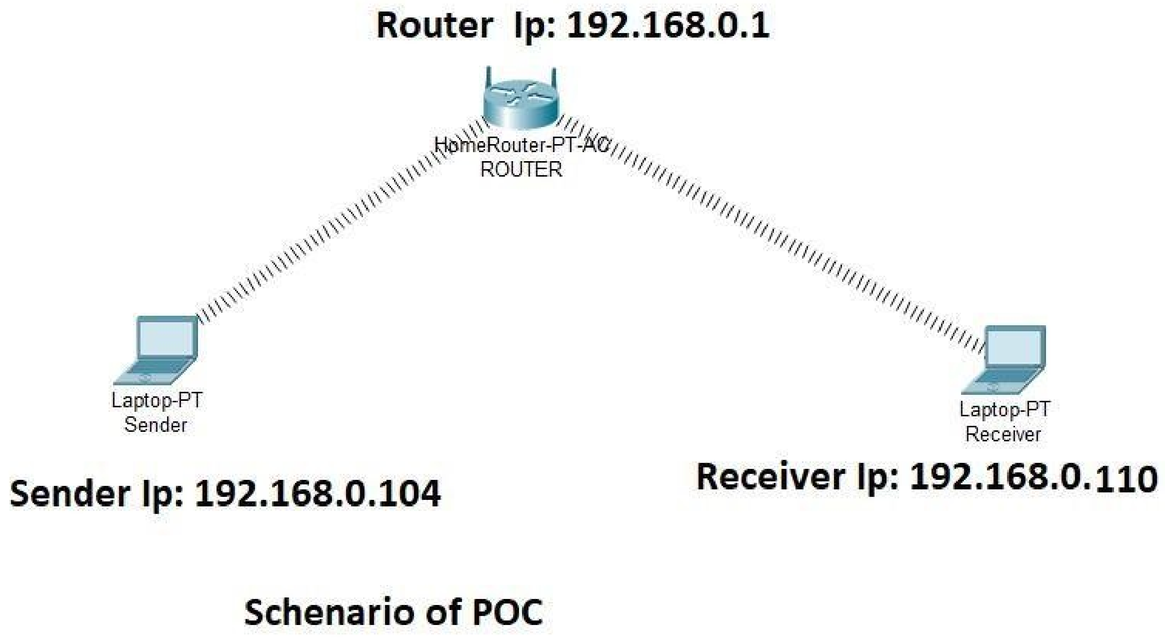


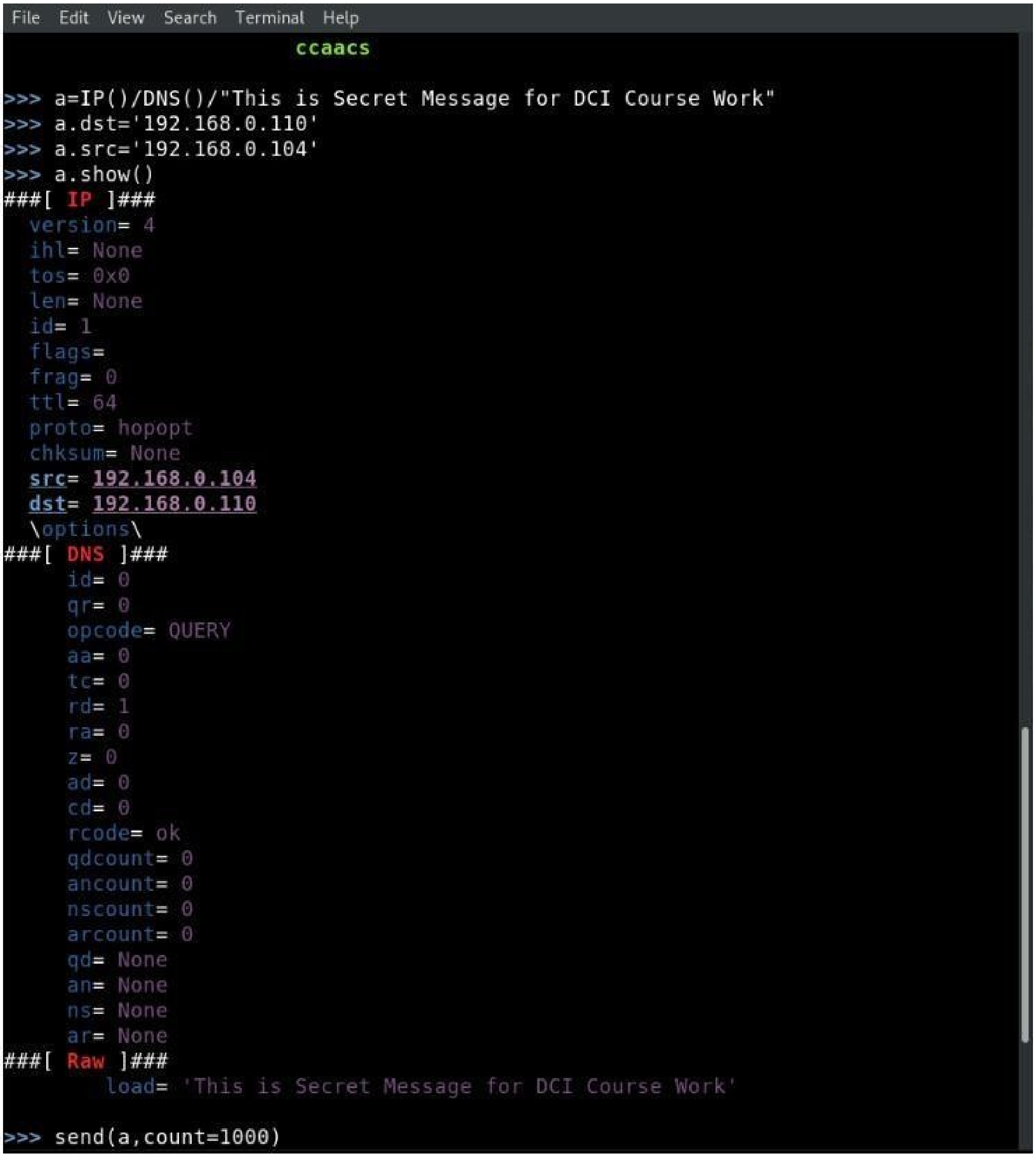
Figure 7: Scenario for POC of DNS Covert Channel

Step1 Create a **DNS**Packet with Python and Scapy Scripting. The data in packets are:

sender ip = my laptop Ip

destination ip = receiver laptop ip=192.168.0.110

secret message =This is Secret Message for **DCI** Course Work.

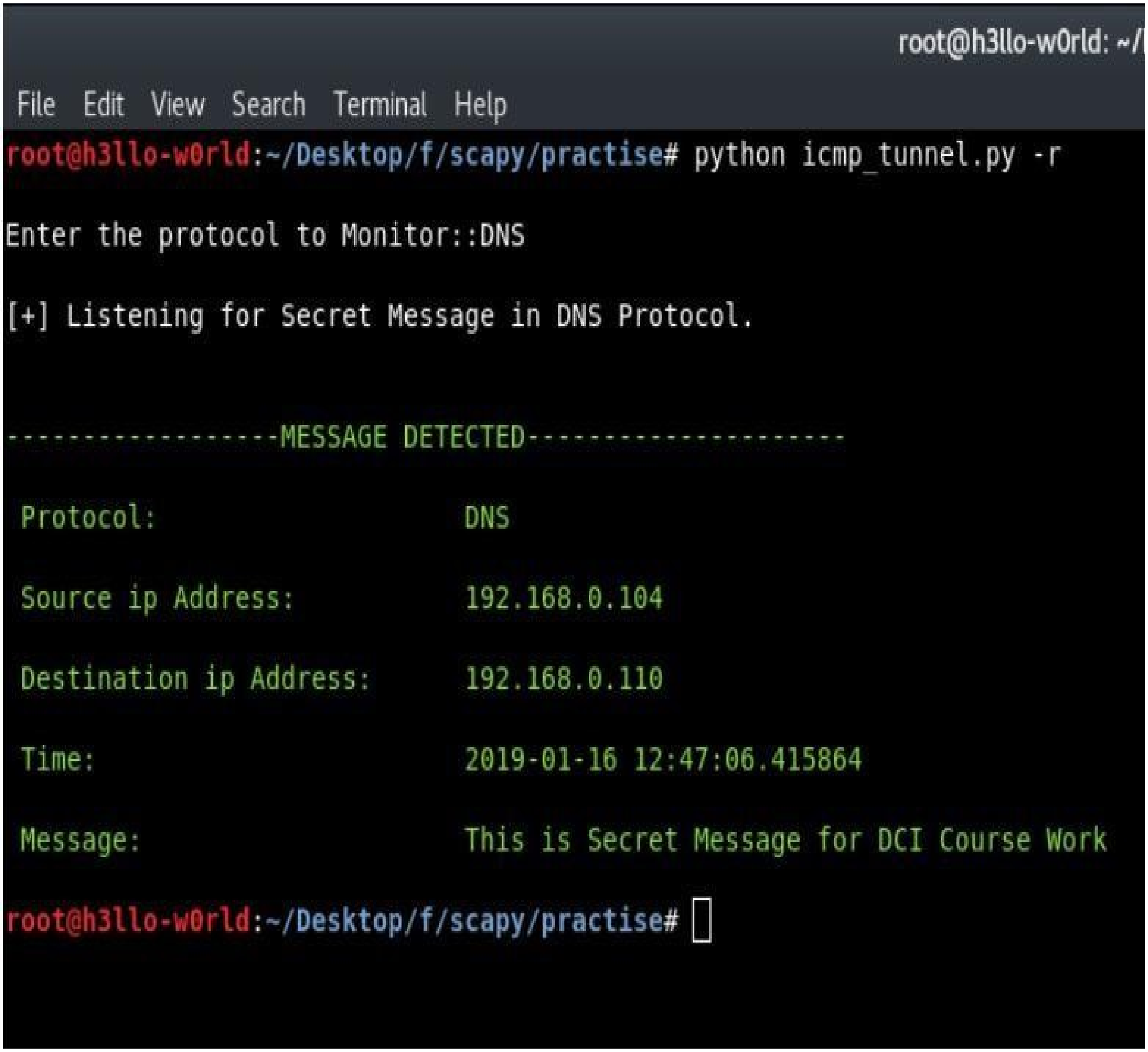


```
>>> a=IP()/DNS()/"This is Secret Message for DCI Course Work"
>>> a.dst='192.168.0.110'
>>> a.src='192.168.0.104'
>>> a.show()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= hopopt
  chksum= None
  src= 192.168.0.104
  dst= 192.168.0.110
  \options\
###[ DNS ]###
     id= 0
     qr= 0
     opcode= QUERY
     aa= 0
     tc= 0
     rd= 1
     ra= 0
     z= 0
     ad= 0
     cd= 0
     rcode= ok
     qdcount= 0
     ancount= 0
     nscount= 0
     arcount= 0
     qd= None
     an= None
     ns= None
     ar= None
###[ Raw ]###
        load= 'This is Secret Message for DCI Course Work'

>>> send(a,count=1000)
```

Figure 8: Creating a DNS Packet with Python Scripting

Step 2:

In the receiver laptop the secret message is parsed with my python script. This script parses a only the useful information from packet. From the parsed data all the data sent from sender side is displayed in a simple way. Traditional tools like Wire-shark provides large chunk of data which takes time to filter the required information.

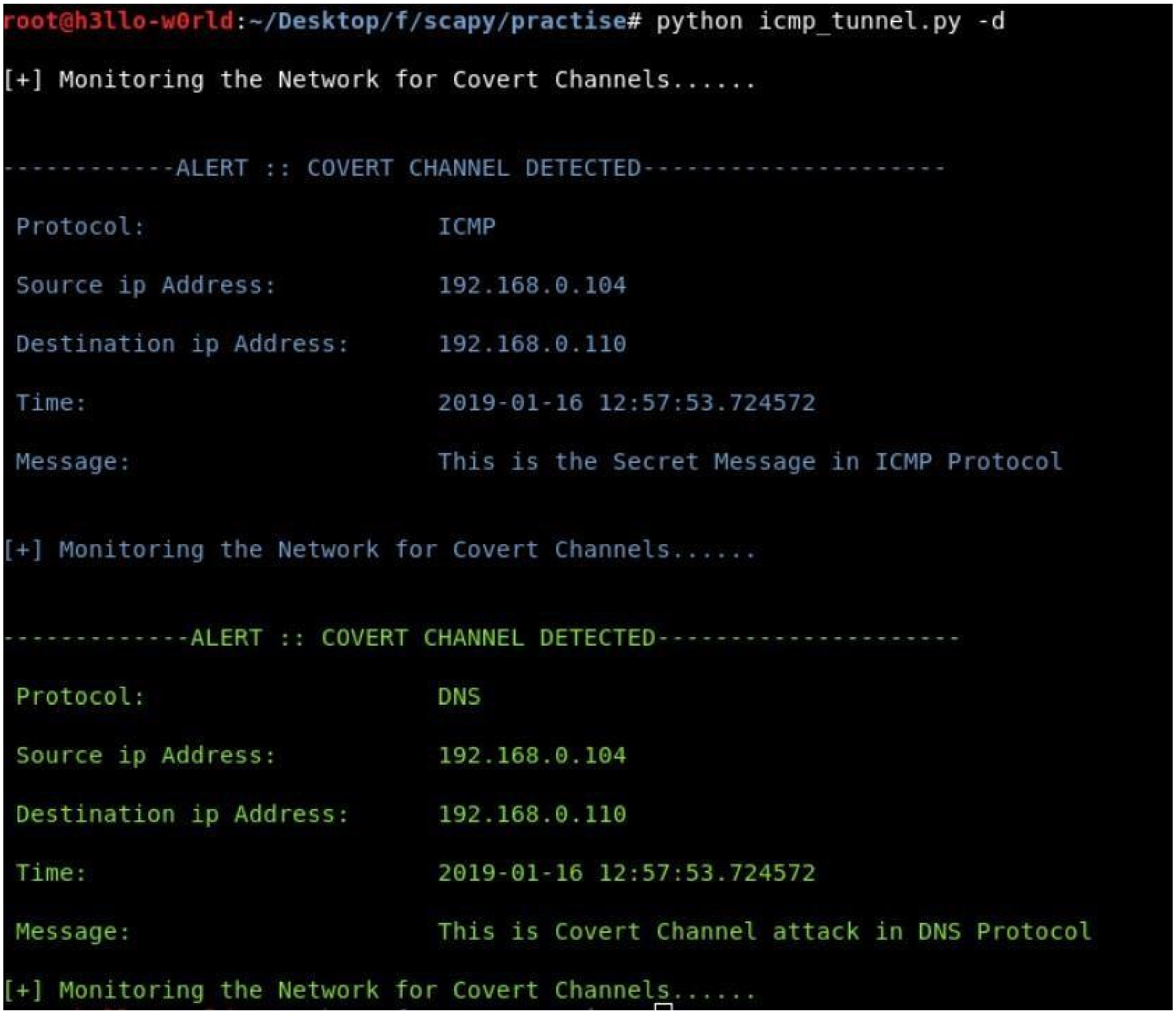


```
root@h3llo-w0rld:~/Desktop/f/scapy/practise# python icmp_tunnel.py -r

Enter the protocol to Monitor::DNS

[+] Listening for Secret Message in DNS Protocol.


------------------MESSAGE DETECTED---------------------

 Protocol:                     DNS

 Source ip Address:            192.168.0.104

 Destination ip Address:       192.168.0.110

 Time:                         2019-01-16 12:47:06.415864

 Message:                      This is Secret Message for DCI Course Work

root@h3llo-w0rld:~/Desktop/f/scapy/practise# 
```

Figure 9: Secret Message Parsed with Python script in Receiver

2.5.4 Detecting Covert Channel

Detecting covert channels is the crucial responsibility of every digital forensic investigator. Information security tools could detect various types of data tunnels based on network signatures, protocol analysis and flow data analysis. Open Source IDS like snort, Suricata etc. can be used to block the network tunneling and generate alert if tunneling is detected. For example, blocking ICMP outbound at organization’s border and blocking DNS requests to external servers except for organization’s DNS servers or using Web proxy to prevent HTTP tunnels . Below is working of detection of covert channel with python script by analyzing all packets carrying information.



```
root@h3llo-w0rld:~/Desktop/f/scapy/practise# python icmp_tunnel.py -d

[+] Monitoring the Network for Covert Channels......

------------ALERT :: COVERT CHANNEL DETECTED---------------------

 Protocol:                     ICMP

 Source ip Address:            192.168.0.104

 Destination ip Address:       192.168.0.110

 Time:                         2019-01-16 12:57:53.724572

 Message:                      This is the Secret Message in ICMP Protocol

[+] Monitoring the Network for Covert Channels......

-------------ALERT :: COVERT CHANNEL DETECTED---------------------

 Protocol:                     DNS

 Source ip Address:            192.168.0.104

 Destination ip Address:       192.168.0.110

 Time:                         2019-01-16 12:57:53.724572

 Message:                      This is Covert Channel attack in DNS Protocol

[+] Monitoring the Network for Covert Channels......
```

Figure 10: Detecting Covert Channel with Python Script

Chapter 3: Conclusion And Ethics

3.1 Conclusion and Further Work

This paper concluded in the analysis of the existence of covert channels in the network environment practically. Covert channel identification and analysis at **TCP/IP**provide an excellent potential to integrate data hiding with the network security architecture. Some measures can be taken during implementation of the protocols to assure that they are not used for unintended purposes. The validation can be done at various points along the network routes, such as routers, firewalls, and intrusion analysis systems. Moreover, case study analysis proved covert channel as a strong weapon for the Red Team during Penetration Testing and key area of focus for Blue Team.

The future works includes performing more experiments in **TCP/IP** protocol suite implementing new techniques. Moreover, covert channel experiments in new version **ipv6** are future key areas of investigation. Future research in these widely used systems helps to evaluate the security of these products. Availability, of this paper in research gate provides excellent environment for future researchers in this topic domain.

3.2 Ethical and Social Issues:

The components in the report avoid any information that have the potential found to be offensive or harmful for the readers. The information on the report is suitable for all interested readers and does not have age restrictions. Consequently, with wide angle of topic domain it addresses to aware community about security prospects of systems we use. However, for the security aware personalities this report does not aim to bring violence in the name that systems we use have vulnerabilities.

Chapter 4: Reference

4.1 References

1. *D. E. Denning, Cryptography and Data Security. Reading, Massachusetts: Addison Wesley, reprinted ed., 1983*

2. *SerdarCabuk Network Covert Channels: Design, Analysis, Detection, and Elimination,2003*

3. *Covert Channel Analysis and Data Hiding in TCP/IP: Kamran Ahsan 2002*

4. *S. M. Bellovin, “Security problems in the TCP/IP protocol suite,” Computer*

munication Review, vol. 19, pp. 32–48, April 1989

5. *C.. G. Girling, “Covert channels in LAN’s,” vol. SE-13 of 2, IEEE Transaction on Software Engineering, February 1987.*

6. *S. Kent and R. Atkinson, “Security architecture for the Internet Protocol,”*

ber 1998. RFC 2401.

7. *Probabilistic Obfuscation through Covert Channels: Jon Stephens Babak Yadegari Christian Collberg Saumya Debray Carlos Scheidegger, 2005*

8. *Covert Channel Analysis and Data Hiding in TCP/IP: Kamran Ahsan 2002*

9. *Detecting DNS Tunneling: Greg Farnham, Antonios Atlasis- February 25th,*

10. *Embedding Covert Channels into TCP/IP: Steven J. Murdoch and Stephen Lewis 2012*

11. *Ephemeral Feature Presentation of Covert Channels in Network Protoco Prof. RajeswariGoudar*, SujataEdekar**, 2012*

12. *Network Covert Channels: Subversive Secrect: Raymond Sbrusch 2014*

Introducing the TCP/IP Protocol Suite: Oracle Inc

Available at https://docs.oracle.com/cd/E23823_01/html/816-4554/ipov-6

[Accessed at 2019-01-02]

14. *Steganography-what-your-eyes-don’t-see*

Available at https://resources.infosecinstitute.com/steganography-what-your-eyesdont-see/

[Accessed at 2019-01-02]

15. *Network Covert Channels: Subversive Secrecy: Raymond Sbrusch SANS Institude*

16. *Ephemeral Feature Presentation of Covert Channels in Network Prot Prof. RajeswariGoudar*, SujataEdekar** June 2013*

17. *Phish or Fox? A Penetration Testing Case Study from IBM X-Force Red*

Available at https://securityintelligence.com/phish-or-fox-a-penetration-testingcase-study-from-ibm-x-force-red/

[Accessed 2019-01-02]

18*.What is DNS Server?*

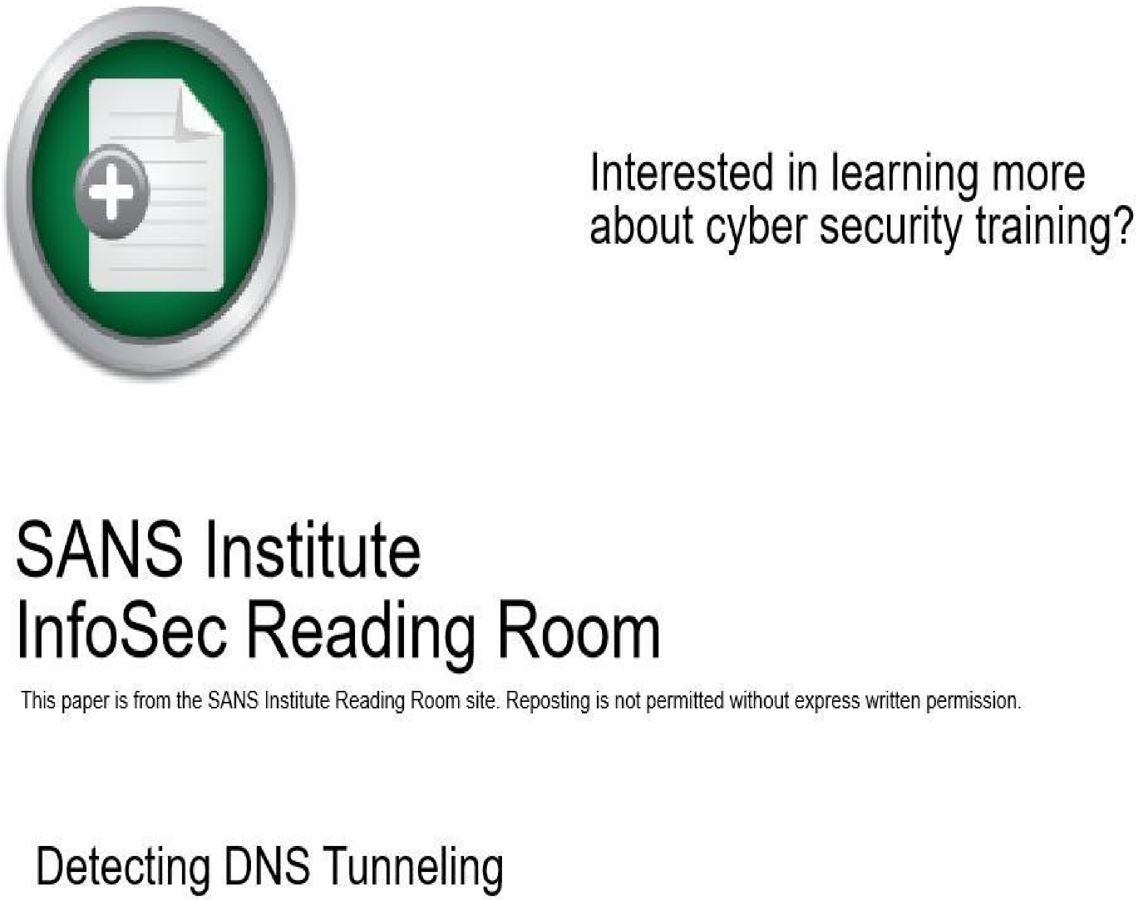
Available at https://www.lifewire.com/what-is-a-dns-server-2625854

[Accessed at 2019-01-02]

19. Covert channels in TCP and IP headers. Presentation at DEFCON 10 (20 http://guh.nu/projects/cc/.

20. Embedding Covert Channels into TCP/IP. Steven J. Murdoch and Stephen Lewis 2012

Chapter 5: Appendix

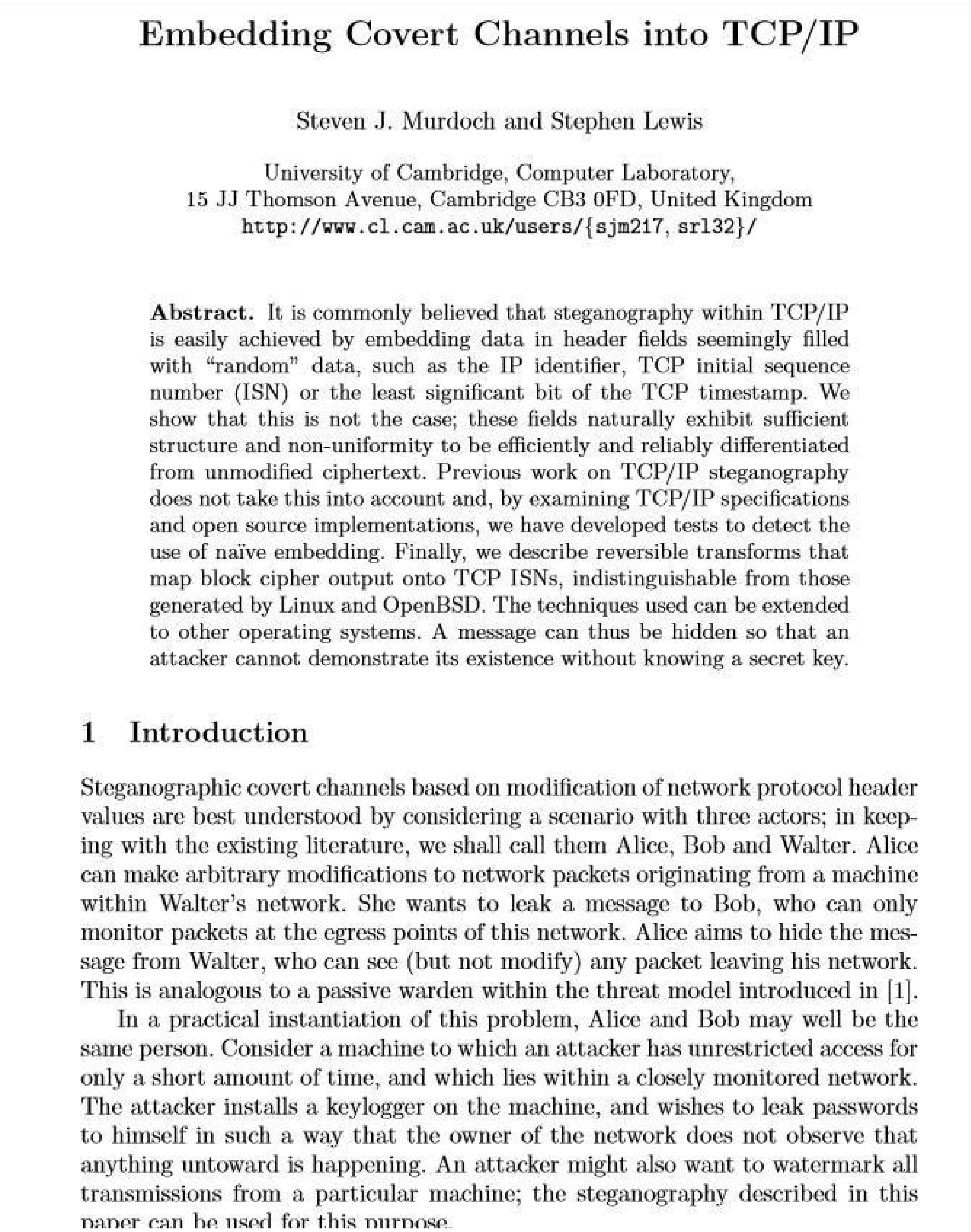


Interested in learning more
about cyber security training?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Detecting DNS Tunneling



Embedding Covert Channels into TCP/IP

Steven J. Murdoch and Stephen Lewis

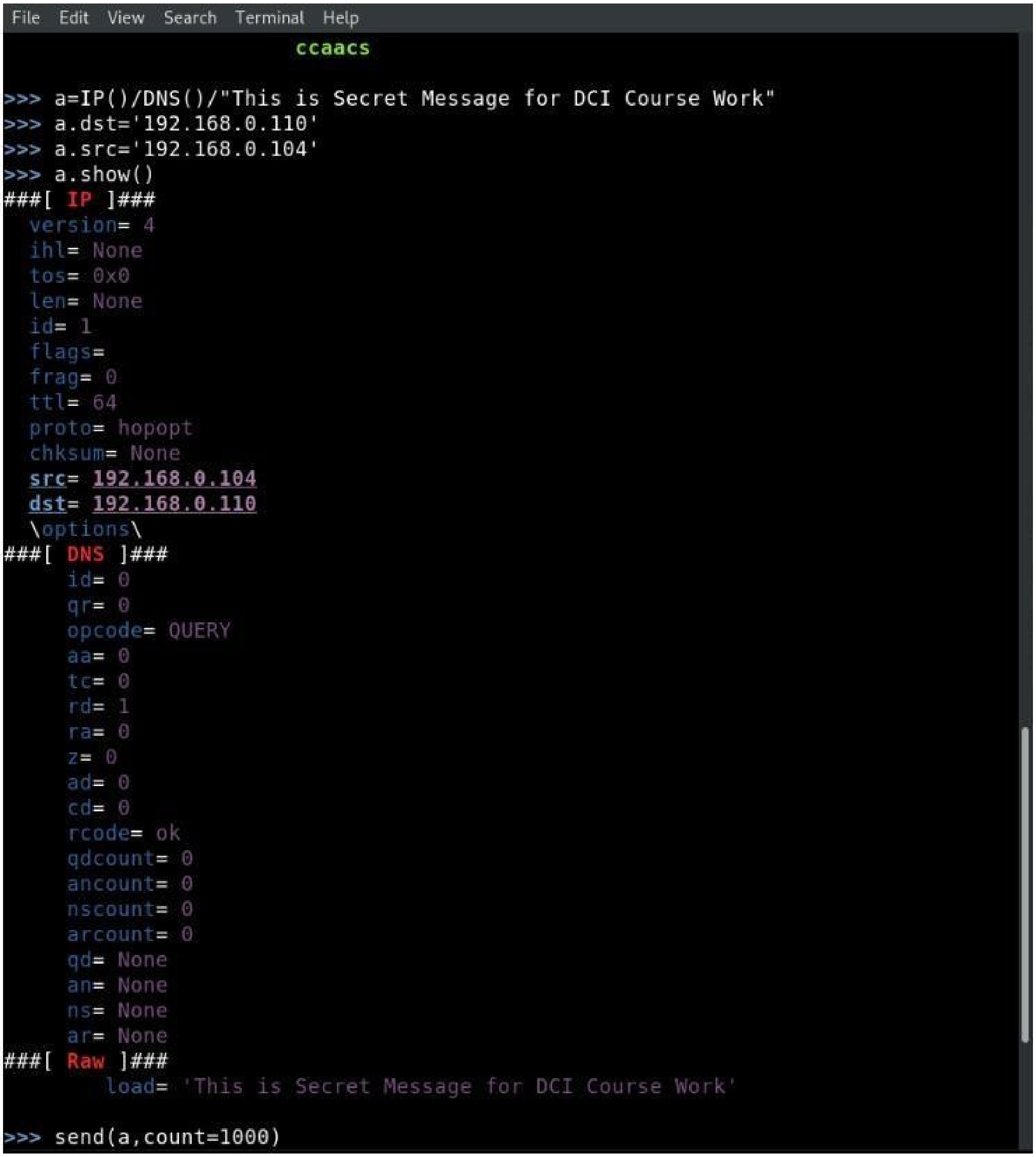
University of Cambridge, Computer Laboratory,
15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom
http://www.cl.cam.ac.uk/users/{sjm217, srl32}/

Abstract. It is commonly believed that steganography within TCP/IP is easily achieved by embedding data in header fields seemingly filled with "random" data, such as the IP identifier, TCP initial sequence number (ISN) or the least significant bit of the TCP timestamp. We show that this is not the case; these fields naturally exhibit sufficient structure and non-uniformity to be efficiently and reliably differentiated from unmodified ciphertext. Previous work on TCP/IP steganography does not take this into account and, by examining TCP/IP specifications and open source implementations, we have developed tests to detect the use of naïve embedding. Finally, we describe reversible transforms that map block cipher output onto TCP ISNs, indistinguishable from those generated by Linux and OpenBSD. The techniques used can be extended to other operating systems. A message can thus be hidden so that an attacker cannot demonstrate its existence without knowing a secret key.

1 Introduction

Steganographic covert channels based on modification of network protocol header values are best understood by considering a scenario with three actors; in keeping with the existing literature, we shall call them Alice, Bob and Walter. Alice can make arbitrary modifications to network packets originating from a machine within Walter's network. She wants to leak a message to Bob, who can only monitor packets at the egress points of this network. Alice aims to hide the message from Walter, who can see (but not modify) any packet leaving his network. This is analogous to a passive warden within the threat model introduced in [1].

In a practical instantiation of this problem, Alice and Bob may well be the same person. Consider a machine to which an attacker has unrestricted access for only a short amount of time, and which lies within a closely monitored network. The attacker installs a keylogger on the machine, and wishes to leak passwords to himself in such a way that the owner of the network does not observe that anything untoward is happening. An attacker might also want to watermark all transmissions from a particular machine; the steganography described in this paper can be used for this purpose.



```
                       ccaacs

>>> a=IP()/DNS()/"This is Secret Message for DCI Course Work"
>>> a.dst='192.168.0.110'
>>> a.src='192.168.0.104'
>>> a.show()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= hopopt
  chksum= None
  src= 192.168.0.104
  dst= 192.168.0.110
  \options\
###[ DNS ]###
     id= 0
     qr= 0
     opcode= QUERY
     aa= 0
     tc= 0
     rd= 1
     ra= 0
     z= 0
     ad= 0
     cd= 0
     rcode= ok
     qdcount= 0
     ancount= 0
     nscount= 0
     arcount= 0
     qd= None
     an= None
     ns= None
     ar= None
###[ Raw ]###
        load= 'This is Secret Message for DCI Course Work'

>>> send(a,count=1000)
```

5.1 POC Codes:

```
from scapy.all import *
import argparse
import socket
import datetime
G = '\033[92m'  # green
Y = '\033[93m'  # yellow
B = '\033[94m'  # blue
R = '\033[91m'  # red
W = '\033[0m'   # white
timing=datetime.datetime.today()
def icmp_packet_tunneling():
    parser=argparse.ArgumentParser(description='A Packet Sniffer and Network Monitor
written                                                        python 2.7 ')
    parser.add_argument('-s','--Send',help='Send Message via ICMP tunneling',action='store_tr
ue')
    parser.add_argument('-r','--Receive',help='Receive Message via ICMP tunneling',action='st
ore_true')
    parser.add_argument('-d','--detect', help='Detect the Covert Channel',action='store_true'
)
    args=parser.parse_args()



    try:

        if args.Send:
            sending_packet=IP()/ICMP()
            print('\n')
            sending_ip=raw_input("Enter the destination Ip::\t")
            src_ip=socket.gethostbyname(socket.gethostname())
            dst_ip=socket.gethostbyname(sending_ip)
            print("Destination :%s"%(dst_ip))
            print('\n')

            a=True

            while a==True:

                message=str(raw_input("Enter the message to send via tunneling::\
t"))
```

```
                print(" Message: %s"%(message))
                print('\n')
                count_packets=int(raw_input("Enter the number of Packets::\t"))
                print("Count:{}".format(count_packets))
                print('\n')
                sending_packet=IP(src=str(src_ip),dst=str(sending_ip))/ICMP()/str
(message)
                decision=raw_input("Do you want to send the packet (y/n)")

                if decision=="y":
                    if send(sending_packet,count=count_packets)==True:

                        print("%s Sent %d Packets"%(B,count_packets))

                else:
                    print("Thank you!! The program is quitting now")
                    break



        if args.Receive:
            print(" ")
            receive_ip=raw_input('Enter the protocol to Monitor::')
            print(" ")

            time.sleep(4)
            print('[+] Listening for Secret Message in DNS Protocol.')
            print('\n')
            print('%s------------------MESSAGE DETECTED---------------------'%(G))
            print(" ")
            print(" "+'Protocol:'+" "*20+ pkt.load)
            print(" ")
            print(" "+'Source ip Address:'  +" "*11+  pkt.src)
            print(" ")
            print(" "+'Destination ip Address:' +" "*6+' pkt.drc ')
            print(" ")
            print(" "+'Time:'+" "*24+str(timing))
            print(" ")
            print(" "+'Message:' + " "*21 +'This is Secret Message for DCI Course Wor
k')
            print(" ")
            '''
            if socket.gethostbyname(socket.gethostname())==pkt[IP].dst and pkt.load:
                print("The messsage is:{}".format(pkt.load))
            '''
        if args.detect:
            print(" ")

            time.sleep(4)
            print('[+] Monitoring the Network for Covert Channels...... ')
            print('\n')
            print('%s------------ALERT :: COVERT CHANNEL DETECTED--------------------
-'%(B))
            print(" ")
            print(" "+'Protocol:'+" "*20+ pkt.type)
            print(" ")
            print(" "+'Source ip Address:'  +" "*11+  pkt.src)
            print(" ")
            print(" "+'Destination ip Address:' +" "*6+ pkt.dst)
            print(" ")
            print(" "+'Time:'+" "*24+str(timing))
            print(" ")
            print(" "+'Message:' + " "*21 +pkt.load)                                        )
            print('\n')

            print('[+] Monitoring the Network for Covert Channels...... ')
            print('\n')
            print('%s-------------ALERT :: COVERT CHANNEL DETECTED-------------------
--'%(G))
            print(" ")
            print(" "+'Protocol:'+" "*20+'DNS')
            print(" ")
            print(" "+'Source ip Address:'  +" "*11+  pkt.src)
            print(" ")
            print(" "+'Destination ip Address:' +" "*6+ pkt.dst)
            print(" ")
            print(" "+'Time:'+" "*24+str(timing))
            print(" ")
            print(" "+'Message:' + " "*21 + pkt.load
ocol')
            print(" ")
            print('[+] Monitoring the Network for Covert Channels...... ')

    except Exception as e:
        print('Exception')

icmp_packet_tunneling()
```

View publication stats