

Zero-Day Attack Packet Highlighting System

Jang Hyeon Jeong, Jong Beom Kim*, Seong Gon Choi

Information & Communication Engineering, Chungbuk National University, Cheongju-si Chungcheongbuk-do, Korea

*Xabyss Inc, Seongnam-si, Gyeonggi-do, Korea

wkdgus4788@chungbuk.ac.kr, jbkim@xabyss.com, choisg@chungbuk.ac.kr

Abstract— This paper presents Zero-Day Attack Packet Highlighting System. Proposed system outputs zero-day attack packet information from flow extracted as result of regression inspection of packets stored in flow-based PCA. It also highlights raw data of the packet matched with rule. Also, we design communication protocols for sending and receiving data within proposed system. Purpose of the proposed system is to solve existing flow-based problems and provides users with raw data information of zero-day packets so that they can analyze raw data for the packets.

Keywords— NIDPS, PCA, Zero-Day Attack, DPI

I. INTRODUCTION

Many people use security systems such as NIDPS (Network Intrusion Detection Prevention System), firewall, anti-DDoS, etc for detecting and preventing from network attack. NIDPS has Intrusion Detection System (IDS) function and Intrusion Prevention System (IPS) function. IDS is system which detects incoming attack packets from external networks. IPS can analyze and detect attack packets as well as preventing from various attacks [1]-[5]. NIDPS is system capable of DPI(Deep Packet Inspection) that inspects not only header of packet but also payload. [6]. snort, suricata and bro are representative open NIDPS engines. NIDPS engines are being used for real network security [7]-[11].

However, all security system including NIDPS have security blind spots that cannot detect attack [12]-[14]. zero-day attack means that unknown attack packet is received and damage system by the packet. Zero-day attack targets undisclosed vulnerability that hackers can be used to adversely affect computer programs [15],[16]. Ways to recognize and respond to the attack should be created and applied to the system [14],[4]. But this process takes from hours to days [14],[17]. Therefore, it is impossible to respond to the damage caused by received zero-day attack packets until new countermeasure is updated [17].

In order to minimize damage caused by zero-day attack, It became necessary to store all received packets. PCA(Packet Capture Appliance) is store all received packets, because they has possibility of zero-day attack. All packets stored in the PCA are rechecked using the existing security system. After new rules update is complete. In [18], SENTRYWIEE Inc. uses PCA for checking if receiver has received unknown attack packets in the past. Their equipment stores all received packets as full-packet. When new rule is updated to detect attack packets with new pattern, it checks received packets in

the past. If attack packets with new pattern are detected in rechecked packets, PCA or administrator traces the packets and then solves problems by zero-day attacks.

But, PCA has problem that it should have enormous storage capacity. Because it should store all received packets for months or more. PCA with enormous storage capacity is very expensive. Worldwide IP traffic was expected to increase steadily, nearly tripling from 2017 to 4.8 zettabytes in 2022 [19]. As a result, it has become very difficult for existing PCA to store all received packets as full-packet for a long time. So, studies have been carried out to increase the efficiency of PCA. In [20]-[24], they studied to increase the efficiency of storage space of PCA. Their proposed systems and methods used techniques to store only a fraction of the received packets, not full-packet. As a result, not only did it have similar detection performance to existing PCA equipment, but it also greatly improves storage efficiency.

Although, there are problems that existing security equipment does not display raw data of detection attack packets, so it is difficult to use them for attack packet analysis. Also, equipment such as flow based PCA cannot detect accurate zero-day attack packets. To solve these problems, authors propose Zero-Day Attack Packet Highlighting System.

In chapter 2, we describe existing studies on previous security systems such as NIDPS and flow-based PCA. Chapter 3 is introduced about proposed system for zero-day attack packet highlighting. In chapter 4, authors describe the experiments and results of the proposed system. Finally, chapter 5 presents conclusions.

II. RELATED WORKS

In [25]-[28] and numerous studies, signature-based NIDPS is used. Signature-based NIDPS can drop and detect packets which are known for attack packet. It detects It defines pattern of attack packet called signature as rule. Packets matched with the rule are detected and processed in method(alert, drop, log, ect) defined in the rule. Figure 1 is shown the log of suricata, a kind of Signature-based NIDPS. Signature-based NIDPS outputs flow information (source IP address, destination IP address, source port number, destination port number, protocol) of the attack packet, but raw data of payload matched with rule is not output.

```
10/09/2019-01:01:25.943297 [**] [1:0:0] Warning protocol
HTTP [**] [Classification: (null)] [Priority: 3] {TCP}
203.255.89.46:80 -> 192.168.0.30:33772
```

Figure 1. Suricata Log

In [23], Authors only stored first n packets of session in PCA to solve the storage problems of existing PCA due to overload of pre-processing and storage of Big Data. This method shows an efficiency of about 99.319% storage space compared with the full packet capture method. Although, Advanced-PCA that they proposed has problems. Advanced-PCA doesn't display raw data of detection attack packets as well as can't detect exact zero-day attack packets. Therefore, we propose a system that can solve these problems.

III. PROPOSED SYSTEM

Figure 2 is architecture of proposed system.

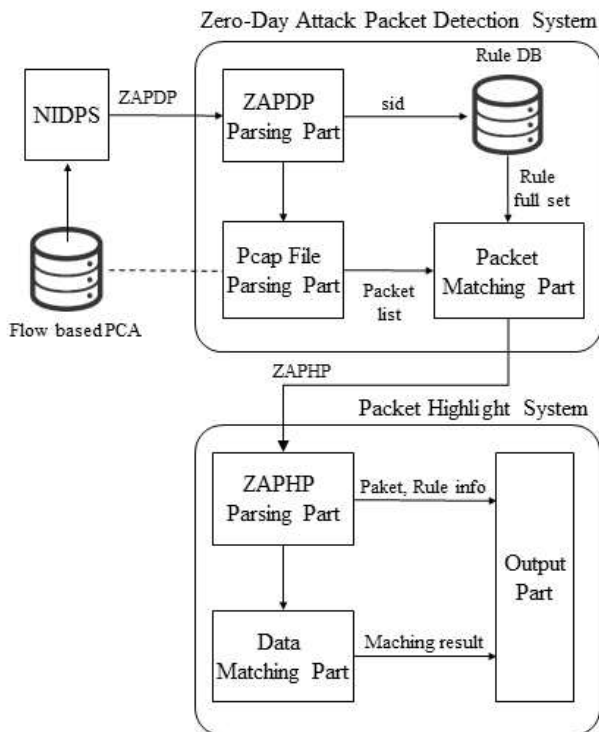


Figure 2. Architecture of proposed system

Proposed system consists of Zero-Day Attack Packet Detection System and Packet Highlight System. Also, it uses ZAPDP (Zero-day Attack Packet Detection Protocol) and ZAPHP (Zero-Day Attack Packet Highlight Protocol) for communication between internal systems. First of all, Flow based PCA sends stored packets when new rules are updated. NIDPS inspects received packets from Flow based PCA. If there are matched packets with updated rules in inspection result, it makes the packet to format of ZAPDP and sends to Zero-Day Attack Packet Detection System.

Zero-Day Attack Packet Detection System performs deep packet inspection on packets in flow. ZAPDP Parsing Part parses ZAPDP after receives it. When parsing is done, ZAPDP Parsing Part sends sid value to Rule DB and timestamp value to Pcap File Parsing Part. Rule DB searches rule full set matched received sid value and sends it to Packet Matching Part. Pcap File Parsing Part parses pcap files and

searches flow with received timestamp value. When matching is finished, Pcap File Parsing Part sends packets in the flow to Packet Matching Part. Packet Matching Part searches packet which is matched with the rule full set in received packets from Pcap File Parsing Part. If Packet Matching Part finds packet that matches the rule, it transmits to Packet Highlight System the packet information into ZAPHP format.

Packet Highlight System displays zero-day attack packet information and raw data of the packet matched with new rule for users to check. ZAPHP Parsing Part parses ZAPHP after receives it. After then, ZAPHP Parsing Part sends information for matching raw data detected in zero-day attack from full packet matched with new rule to Data Matching Part. Also, it sends sid, timestamp to Output Part. Data Matching Part sends matched result to Output Part after it compares payload of full packet and received information (Start Byte field, End Byte field, Matching Data field, Full Packet field). Output Part displays information which is received from ZAPHP Parsing Part and Data Matching Part. So, users can conform information about zero-day packet, rule and matched raw data.

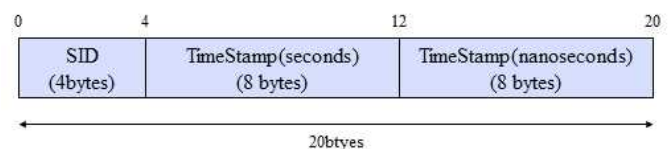


Figure 3. Zero-Day Attack Packet Detection Protocol Format

Figure 3 is format of Zero-Day Attack Packet Detection Protocol. ZAPDP is communication protocol for detecting zero-day attack packet within flow that includes it and ZAPDP has fixed size of 20 bytes. SID (Signature ID) field is 4 bytes size and indicates assigned ID each rule of NIDPS. TimeStamp field has time value that packet is received. It is 16 bytes size and consists of TimeStamp(seconds) of 8 bytes and TimeStamp(nanoseconds) of 8 bytes. TimeStamp(seconds) field indicates value from year to second (year, month, day, hour, minute, second). TimeStamp(nanoseconds) field indicates value in units of nanoseconds less than the second unit.

Figure 4 is format of Zero-Day Attack Packet Highlight Protocol. ZAPHP is communication protocol for extracting raw data judged as security threat of zero-day packet and displaying the information to users. ZAPHP consists of header which has fixed size of 40 bytes and payload of unfixed size. SID and TimeStamp field of ZAPHP mean same as those of ZAPDP. Length field indicates payload size of ZAPHP and value is in bytes. For example, if Length field value is 1000, payload size of ZAPHP is 1000 bytes. Packet Offset field indicates where Full Packet field begins. For instance, if Packet Offset field value is 100, position of Full Packet field starts with the 100th bytes. Start Byte field indicates first location of raw data which matched with updated new rule. Similarly, End Byte field indicates last location of raw data which matched with updated new rule. Reserved field reserved for future use and it should be zero. Matching Data field indicates raw data that matched with updated new rule in

payload of full packet. Also, it must be hexadecimal value which can convert to ASCII(American Standard Code for Information Interchange). Full Packet field indicates full packet that matched with updated new rule and must be hexadecimal value which can convert to ASCII.

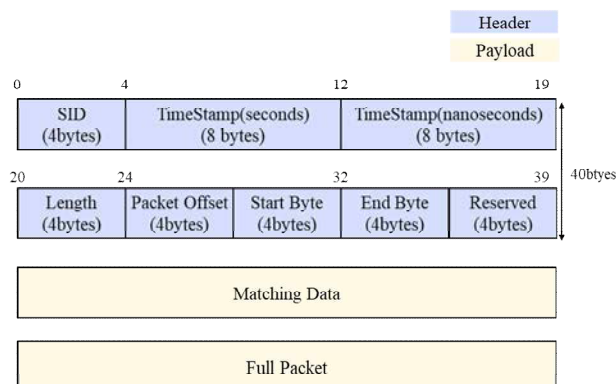


Figure 4. Zero-Day Attack Packet Highlight Protocol Format

IV. EXPERIMENT

To experiment about proposed system, authors presuppose that packet shown in figure 5 is zero-day attack packet.

No.	Time	Source	Destination	Protocol	Length
34	2020-08-24 00:00:00.001306730	220.66.1.5	210.119.144.5	DNS	72
Frame 34: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) Ethernet II, Src: Cisco_8b:46:40 (58:0a:20:8b:46:40), Dst: Prolink_82:11:17 (00:06:c4:82:11:17) Internet Protocol Version 4, Src: 220.66.1.5, Dst: 210.119.144.5 User Datagram Protocol, Src Port: 63412, Dst Port: 53 Domain Name System (query) Transaction ID: 0x326b Flags: 0x0100 Standard query Questions: 1 Answer RRs: 0 Authority RRs: 0 Additional RRs: 0 Queries [Response In: 36]					
0000	00 06 c4 82 11 17 58 0a	20 8b 46 40 08 00 45 00	-----X- f0--E-		
0010	00 3a 51 9f 00 00 7e 11	ab 4f dc 42 01 05 d2 77	--Q-----0-B---w		
0020	90 05 f7 b4 00 35 00 26	06 be 32 6b 01 00 00 01	-----5-8 -2k---		
0030	00 00 00 00 00 02 61	70 05 6c 69 6a 69 74 03	-----a p-11jit-		
0040	63 6f 6d 00 00 1c 00 01	com-----			

Figure 5. Presupposed Packet that Zero-Day Attack

Figure 6 is shown that updated rule which can detect zero-day attack packet.

sid	rule full set
255	alert udp \$EXTERNAL_NET any -> \$HOME_NET any (msg:"zero-day attack"; content:"[32 6b]";sid:255);

Figure 6. Rule for Detecting Zero-Day Attack Packet

New rule can detect packet which used UDP(User Datagram Protocol) at transport layer and contained hexadecimal value '32 6b' in payload. When new rule is updated at NIDPS, NIDPS sends ZAPDP to Zero-Day Attack Packet Detection System.

ZAPDP		
SID	TimeStamp(seconds)	TimeStamp(nanoseconds)
00 00 00 ff 00 00 12 5f 5e f0 4e 00 00 00 00 00 00 00 00 00 00		

Figure 7. ZAPDP in Experiment

Figure 7 is shown that ZAPDP which is received at Zero-Day Attack Packet Detection System in experiment. In experiment, converting SID of ZAPDP to decimal makes '255'. Likewise, TimeStamp(seconds) is '20200824000000' and TimeStamp(nanoseconds) is '0'. The value of TimeStamp fields means that it was received between '2020-08-24 00:00:00' and '2020-08-24 00:00:01' ('2020-08-24' is means August 24, 2020).

Zero-Day Attack Packet Detection System which is received ZAPDP searches zero-day attack packet in flow-based PCA using TimeStamp and rule full set that sid is '255'.

```

No: 34 caplen: 72 length: 72
=====ETHERNET Header=====
dest mac: 0x00 06 c4 82 11 17 src mac: 0x58 0a 20 8b 46 40 type: 0x800

=====IPv4 Header=====
src: 220.66.1.5, dst: 210.119.144.5
header length: 20 bytes, version: 4, total length: 58 bytes
id: 20895, last fragment, offset: 0 checksum is correct, TTL: 126

UDP

=====UDP Header=====
src port: 63412
dest port: 53
len: 38
start byte = 42
end byte = 43
target value : 32 6b

payload :
32 6b 01 00 00 01 00 00 00 00 00 02 61 70 05 6c 69 6a 69 74 03 63 6f
6d 00 00 1c 00 01
  
```

Figure 8. Detection Result of Zero-Day Attack Packet Detection System

Figure 8 is shown that detection result of Zero-Day Attack Packet Detection System. Raw data matched with rule begins at the 42th byte and ends at the 43th byte (packet begins at the 0th byte). After Zero-Day Attack Packet Detection System makes detection result into ZAPHP format, it sends ZAPHP to Packet Highlight System.

ZAPHP				
SID	TimeStamp(seconds)	TimeStamp(nanoseconds)		
00 00 00 ff 00 00 12 5f 5e f0 4e 00 00 00 00 00 00 00 00 00 00				
Length	Packet Offset	Start Byte	End Byte	Reserved
32 6b				
Matching Data				
00 06 c4 82 11 17 58 0a 20 8b 46 40 08 00 45 00 00 3a 51 9f				
Full Packet				
00 00 7e 11 ab 4f dc 42 01 05 d2 77 90 05 f7 b4 00 35 00 26				
06 be 32 6b 01 00 00 01 00 00 00 00 00 02 61 70 05 6c 69				
6a 69 74 03 63 6f 6d 00 00 1c 00 01				

Figure 9. ZAPHP in Experiment

Figure 9 is shown that ZAPHP which is received at Packet Highlight System in experiment. SID value of ZAPHP decimal means '255'. Likewise, TimeStamp(seconds) is '20200824000000' and TimeStamp(nanoseconds) is '001306730'. TimeStamp fields means time that zero-day packet is received. Also, it is same packet as figure 4. Length field means that payload size is 74 bytes and Packet Offset field means that Full Packet field begins at the 44th byte. Also, it can be checked that raw data matched with rule begins at the 42th byte and ends at the 43th byte by Start Byte field and End

Byte field. Lastly, ZAPHP includes raw data matched with rule and full packet data.

Packet Highlight System displays full packet by highlighting data that matches with rule.

```
sid : 225
rule full set :
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"zero-day attack";
content:"132 bbl";sid:255;)

packet no. : 34
timestamp : 2020.08.24 00:00:00.001306730

full packet :
00 06 c4 82 11 17 58 0a 20 8b 46 40 08 00 45 00 00 3a 51 9f 00 00 7e
11 ab 4f dc 42 01 05 d2 77 90 05 f7 b4 00 35 00 26 06 be 32 6b 01 00
00 01 00 00 00 00 00 02 61 70 05 6c 69 6a 69 74 03 63 6f 5d 00 00
1c 00 01
```

Figure 10. Output of Proposed System

Figure 10 is shown that output of Packet Highlight System. Users can check information about rule, zero-day packet and raw data that matches with rule.

V. CONCLUSIONS

In this paper, we propose Zero-Day Attack Packet Highlighting System. Proposed system outputs zero-day attack packet information from flow extracted as result of regression inspection of packets stored in flow-based PCA. It also highlights raw data of the packet matched with rule. It performs a DPI regression inspection to solve problems in existing security systems. Also, we design communication protocols for sending and receiving data within proposed system. We tested proposed system through an experiment. As a result, proposed system could solve problems of flow-based PCA and NIDPS. Purpose of the proposed system is to solve existing flow-based problems and provides users with raw data information of zero-day packets so that they can analyze raw data for the packets.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1A2C1006167).

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2020R1A6A1A12047945)

* Corresponding Author : S.G. Choi(choisg@cbnu.ac.kr)

REFERENCES

- [1] Jang Hyeon Jeong, Si Young Lee, Gyung Ho Park, Seong Gon Choi, "Rule conversion mechanism between NIDPS engines", 2020 22nd International Conference on Advanced Communication Technology (ICACTION), Feb. 2020, pp.581-584.
- [2] Akash Garg and Prachi Maheshwari, "A hybrid intrusion detection system: A review", 2016 10th International Conference on Intelligent Systems and Control (ISCO), Jan. 2016.
- [3] Alaa Hussein Al-Hamami, Ghossoon M. Waleed Al-Saadoon, "Development of a network-based: Intrusion Prevention System using a Data Mining approach", 2013 Science and Information Conference, pp. 641-644, Oct. 2013.
- [4] Nisreen Innab, Eman Alomairy, Lamy Alsheddi, "Hybrid System Between Anomaly Based Detection System and Honeypot to Detect Zero Day Attack", 2018 21st Saudi Computer Society National Computer Conference (NCC), April 2018.
- [5] Masaaki Sato, Hirofumi Yamaki, Hiroki Takakura, "Unknown Attacks Detection Using Feature Extraction from Anomaly-Based IDS Alerts", 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet, July. 2012.
- [6] Osborn N. Nyasore, Pavol Zavarsky, Bobby Swar, Raphael Naiyeju, Shubham Dabra, "Deep Packet Inspection in Industrial Automation Control System to Mitigate Attacks Exploiting Modbus/TCP Vulnerabilities", May 2020, pp.241-245
- [7] Kiho Nam, Keecheon Kim, "A Study on SDN security enhancement using open source IDS/IPS Stricata", 2018 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1124-1126, Oct. 2018.
- [8] Satyendra Kumar Patel, Abhilash Sonker, "Internet Protocol Identification Number Based Ideal Stealth Port Scan Detection Using Snort", 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), pp. 422-427, Dec. 2016.
- [9] Tomas Zitta, Marek Neruda, Lukas Vojtech, Martina Matejkova, Matej Jehlicka, Lukas Hach, Jan Moravec, "Penetration Testing of Intrusion Detection and Prevention System in Low-Performance Embedded IoT Device", 2018 18th International Conference on Mechatronics - Mechatronika (ME), Dec. 2018.
- [10] Xiangning Hou, Zhiping Jiang, Xinli Tian, "The detection and prevention for ARP Spoofing based on Snort", 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), vol. 5, pp. 137-139, Oct. 2010.
- [11] Hussein Alnabulsi, Md Rafiqul Islam, Quazi Mamun, "Detecting SQL injection attacks using SNORT IDS", Asia-Pacific World Congress on Computer Science and Engineering, Nov. 2014.
- [12] Jang Hyeon Jeong, Seong Gon Choi, "SCA System for minimizing damage of Zero-Day attack", KICS Winter Conference 2020, Feb. 2020, pp. 100-101
- [13] (2019) The Xabyss website [Online]. Available: <http://www.xabyss.com/>
- [14] Jang Hyeon Jeong, Seong Gon Choi, "Feedback System to Minimize Damage by Zero-Day High-Volume Attack based on NIDPS", KICS Summer Conference 2020, Aug. 2020, pp. 92-93
- [15] S. A. Zonouz, R. Berthier, H. Khurana, W. H. Sanders and T. Yardley, "Seclius: an information flow-based consequence-centric security metric", Parallel and Distributed Systems IEEE Transactions, 2015, vol. 26, pp. 562-573.
- [16] Farag Azzedin, Husam Suwad, Zaid Alyafeai, "Countermeasuring Zero Day Attacks: Asset-Based Approach", 2017 International Conference on High Performance Computing & Simulation (HPCS), July 2017, pp.854-857
- [17] Yehuda Afek, Anat Bremner-Barr, Shir Landau Feibish, "Zero-Day Signature Extraction for High-Volume Attacks", IEEE/ACM Transactions on Networking, Vol: 27, Issue: 2, April 2019, pp.691-706
- [18] (2020) sentrywire, Available: <https://www.sentrywire.com/packet-capture>.
- [19] Cisco Visual Networking Index, "Cisco Visual Networking Index: Forecast and Trends, 2017-2022", Fed. 2019
- [20] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Vijay Sivaraman, Arun Vishwanath, "Low-cost flow-based security solutions for smart-home IoT devices", 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Nov. 2016
- [21] John A. Copeland, John Jerrim, "Packet sampling flow-based detection of network intrusions", U.S. Patent 7 512 980 B2, Mar. 31, 2009
- [22] John A. Copeland, "Flow-based detection of network intrusions" U.S. Patent 7,185,368 B2, Feb. 27, 2007
- [23] Seung Min Ryu, Si Young Lee, Chang woo Ryu, Seong Gon Choi, "A Method of the First-N Packets Storage by Using the Advanced-PCA" Contemporary Engineering Sciences, Vol. 9, 2016, no. 14, pp.647-653
- [24] John Hastwell, Marco E. Foschiano, Ravichandra RyaliNirmalendu Das, "Event-based capture of packets from a network flow", U.S. Patent 8 958 318 B1, Feb. 17, 2015

- [25] L.J.G. Villalba, A.L.S. Orozco, J.M. Vidal, "Malware Detection System by Payload Analysis of Network Traffic", IEEE Latin America Transactions, March 2015, pp.850-855
- [26] Kiho Nam, Keecheon Kim, "A Study on SDN security enhancement using open source IDS/IPS Suricata", 2018 International Conference on Information and Communication Technology Convergence (ICTC), Oct. 2018, pp. 1124-1126
- [27] Eugene Albin, Neil C. Rowe, "A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems", Eugene Albin;Neil C. Rowe, March 2012, pp.122-127
- [28] Kevin Wong, Craig Dillabaugh, Nabil Seddigh, Biswajit Nandy, "Enhancing Suricata intrusion detection system for cyber security in SCADA networks", 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), May 2017



Jang Hyeon Jeong received B.S. degree in the College of Electrical & Computer Engineering, Chungbuk National University, Korea in 2019. He is currently a M.S. candidate in School of Electrical & Computer Engineering, Chungbuk National University. His research interests include Network Security, Smart Grid.



Jong Beom Kim received B.S. and M.S. degree in the College of Electrical & Computer Engineering, Chungbuk National University, Korea in 2017 and 2019. His research interest is network programming, He is currently researcher in Xabyss Inc. His research interest is network security.



Seung Gon Choi received B.S. degree in Electronics Engineering from Kyeongbuk National University in 1990, and M.S. and Ph.D. degree from Information Communications University, Korea in 1999 and 2004, respectively. He is currently an associate professor in College of Electrical & Computer Engineering, Chungbuk National University. His research interests include smart grid, IoT, mobile communication, high-speed network architecture and protocol.