패킷 스트림 분석을 위한 분류기 구조 설계

[1*]유진호
[1] 백석대학교 ICT 학부, 교수, yoojh@bu.ac.kr*

A Structure Design of Classifier for Analyzing Network Packet Streams

[1]Jinho Yoo
[1] Division of ICT, Baekseok University, Professor, yoojh@bu.ac.kr*

요 약

본 연구는 네트워크 상에 존재하는 다량의 데이터 분석에 관련한 것이다. 인터넷 상에는 패킷형태로 수많은 데이터를 주고 받는다. 본 연구에서는 이들 데이터 비트열을 무작위로 추출하여 이를 분석하는 프로토콜 분류기를 설계한다. 이들 비트열들은 네트워크 상에서 네트워크 관련 정보와 페이로드로 구성되지만 이들은 무작위로 섞여 있다. 본 연구의 분류기는 네트워크 상의 무작위 정형 비트열을 추출하여 바이트단위로 데이터를 생성하고 빈도점검, 공통비트열 추출, 비연관 데이터 분류, 카운터 추출, 매직번호 탐지 등을 수행한다. 이를 바탕으로 주소 검출과 프로토콜 관련 데이터 추출, 특징분석을 통한 비트열 분석기를 설계할 것이다. 이러한 분석기를 통해 프로토콜을 예측하고 페이로드 내용분석에 대한 가능성도 제시한다.

Abstract

This research is to study related to Analysing the data on internet. There are many packet data transmitted and received on internet. This paper will design protocol classifier which extracts the bit streams from internet and analysises them randomly. These extracted bit streams include the network information and their payloads and are mixed randomly. The classifier in the research extracts bit streams from internet randomly and creates byte units from them, and then does checking frequency, extracting common bit sequences, classifying of unassociated data, extracting protocol count, detecting magic number. This paper will design the classifier which is able to detect addresses, extract protocol related information and analyse features with these bases. This paper proposes the possibilities of prediction their protocols and analysis of payload contents.

Keywords: *Boltzmann, Feature Extraction, Protocol Analysis, Classifier, Network Packet Stream*

I. 서론

인터넷에는 수많은 데이터들이 있고 이 네트워크 상에 존재하는 데이터들 중에는 유해한 데이터도 포함되어 있으며 불법적이거나 네트워크를 공격하는 데이터들도 있다. 최근에는 이러한 데이터들을 파악하여 공공에 유해한 접근을 하는 패킷검출 기술이 개발되고 있다. 네트워크 상의 데이터는 MAC, TCP, UDP, IP 등의 프로토콜 데이터와 페이로드로 분류가 되며 이들 페이로드는 응용프로토콜 규약이 포함되어 다시 프로토콜 데이터와 해당 응용의 페이로드로 나누어 진다. 해당 프로토콜 데이터를 미리 정의된 지식없이 검출하여 동종의

* Corresponding Author
Received: Dec. 16, 2019, Revised: Dec. 31, 2019, Accepted: Dec. 31, 2019

프로토콜임을 판별하고 페이로드의 형태를 학습하여 페이로드의 예측된 분류를 수행하도록 한다. 페이로드의 경우에도 프로토콜에 따라 의미를 알 수 없는 바이너리로 구성이 되는 반면, 웹데이터의 경우 텍스트 기반 데이터를 포함하게 되며 본 연구에서는 이러한 형태의 데이터를 인식하고 프로토콜 을 예측하여 분류하는 분류기 모델을 제안한다.

II. 관련연구

네트워크 상의 데이터에 대한 다수의 연구가 있으며 여러 가지 접근방법이 있다. 네트워크 상의 패킷에 대한 의도파악 관련하여 연구는 침입탐지, 바이러스 예방, 네트워크 공격 등을 미리 파악하는데 사용한다. 또한 이미 알려진 프로토콜의 페이로드를 추출하여 처리하는데 사용된다. 이러한 연구는 리버스 엔지니어링 분야에서 주로 다루어져 왔다. 특정 내용을 담는 프로토콜 패킷을 반대로 추적하여 원본 데이터의 정보를 얻고자 하여 시작되었다. 이러한 기술은 두가지로 분류되며 분석적 방법과 데이터 획득방법이다. 분석적 방법은 프로토콜 문법생성 기술과 프로토콜의 유한기계 생성기술이다.

분석적 방법의 프로토콜 문법생성기술은 네트워크 추적 분석과 동적분석 기법으로 분류되며 네트워크 추적 분석을 위한 프로그램 도구로써 PIP(2004 년, HTTP, ICMP), Discoverer(2007 년, HTTP, RPC, CIFS/SMB)[1], Biprominer(2011 년, Xunlei, QQLive, SopCast)[2], ProDecoder(2012 년, SMB, SMTP)[3], AutoReEngine(2013 년, HTTP, POP3, SMTP, FTP, DNS, NetBIOS)[4], ReverX(2011 년, FTP) 등이 있다. 또한 동적분석 방법도구로는 Polyglot(2007 년, HTTP, DNS, IRC, SAMBA, ICQ)[5], AutoFormat(2008 년, HTTP, SIP, DHCP, RIP, OSPF, SMB)[6], Tupni(2008 년, WMF, MBP, JPG, PNG, TIF, DNS, RPC, TFTP, HTTP, FTP)[7], Wonracek(2008 년, HTTP, IRC, SMTP, DNS, SMB), Dispatcher(2009 년, MegaD, HTTP, DNS, FTP, ICQ)[8], ReFormat(2009 년, HTTPS, IRC, MIME)[9], Hierarchical Approach(2010 년, bc, wu-ftpd, gcc), Backward Slicing(2013 년, Agobot3, SDBot) 등으로 생성년도와 분석대상 프로토콜을 명시하고 있다. 유한기계 생성기술의 경우는 네트워크 추적분석 기법과 동적분석 기법, 대화형 분석이 있으며 네트워크 추적분석 기법 도구는 ScriptGen(2005 년), RolePlayer(2006 년), Pext(2007 년, FTP), Trifilo 작업(2009 년, ARP, DHCP, TCP, KAD), PreverX(2011 년, FTP), Veritas(2011 년, SMTP, PPLIVE, XUNLEI) 등이 있다. 동적분석 도구로는 Xiao 작업(2009 년, HTTP, FTP, SMTP), Prospex(2009 년, SMB, SMTP, SIP, Agobot) 등이 있다. 대화형 분석기법은 Cho 의 연구(2010 년, SMTP), Zhang 의 연구(2012 년, SNMP, HTTP, ISAKMP), LaRoche(2013 년, FTP, DHCP) 등이 있다[10]. 또한 알려지지 않은 비트열처리의 F. Zhang, J. Zhang, H. Zhou 의 연구가 있다 [11].

III. 분류기 시스템 설계 및 구현

3.1 전체 시스템 구성

본 연구는 네트워크 상의 무작위로 패킷을 분석하여 프로토콜 특성을 추출하고 그 추출된 특징을 기반으로 특정 프로토콜을 분류한다. 그림 1 은 전체 시스템의 구성도를 나타낸다.

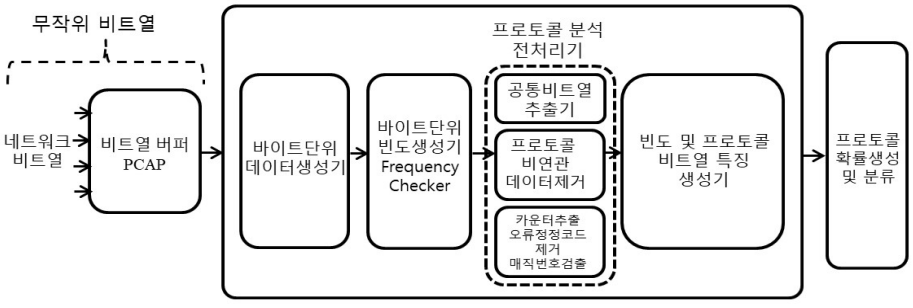

Figure 1 Overall System Block Configuration

시스템은 네트워크 상의 패킷 비트열을 받아서 pcap 형식으로 입력된다. 입력된 패킷열은 데이터처리의 간편성을 위해 바이트 단위로 생성한다. 바이트 단위 빈도생성기는 분석 대상이

되는 데이터를 바이트 단위로 빈도를 측정하기 위한 것이다. 네트워크 패킷에서 바이트 값의 빈도는 프로토콜 정보인지를 판단하는데 중요한 지표가 된다[11]. 프로토콜 분석 전처리기는 프로토콜에 분류를 위한 작업을 하기 전에 입력된 프로토콜이 무엇인지 판단하기 위해 패킷 스트림에서 분류에 방해가 되는 데이터를 제거하거나 프로토콜의 핵심적인 정보를 추출하여 파악할 때 전처리기에서 수행한다. 즉 전처리기는 빈도 등의 프로토콜 특징을 바이트 단위로 검사하여 특징을 생성하게 된다. 이렇게 추출된 바이트열을 특징으로 만들어 실제 패킷 스트림을 대표하여 분류기의 입력으로 사용된다. 즉, 네트워크로부터 얻어진 무작위 프로토콜 바이트열에서 방해되는 부분과 분류에 도움이 되는 부분을 구별하여 데이터를 구성한 후 특징형태의 바이트열로 생성하여 분류기의 입력으로 사용하게 된다.

3.2 프로토콜 분석 전처리기

프로토콜 전처리기는 입력된 데이터 스트림에서 주소 및 프로토콜 연관데이터를 분석한 결과를 바이트 단위의 데이터로 구성하여 특징을 생성한다.

3.2.1 프로토콜 연관 데이터 분석

패킷주소는 MAC 주소와 IP 주소 그리고 해당 프로토콜이 사용하는 포트번호가 있다. 프로토콜을 판명하는데 있어서 MAC 주소와 IP 주소는 프로토콜을 결정하는데 필요하지 않은 데이터이다. 주소는 어떤 네트워크 엔터티 간에 통신이 이루어지는가를 파악하는데 사용되고 엔터티 간의 프로토콜 내용에는 연관이 없다. 그러나 주소는 네트워크 엔터티 간의 네트워크 세션을 유지하고 있을 때의 데이터를 주고 받는 상황의 순서파악에 도움이 되어 주고 받는 순서가 있는 정보로써 데이터의 의미를 추측할 수 있다. 특정 네트워크 엔터티 간에 데이터 수집에 주소가 사용될 것이다. 그러나 포트번호와 같은 경우는 프로토콜의 특징을 나타내는 매우 중요한 요소가 된다. 프로토콜이 사용하는 주된 포트가 정해져 있기 때문이다. 이러한 네트워크 특성들을 전처리기를 통해서 추출하고 이를 분류기의 특징으로 활용한다.

3.2.2 주소검출 및 프로토콜 연관데이터 검출

데이터 검출을 위해서 모든 패킷은 바이트 배열로 정의되며 일정량의 패킷을 대상으로 IP 주소 등의 네트워크 일정세션에 대한 검출을 수행한다. 검출된 주소는 패킷을 추적하는데 주로 사용되며 주소구간은 프로토콜의 특징요소로 사용되지 않는다. 그림 2 는 주소검출과정을 설명하고 있다.

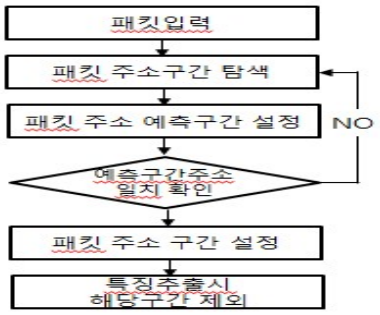

Figure 2 Finding Address Area

각 바이트들은 위치와 카운팅 정보 등을 저장하여 어느 위치에 얼마나 자주 나타나는 가를 주된 정보로 활용한다. 이와 같이 처리된 데이터는 프로토콜 연관데이터의 범주에서 세션 별로 수행되게 된다.

$$
\begin{aligned}
&Packet\ Set: Packets = Pck_1, Pck_2, Pck_3, ..., Pck_n \\
&\text{for } Pck_i \in Packets: \\
&\quad Byte_i, Pos_i = Pck_i.getbyte() \\
&\quad \text{for } Pck_{i+1} \in Packets: \\
&\qquad Byte_{i+1}, Pos_{i+1} = Pck_{i+1}.getbyte() \\
&\qquad \text{if } Byte_i.value == Byte_{i+1}.value \text{ and } Pos_i == Pos_{i+1}: \\
&\qquad\quad Byte_i.fcount \mathrel{+}= 1 \\
&\qquad\quad Byte_{i+1}.fcount \mathrel{+}= 1
\end{aligned}
$$

Figure 3 Checking Port Number

$$
\begin{aligned}
&Packet\ Set: Packets = Pck_1, Pck_2, Pck_3, ..., Pck_n \\
&\text{for } Pck_i \in Packets: \\
&\quad Byte_i, Pos_i = Pck_i.getbyte() \\
&\quad \text{for } Pck_{i+1} \in Packets: \\
&\qquad Byte_{i+1}, Pos_{i+1} = Pck_{i+1}.getbyte() \\
&\qquad \text{if } Pos_i == Pos_{i+1} \text{ and } Is\_freq(Byte_i.value, Byte_{i+1}.value \\
&\qquad\quad declare\ candidate\_frequent
\end{aligned}
$$

Figure 4 Checking Count

$$
\begin{aligned}
&Packet\ Set: Packets = Pck_1, Pck_2, Pck_3, ..., Pc \\
&\text{for } Pck_i \in Packets: \\
&\quad Byte_i, Pos_i = Pck_i.getbyte() \\
&\quad \text{if } Byte_i == Packi.length(): \\
&\qquad declare\ candidate\_length
\end{aligned}
$$

Figure 5 Extracting Length Byte

전처리를 수행하기 위해서는 프로토콜 관련바이트를 페이로드와 분리한다. 그림 3 은 포트번호를 검사하고 바이트빈도를 계산하며, 그림 4 는 위치와 값을 통해서 빈도수를 계산한다. 그림 5 는 프로토콜 정보를 조사하여 길이 바이트를 추출하는 것으로 어떤 프로토콜이든 한계를 나타내기 위해 길이라는 특성을 사용한다. 이는 전체 패킷에서 페이로드를 제거하는데 사용된다. 위와 같이 특징추출을 위해 각 바이트 빈도와 통계적 방법을 사용하고 이를 그림 6 과 같이 알고리즘화 하였다.

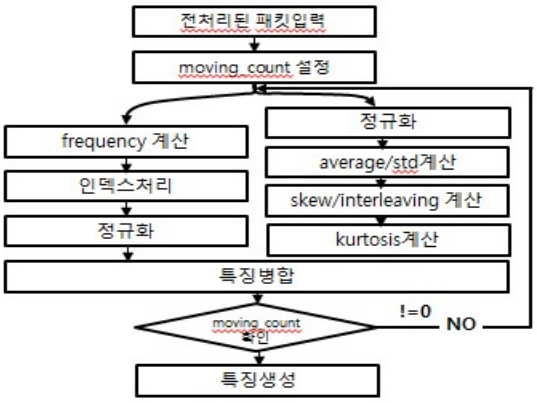

Figure 6 Feature Extraction Sequence

3.3 분류기 설계

원본 네트워크 패킷 데이터가 전처리기 단계로 입력이 되면 전처리기는 패킷 데이터를 분석하여 특징을 추출한다. 이렇게 추출된 특징 데이터는 입력되는 패킷을 대상으로 바이트 단위로 만들어 지며 특징을 입력으로 하여 분류기는 분류를 수행한다. 분류기를 테스트하기 위해 기존에 수집된 패킷데이터를 7:3 비율로 나누어 트레이닝 데이터와 테스트 데이터로 구성하며 각각의 목표값인 레이블 값을 정한다. 트레이닝 데이터를 사용하여 모델을 만드는데 사용하고 테스트 데이터를 사용하여 만들어진 모델을 평가한다. 분류기 모델은 감독 딥빌리프네트워크 모델을 사용하였다. 딥빌리프네트워크는 제한볼쯔만머신이라는 신경망 모델을 단위 블록의 레이어를 만들어 사용하는 딥러닝 모델이다. 제한볼쯔만머신은 주어진 입력과 똑같은 출력을 생성하도록 하는 오토인코딩을 수행하는 모델이며 신경망이 계산을 수행할 때 결정적 계산을 수행하는 것이 아닌 확률적으로 출력값을 계산한다.

제한 볼쯔만 머신을 사용하는 이유는 패턴 생성시 발생하는 문제를 해결하기 위함이다. 제한 볼쯔만 머신의 패턴 생성 능력은 입력한 데이터인 트레이닝 데이터에 있는 데이터의 패턴 뿐만

아니라 입력하지 않았던 데이터도 생성해낼 수 있는 능력을 가지고 있다. 단순히 매치되는 정보검색의 정보를 단순히 인출하는 처리가 아니라 제한볼쯔만머신은 에너지를 이용하기 위해 정의된 확률분포를 학습하는 모델로 학습 데이터의 분포에 따라 생성 데이터의 분포도 영향을 받게 되며 직접 트레이닝 되지 않은 데이터를 생성할 수 있는 능력을 가지고 있다.

본 연구에서 사용하는 딥빌리프 모델은 제한볼쯔만머신을 빌딩 블록으로 하고 블록을 여러 층으로 구성한 딥러닝 구조이다. 딥빌리프 모델은 무감독 학습구조이지만 마지막 레이어인 은닉층을 입력으로 하고 또 다른 출력층을 추가하여 감독학습 구조를 만든다. 해당 구현에 맞게 적절한 매개변수를 선별하였고, 히든 레이어 구조는 [256, 128, 256], 제한볼쯔만머신의 학습률은 0.001, 학습률은 0.01, rbm 의 epoch 값은 5, 반복 백프로퍼게이션은 500, 빠른 수행을 위한 배치사이즈는 2000, activation 함수는 relu 로 설정하였고 드롭아웃은 0.2 로 설정하여 실험하였다.

IV. 실험

4.1 실험대상 프로토콜과 수집방법

실험대상 프로토콜은 ftp, http, smtp, mysql, tds, ldap, smb, ssh, arp, icmp 등의 10 개 프로토콜을 기준으로 실험하였다. 실험대상 프로토콜은 일반적으로 많이 사용하는 프로토콜로 실험시 프로토콜 식별의 성능을 위해 사용되며 분류기는 해당 프로토콜을 모르는 상태에서 분석을 수행하게 된다. 이들 프로토콜은 pcap 데이터 형식으로 입력되어 특징분석 및 생성 과정을 거치고 특정 파라미터와 함께 분류기 모델을 생성한다. 본 연구에서 사용되는 프로토콜 데이터는 www.netresec.com 으로 부터 수집하였다.

4.2 실험 데이터 분석

그림 7 은 특성데이터를 분석한 특징 그래프이며 프로토콜 별로 특성이 강하게 나타내는 부분을 시각화하여 표현하였고 특성을 계산하여 산출해낸 특성값들을 표현한 것으로 패킷 앞쪽의 값이 큰값으로 분포되어 있는 것을 관찰할 수 있다. 가로축은 패킷열을 나타내며 세로축은 특성 분포의 정도를 나타내는 크기값이다.

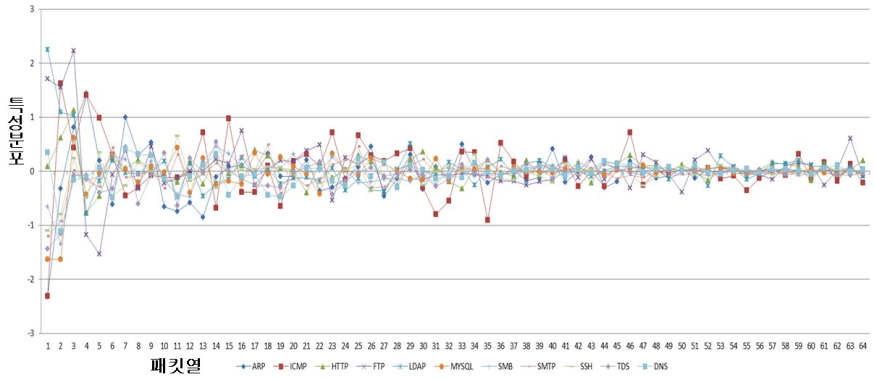

Figure 7 Feature Analysis Graph

4.3 실험 결과

표 1 에서 보는 바와 같이 총 8 개의 프로토콜을 실험하였고 훈련세트 정확도는 0.978, 테스트 세트의 정확도는 0.981 로 매우 양호한 결과를 산출하였으며, ldap 의 경우 91%, 0.02%(mysql 연관도), 0.08%(dns 연관도), http 의 경우 99%, 0.007(ldap 연관도), dns 의 경우 95%, 0.053%(mysql 연관도)와 같은 실험결과를 얻었다.

Table 1 Experiment Result

	arp	icmp	ldap	mysql	smtp	tds	http	dns
arp	100%	0	0	0	0	0	0	0

icmp	0	100%	0	0	0	0	0	0
ldap	0	0	91%	0.02%				0.08%
mysql	0	0	0	100%	0	0	0	0
smtp	0	0	0	0	100%	0	0	0
tds	0	0	0	0	0	100%	0	0
http	0	0	0.007	0	0	0	99%	0
dns	0	0	0	0.053%	0	0	0	95%

V. 결론

본 연구에서는 네트워크 상의 임의 패턴 데이터들을 대상으로 프로토콜을 식별하는 실험을 하였다. 네트워크의 특성을 가지는 공통 비트열을 찾기 위해 전처리를 수행하였고 전처리 단계에서 일정패턴이 아닌 부분들을 제거하였다. 이러한 일련의 전처리 작업을 통해 특성화될 수 있는 비트열을 추출하였다. 또한 추출된 비트열을 특성화 하여 제한 볼쯔만 머신을 단위블록으로 하는 딥빌리프네트워크 분류기 모델을 사용하여 프로토콜 분류기를 설계 및 구현하였고 실험과 같은 결과를 얻을 수 있었다. 본 연구를 통해 인터넷 상의 패킷 비트열에 대한 프로토콜 분석이 가능하고 추후에는 알려지지 않은 프로토콜의 구조분석과 페이로드 분석이 가능할 것이다.

VI. 감사의 글

이 논문은 2019 학년도 백석대학교 학술연구비 지원을 받아 작성되었음.

VII. 참고문헌

[1] Weidong Cui, Jayanthkumar Kannan, Helen J. Wang, " Discoverer: Automatic Protocol Reverse Engineering from Network Traces", USENIX Security Symposium, pp.199-212, 2007.
[2] Y. Wang, Y. Zhao, Z. Zhang, X. Li, J. Meng, L. Guo, "Biprominer: Automatic Mining of Binary Protocol Features", International Conference on Parallel and Distributed Computing Applications and Technologies Pdcat 2011, pp. 179-184, 2011.
[3] Y. Wang, X. Yun, M. Z. Shafiq et al., “A semantics aware approach to automated reverse engineering unknown protocols,” in Proceedings of the 20th IEEE International Conference on Network Protocols (ICNP '12), pp. 1–10, IEEE, Austin, Tex, USA, November 2012.
[4] J.-Z. Luo and S.-Z. Yu, “Position-based automatic reverse engineering of network protocols,” Journal of Network and Computer Applications, vol. 36, no. 3, pp. 1070–1077, 2013.
[5] J. Caballero, H. Yin, Z. Liang, and D. Song, “Polyglot: automatic extraction of protocol message format using dynamic binary analysis,” in Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07), pp. 317–329, ACM, November 2007.
[6] Z. Lin, X. Jiang, D. Xu, and X. Zhang, “Automatic protocol format reverse engineering through context-aware monitored execution,” in Proceedings of the 15th Symposium on Network and Distributed System Security (NDSS '08), February 2008.
[7] W. Cui, M. Peinado, K. Chen, H. J. Wang, and L. Irun-Briz, “Tupni: automatic reverse engineering of input formats,” in Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS '08), pp. 391–402, ACM, Alexandria, Va, USA, October 2008.
[8] J. Caballero, P. Poosankam, C. Kreibich, and D. Song, “Dispatcher: enabling active botnet infiltration using automatic protocol reverse-engineering,” in Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09), pp. 621–634, ACM, Chicago, Ill, USA, November 2009.
[9] Z. Wang, X. Jiang, W. Cui, X. Wang, and M. Grace, “ReFormat: automatic reverse engineering of encrypted messages,” in Computer Security—ESORICS 2009. ESORICS 2009, M. Backes and P.

Ning, Eds., vol. 5789 of Lecture Notes in Computer Science, pp. 200–215, Springer, Berlin, Germany, 2009.

[10] Baraka D. Sija, Young-Hoon Goo, Kyu-Seok Shim, Huru Hasanova, and Myung-Sup Kim, "A Survey of Automatic Protocol Reverse Engineering Approaches, Methods, and Tools on the Inputs and Outputs View," Security and Communication Networks, Volume 2018, Article ID 8370341, 17 pages, 2018.

[11] Fengli Zhang, Junjiao Zhang and Hongchuan Zhou, “Unknown Bit Stream Protocol Message Discovery with Zero Knowledge”, ICA3PP 2015 Workshops, LNCS 9532, pp.800-809, 2015.