An approach for detecting encrypted insider attacks on OpenFlow SDN Networks

Charles V. Neu*, Avelino F. Zorzo§, Alex M. S. Orozco† and Regio A. Michelin‡
*University of Santa Cruz do Sul (UNISC), §Pontifical University of Rio Grande do Sul (PUCRS),
†Sul-rio-grandense Federal Institute (IFSul), ‡Federal Institute of Rio Grande do Sul (IFRS)
e-mail: *charles1@unisc.br, §avelino.zorzo@pucrs.br, †orozco@sapucaia.ifsul.edu.br,
‡regio.michelin@restinga.ifrs.edu.br

***Abstract*—Data traffic on the Internet is growing continuously due to the high number of connected devices and increased number of applications and transactions performed online. To ensure information security, integrity and confidentiality, cryptography is applied over transmitted or stored data. Hence, even if an attacker capture data packets, its reading is hampered or not even possible. However, an attacker can also use cryptography to mask an attack in order to avoid detection, for example, by an Intrusion Detection System (IDS). Recent studies in network technologies introduced a new paradigm called Software Defined Networking (SDN). By decoupling data and control plans, the SDN architecture allows centralizing the network management, intelligence and control into a single point, called Controller. The OpenFlow protocol, widely adopted in SDN, provides specific messages to get statistical information of an OpenFlow switch. A Controller can request this information, which enables the development of new IDS models to detect encrypted attacks. In this work, we intend to identify encrypted insider attacks in SDN by developing a new IDS approach that can detect encrypted attacks.**

***Index Terms*—Cryptography, encrypted attacks, insider attacks, SDN, security, network attack, IDS, OpenFlow, OpenDaylight.**

I. Introduction

Computer networks are broadly demanded due to the increasing popularity of the Internet access. Thus, several devices need to establish communication in order to exchange information. However, this usage expansion also increases the possibility of malicious utilization of the network that can be promoted by either systems or people. Usually, that activity aims to damage the correct network behavior, using the available resources harmfully or illegally, or even to obtain information without proper authorization [1].

In order to prevent that, Intrusion Detection Systems (IDS) [2] are used to monitor, identify, register and report systems and/or networks managers when some suspect activity is detected. Those systems analyze packet information on the network to define if they could be malicious or not [3].

In the past years, a wide range of attacks were described in the literature. One of such attacks are insider attacks [4], or known as insider threats. Those attacks could be used, for example, to steal sensitive data or to damage a company's image. Moreover, an insider may also be able to compromise system availability by overloading computer resources, like network, storage or processing capacity, performing, for example, Denial of Service (DoS) attacks [5], which can lead to system crashes. Insiders that perform attacks may have authorized system access and may also know the network architecture and system policies and procedures, which give them an advantage over external attackers. Besides, normally, organizations focus on protecting this system from external attacks and do not consider inside intruders [3].

On way to reduce the chance for either internal or external attacks, would be to provide communication using cryptography. When using cryptography, even if an attacker is able to capture network packets, if the data is transmitted using cryptography, its reading will be hampered or not even possible. Although cryptography reduces overall chances of successful attacks, an attacker could also use cryptography in order to mask an attack. As a consequence, usually this ciphered attack will bypass the protection systems, since traditional IDS do not analyze ciphered packets. Indeed, to the best of our knowledge, there are no efficient models of encrypted insider attack detection found in the literature.

Furthermore, the concerns with security and data privacy imply an increase in the use of encrypted traffic. In this way, the traffic of ciphered data tends to be a standard for applications and systems on the Internet in a near future, or even nowadays. Besides, recent advances of network technologies result in a new network paradigm, called Software Defined Networking (SDN) [6]. Due the dissociation between the data and control planes, the SDN architecture allows to centralize network intelligence, control and management on an entity called Controller [7]. This centralization gives a global view of the network to the Controller, which enables the development of new IDS approaches to detect encrypted attacks.

Therefore, this paper proposes a novel approach to detect encrypted insider attacks on OpenFlow[8] SDN networks. The remain of this paper is organized as follows. Section II presents some theoretical concepts as well as some related work that help to understand the current trends and challenges on encrypted data detection. Section III describes the proposed approach. The proposed architecture and evaluation are described in Section IV. Section V brings the conclusion and future directions that are being performed in order to achieve the research goal of this paper.

978-1-908320/73/5/$31.00 © 2016 IEEE

II. Background

This section gives a brief introduction on the main topics related to the new approach being proposed, *i.e.*, SDN, cryptography, IDS and related work.

A. Software Defined Networking

Recent studies in network technology point to a new network paradigm called Software Defined Networking (SDN) [6]. This paradigm is based on the separation of data plane and its control. The SDN network architecture allows the intelligence, control and its management to be centralized in a single entity that is called Controller [9] [10].

The Controller is basically a software application responsible for taking decisions related to network management. This management consists of adding and removing entries (that define network routes) in the flow table. Despite of that, it also acts as a physical abstraction, facilitating application development and services responsible for network flow management. It works as a single logical switch used by applications, security mechanisms and management. In an SDN, the Controller acts as a network management software [7] [10], which centralizes the management and control tasks[6] [7]. This brings several advantages [11], for example:

- Through this centralization, network policy modification becomes simpler and less error prone. It can even be compared to the low level device configuration. This is possible due to the modification through high level languages and software components.
- A control program can automatically react to any forged change on the network state, keeping the high level policy straightforward.
- The Controller logical centralization in a single Controller that knows all the network state, simplifies the development of function, services and sophisticated network applications.

In order to build a new SDN network, it is possible to use an existent Controller or even customize it. Several Controllers are described in the literature. Khondoker *et al.* [11] describe some of the main Controllers, which are POX, RYU, Floodlight and OpenDaylight.

B. Cryptography

Cryptography is the science of writing secret code. The messages to be encrypted, called plaintext, are transformed by a function that is parameterized using a key. The result of this encryption process is the ciphertext, which will be the transmitted data. An intruder may be able to hear and to accurately copy the complete ciphertext, but as the intruder does not know the decryption key, data will still be protected [12] [13].

According to Kumar *et al.* [13] and Ferguson *et al.* [12], a lot of encryption techniques are used on communications in order to make them more secure, like AES, RSA, ECC or Blowfish. The goal is to ensure that four information security principles will be respected:

- Privacy: only the authorized recipient can read the message content, *i.e*, to understand a message the decipher key is needed.
- Authentication: the recipient must be able to identify the sender and verify that it was him who sent the message. It proves their identities.
- Integrity: the recipient must be able to determine that the message has not been modified or altered from its original form during the transmission.
- Non-repudiation: ensure that the sender cannot deny the authorship of the message and the message was received by the specified person.

Another important feature of a secure system, is an efficient access control, which consists on managing who is authorized to access a resource and under which restrictions and conditions. This access control can have different levels of security.

C. Intrusion Detection

An intrusion attack can be defined as a set of actions that attempt to commit resources of a computer system or numerous attempts to exploit any kind of information, regardless of whether successful or not. These attacks aim to corrupt the privacy, integrity and authenticity, which are three of the principles of information security [14] (see Section II-B). An intruder can explore a lot of weaknesses on security systems, protocols, applications or settings, by using specific techniques and tools. Moreover, it can be performed based on social engineering, where an attacker exploits a user who can grant access to the resource (a password or other information that compromise the security of the network and allows access to it), tricking them in order to reach the attacker goals [14].

When using intrusion techniques, attackers exploit vulnerabilities in the implementation of systems, services, protocols, and others. There are also the problems generated by users and administrators, such as miss configuration and improper maintenance of the systems, inefficient passwords and outdated systems. Intrusion attacks are usually intended to steal or damage data [14].

Intrusion attacks can be classified according to their nature, motivated by insider and outsider threats [4]. On one hand, outsider threats are generally outside the corporation (rivals, enemies or criminals) and they have limited opportunity to carry out their attacks. Outside attackers can only gain access by exploiting gaps or weaknesses in protection systems. On the other hand, insider threats have privileged access that enables them to cause serious consequences, compared to outsiders. Normally, the access that enables insider attackers to cause so much damage is also essential to enable them to do their propose.

Usually, insiders threats can be classified by unintentional threats and malicious threats. Unintentional threats are insiders who accidentally expose the organization data or the whole organization Information Technology (IT) infrastructure. Malicious threats are insiders who promote IT sabotage, theft of intellectual property or fraud [15]. Malicious insiders can be

involved in different activities, such as unauthorized extraction, exfiltration of data, tampering with data or resources of an organization, destruction or deletion of critical data and assets, eavesdropping and packet sniffing with will intend and impersonation of other users via social engineering [16].

The internal attacks may not be result of a single problem, but of a set of small failures or vulnerabilities. Failures in safety procedures may allow users to find bugs that allow access to materials and tasks that they would not have authorization to. Incomplete or outdated documentation and poor access and permissions control can also contribute to insider attacks [14].

Furthermore, network architectures and current systems are becoming more complex, making them even more vulnerable to this kind of attack, since they are more difficult to manage and therefore it is easier for the manager to "forget" to set some important security feature in the network or system. One of the main motivations of inside attackers, is the sale of sensitive data, for example, in banking or e-commerce, for illicit enrichment [16] [4].

D. Intrusion Detection System

Intrusion attacks occur in several forms and in all the layers of the TCP/IP model. However, there are systems that perform functions such as sensors and event analyzers, intended to detect, analyze and identify malicious attempts [3] [2]. These systems are called Intrusion Detection System (IDS) and they usually use two main detection approaches:

- Signature-based: Using this approach, an IDS uses a database with information about known attacks. To identify an intrusion attempt, the content of each packet is analyzed, by searching for a set of characters that identifies the attack. This set of characters is called Attack Signature [3] [17].
- Anomaly-based: an IDS is able to identify an attack when some behavior is different from any pattern considered normal, for example, some application performing an attempt of unauthorized access to a system resource [3] [17] [2].

An IDS may be responsible to monitor a specific host, *i.e.* Host Intrusion Detection System (HIDS), or to monitor traffic on a network, *i.e.* Network Intrusion Detection System (NIDS). In addition to detect intrusions, IDS researchers are developing systems able to prevent them. In this case, the systems are called Intrusion Prevention Systems (IPS) [2].

E. Approaches for IDS over encrypted traffic

Similarly to traditional networks, an SDN signature-based IDS typically cannot analyze encrypted packets, because they need to analyze the payload data that is encrypted. However, anomaly-based IDS may be applied, using three main approaches: protocol-based, modification-based and statistical-based [3]:

- Protocol-based analysis: this approach, also know as stateful protocol analysis, searches deviations from the packets in each state of the protocol. Universal profiles that specify how a given protocol may or may not be used in data transfer are analyzed. They are based on the protocol specification of software and hardware vendors, and also official protocols standards. However, since this type of approach only analyses whether the protocol is being applied in a proper way, it is not possible to detect attacks that are being performed at the application layer, which are the most widely used.
- Modification-based: this approach consists on changing the encryption protocol and infrastructure to detect attacks in encrypted data on the network. Basically, the key (password) to encrypt and decrypt the data is distributed to the IDS. With this secret, the IDS can decipher the package payload and analyze it. However, this technique can turn the network vulnerable and the privacy principle may be broken. In addition, it may consume a lot of processing power, which may turn this method slow and even making it impossible to apply in real environments with large data traffic.
- Based on statistical analysis: it is also possible to develop intrusion detection methods using statistical analysis of observable parameters on encrypted data traffic. Moreover, Network Behavior Analysis (NBA) [18] methods can be applied. Some information, like source and destination IP address, besides the used ports, the header fields and payload size are analyzed. It allows detection of DoS attacks, scans, worms, network policy violations and not expected services or applications[19]. Another statistical approach for intrusion detection on encrypted data traffic is the development of methods based on packet flows, without payload analysis. In this way, some kinds of attacks, like scans, DoS, worms and botnets can be detected [20].

F. Related Work

In this section some related studies that describe methods to identify encrypted packets or encrypted flows are presented. Moreover, studies intended to identify encrypted attacks are presented as well.

a) Encrypted packets identification:

- Encrypted data can be detected using entropy calculation [21]. However, this method usually cannot distinguish between compressed or encrypted data, yielding a high false positive rate [22] if applied to encrypted data detection. Therefore, Thurne *et al.* [22] developed a novel approach to detect encrypted data with the help of statistical tests. A tool that implements several statistical tests that allow encryption detection on block-based storage devices was implemented. A multitude of statistical tests is used to classify blocks of input data as either encrypted or not encrypted. Besides, they suggest a workflow for device analysis. Their results show that this approach is able to differentiate between compressed and encrypted data allowing encrypted data detection.
- A system to identify Web pages that use encrypted packets, such as WEP, WPA and IPSec is described by

Bissias *et al.* [23]. Their results show a success detection rate around 23%, which can increase to 40% when a specific set of pages easily identifiable are analyzed.

- Wright [24] and Bar-Yanai [25] propose architectures to analyze protocols to perform traffic classification, allowing identification of encrypted connections. Their results show a detection efficiency up to 90% of correctness.

b) Encrypted attack identification:

- Sherry *et al.* [26] propose BlindBox, a system to enable Deep Packet Inspection (DPI) over encrypted traffic without requiring decryption of the underlying traffic. BlindBox is intended to support applications such as IDS, exfiltration detection and parental filtering. The BlindBox goal is to perform the DPI directly on the encrypted traffic through a new protocol and new encryption schemes.
- Koch *et al.* [3] propose a new architecture based on the use of inherent knowledge of data connections by calculation of their similarities. This architecture does not need a learning phase nor a complex configuration or knowledge about the service to protect. Their results show an accuracy of 74% when 1% of malicious traffic is injected and 72% when 2.7% of the packets are malicious. Their false alarm rate is over 26%.
- On their work, Foroushani *et al.* [27] present an IDS based on the analysis of packets size and time in the network. With these techniques, it is possible to analyze data at the application layer, but more extensive configurations or detailed server profiles settings are required. Moreover, this work presents high false positive rates, which are up to 47%.
- A novel method to improve cloud security is proposed by He *et al.* [28]. This method is intended to detect encrypted data exfiltration. Initially, they use DPI and sample entropy estimation to identify encrypted traffic. After that, they built the network behavior profile to determine the state of encryption. They chose the occurrence time, destination IP and port, network layer protocol and application layer protocol of encrypted traffic to represent network behaviors. In their experiments, this method took around 45 seconds to identify encrypted traffic and the detection rates are over 90%. The accuracy of determining the state of encryption was over 80% and the false positive rate is low.
- Some recent research papers describe proposals of intrusion detection mechanisms based on SDN capabilities. According to Jankowski *et al.* [7], four sample solutions are:
  Method 1: this method is based on assessment of the first packet transmitted to the Controller and is intended to detect DoS and probe attacks. It performs revisiting traffic anomaly detection using SDN.
  Method 2: this is a fuzzy logic-based information security management for SDN. The operation principle of this method is the evaluation of the threads level and is intended to detect DoS attacks.
  Method 3: this method proposes the profile creation using sFlow [29] and OpenFlow. It combines OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments in order to detect DDoS and worm propagation probe attacks.
  Method 4: this is a lightweight DDoS flooding attack detection using NOX/OpenFlow based on flow statistic collection. The method is intended to identify DDoS attacks performed by a botnet.
  The methods described by Jankowski [7] may be used to detect common types of malicious activities, such as denial of service, port scan, and malicious software propagation attempts. However, there were no methods describing SDN solutions able to detect encrypted malicious activities.

III. Proposed Approach

The main limitation of current IDS, found in the literature, is their inability to analyze encrypted data packets or encrypted flows. However, the new network model introduced by SDN, brings new possibilities to address this limitation, allowing the development of new and more effective intrusion detection methods. The OpenFlow protocol[8], widely adopted in SDN, provides specific messages to get statistical information of an OpenFlow switch. On SDN networks, the OpenFlow version 1.0 protocol [30] provides encryption through TLS [30].

A Controller can request some statistical information to an OpenFlow switch. Specific messages, called Read_State, can be used to collect statistics from the switch flow tables, ports and individual entries for each flow. It is possible to request statistical information about individual flows as well as aggregate information from multiple flows [8]. Table 1 shows the statistical information that an SDN Controller can request to an OpenFlow switch [30].

In this way, SDN allows the aggregation of statistics logs collected from network devices memory and forwarding them to the Controller. Those data can be used as a data source for intrusion detection methods. For detection, our proposed IDS uses some OpenFlow provided statistic features like average bytes per flow, average packets per flow, grow of single flows, grow of different ports, percent of pair-flow and average of flow duration. Besides, destination and source IP address and port numbers of transport layer will be used in order to match traffic flows.

Initially, it is necessary to identify encrypted flows, which are under TLS connections. On IPv6 connections, the OpenFlow protocol defines that encrypted payloads have an extension header with the flag OFPIEH_ESP set to 1. By default, TLS connections are done trough the port 6653 [8]. Although these fields could be masked, we can use a strategy to detect encrypted data, based, for example, on the approach presented by Thurne *et al.* [22].

Then the OpenFlow switch sends the flows to the Controller. After this, the flow may be sent to the flow information logger in order to extract the features using the new flow, stores the flow information, and sends the features to our proposed

TABLE I
OPENFLOW INFORMATION [30]

Counter	bits
Per table	
Active entries	32
Packet lookups	64
Packet matches	64
Per flow	
Received Packets	64
Received Bytes	64
Duration (seconds)	32
Duration (nanoseconds)	32
Per port	
Received Packets	64
Transmitted Packets	64
Received Bytes	64
Transmitted Bytes	64
Receive Drops	64
Transmit Drops	64
Receive Errors	64
Transmit Errors	64
Receive Frame Alignment Errors	64
Receive Overrun Errors	64
Receive CRC Errors	64
Collisions	64
Per queue	
Transmit Packets	64
Transmit Bytes	64
Transmit Overrun Errors	64

statistical-based IDS. This IDS performs anomaly detection to verify if the flow has normal or malicious behavior.

The presented approach is based on the flows classification using statistical features from the transport layer level. Hence, it is possible to identify a specific connection representing the unauthorized action that may characterize a malicious activity flow from an insider.

IV. ARCHITECTURE AND EVALUATION

Most of the methods for intrusion detection found in literature use the KDD99Cup dataset [31] for testing and validation. However, this dataset is very old and therefore not suitable for our approach. Besides, recent datasets that provide complete packets with and without insider attacks are not available. Furthermore, we could not find, also, any dataset with encrypted packets or flows. Therefore we intend to develop a new testbed in order to build a new dataset with legitimate flows, flows that contain attacks and encrypted flows.

This testbed will be based on a Mininet [32] architecture, which allows a realistic virtual network creation, running real kernel, switch and application code, on a single machine to emulate SDN networks. Mininet creates virtual hosts by using a process-based virtualization method and the network namespace mechanism.

Furthermore, the OpenDaylight platform (ODL) [33], which provides a flexible common platform underpinning a wide breadth of applications and use cases, will be used as a SDN Controller. It can provide centralized control for any SDN architecture, regardless of hardware and software vendors, and its Controller is open-source. The proposed network topology is illustrated in Figure 1.

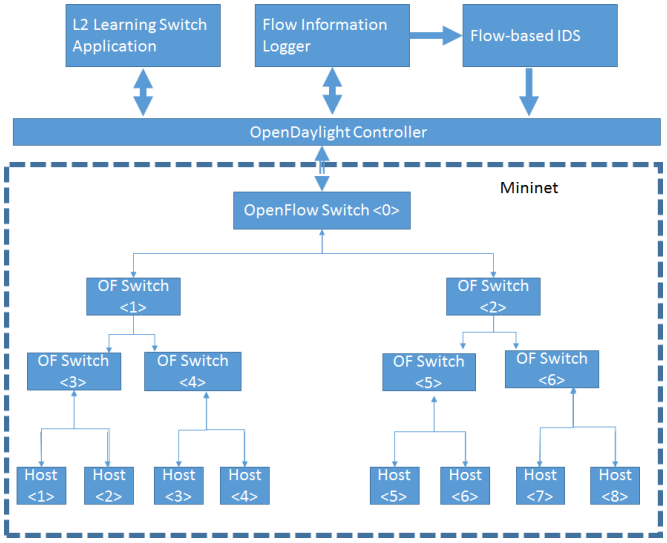

Fig. 1. Proposed network topology

Initially, a traffic-generator is used to inject normal and encrypted flows. Besides, some insider attacks will be injected as well. These insider attacks will also be encrypted. Therefore, four types of flows will be produced by our traffic-generator.

In the next step, a method to identify encrypted flows[22] is applied. This is an important step because our approach is intended only to identify encrypted insider attacks. When the attacks do not use encryption, any other IDS, like Snort [17], could be used.

After that, a statistical information collector is used to get important information about the flows (see Table I. Finally, our proposed IDS is used to perform insider intrusion detection on the encrypted flows based on the collected statistics. In our IDS, we also provide an algorithm to block malicious encrypted flows and to generate alerts in the case of a recognized attack. Those steps, are illustrated in Figure 2.

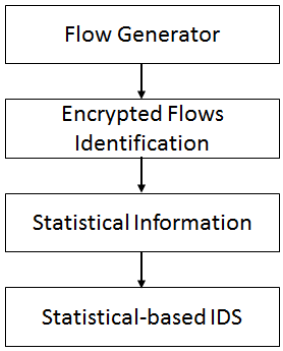

Fig. 2. Proposed methodology steps

In order to evaluate our method, it is important to analyze the accuracy, the false alarm rate as well as the system overhead introduced on the network. To measure

network overhead, OpenFlow messages, such as *Read_State*, *flow_stats_request* and *flow_removed* messages, are used to collect flows information.

V. Conclusion and future work

Several applications use cryptography to ensure security on information that is transmitted through a network. However, insider attacks may also be executed using encrypted packets or encrypted flows to bypass protection systems, like traditional IDS and Firewall. Therefore, since current IDS do not detect attacks on encrypted data, the development of a new IDS is necessary.

This paper presented an approach to identify encrypted insider attacks on SDN OpenFlow networks. This method is based only on statistical information requested by an SDN OpenDaylight Controller to the OpenFlow switches. This strategy will provide a lightweight IDS. As a future work, we will implement this method on a real SDN environment, creating a new IDS as described on this paper.

The development of the DPI analysis [26], which allows some encrypted payload information analysis, will help to improve our method by adding a second detection step. In this step, the flows selected as malicious will be redirected to a DPI IDS analysis in order to reduce false positives. In this case, our statistical-based IDS will perform the lightweight initial analysis, and redirect only suspicious flows to the packet-based DPI IDS.

Acknowledgment. Charles V. Neu receives a grant from Brazilian CAPES agency.

References

[1] M. Bhuyan, D. Bhattacharyya, and J. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 16, no. 1, pp. 303–336, 2014.

[2] F. Sabahi and A. Movaghar, "Intrusion detection: A survey," in *2008 Third International Conference on Systems and Networks Communications*, Oct 2008, pp. 23–26.

[3] R. Koch, M. Golling, and G. Rodosek, "Behavior-based intrusion detection in encrypted environments," *IEEE Communications Magazine, IEEE*, vol. 52, no. 14450733, pp. 124–131, 2014.

[4] R. Walton and W.-M. Limited, "Balancing the insider and outsider threat," *Computer Fraud & Security*, vol. 2006, no. 11, pp. 8–11, 2006.

[5] R. A. Michelin, A. F. Zorzo, and C. A. D. Rose, "Mitigating dos to authenticated cloud rest apis," *Internet Technology and Secured Transactions (ICITST), 9th International Conference for, IEEE*, pp. 106–111, 2014.

[6] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 4, pp. 2181–2206, 2014.

[7] D. Jankowski and M. Amanowicz, "Intrusion detection in software defined networks with self-organized maps," *Journal of Telecommunications & Information Technology*, pp. 3–9, 2015.

[8] O. N. Foundation, "Openflow switch specification, version 1.5.0," [Online] https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf, 2014.

[9] M.Mushi and R.Dutta, "Data-driven study of network administration in the evolving landscape of software defined networking," in *Proceedings of the 2014 Workshop on Human Centered Big Data Research*. New York, NY, USA: ACM, 2014, pp. 14:14–14:18.

[10] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "Sdn security: A survey," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, 2013, pp. 1–7.

[11] R. Khondoker, A. Zaalouk, R. Marx, and K. Bayarou, "Feature-based comparison and selection of software defined networking (sdn) controllers," *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*, pp. 1–7, 2014.

[12] N. Ferguson and B. & Schneier, *Practical Cryptography*. New York: John Wiley & Sons, 2003.

[13] M. G. V. Kumar and U. S. Ragupathy, "A survey on current key issues and status in cryptography," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, March 2016, pp. 205–210.

[14] J. R. Vacca, "Computer and information security handbook," 2012.

[15] D. M. Cappelli, A. P. Moore, and R. F. Trzeciak, *The CERT guide to insider threats: how to prevent, detect, and respond to information technology crimes (Theft, Sabotage, Fraud)*. Addison-Wesley, 2012.

[16] A. Sanzgiri and D. Dasgupta, "Classification of insider threat detection techniques," in *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, ser. CISRC '16, New York, NY, USA, 2016, pp. 25:1–25:4.

[17] Snort, "The snort project," [Online]. https://snort.org. AccessedonSeptember,2016, 2016.

[18] R. Koch, "Changing network behavior," *Third International Conference on Network and System Security, IEEE*, 2009.

[19] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (idps)," *NIST special publication*, vol. 800, no. 2007, p. 94, 2007.

[20] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of ip flow-based intrusion detection," *IEEE Communications Surveys Tutorials*, vol. 12, no. 3, pp. 343–356, Third 2010.

[21] I. Jozwiak, M. Kedziora, and A. Melinska, "Theoretical and practical aspects of encrypted containers detection-digital forensics approach," in *Dependable Computer Systems*. Springer, 2011, pp. 75–85.

[22] S. Thurner, M. Grn, S. Schmitt, and H. Baier, "Improving the detection of encrypted data on storage devices," in *IT Security Incident Management IT Forensics (IMF), 2015 Ninth International Conference on*, May 2015, pp. 26–39.

[23] G. Bissias, M. Liberatore, D. Jensen, and B. Levine, "Privacy vulnerabilities in encrypted http streams," *Proceedings of the 5th international conference on Privacy Enhancing Technologies*, pp. 1–11, 2005.

[24] C. V. Wright, F. Monrose, and G. Masson, "On inferring application protocol behaviors in encrypted network traffic," *The Journal of Machine Learning Research*, vol. 7, pp. 2745–2769, 2006.

[25] R. Bar-Yanai, M.Langberg, D.Peleg, and L.Roditty, "Realtime classification for encrypted traffic," *SEA'10 Proceedings of the 9th international conference on Experimental Algorithms , Springer-Verlag Berlin, Heidelber*, pp. 373–385, 2010.

[26] J. Sherry, C. Lan, R. A. Popa, and S. Ratnasamy, "Blindbox: Deep packet inspection over encrypted traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 213–226, aug 2015.

[27] V. A. Foroushani, F.Adibnia, and E.Hojati, "Intrusion detection in encrypted accesses with ssh protocol to network public servers," *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*, vol. 52, no. 98, pp. 314–318, 2008.

[28] G. He, T. Zhang, Y. Ma, and B. Xu, "A novel method to detect encrypted data exfiltration," in *Advanced Cloud and Big Data (CBD), 2014 Second International Conference on*, Nov 2014, pp. 240–246.

[29] L. Huang, X. Zhi, Q. Gao, S. Kausar, and S. Zheng, "Design and implementation of multicast routing system over sdn and sflow," in *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*. IEEE Conference Publications, 2016, pp. 524 – 529.

[30] O. N. Foundation, "Openflow switch specification, version 1.0.0," [Online] https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf, 2009.

[31] DARPA, "Kdd cup 1999 data," [Online]. http://www.kdd.ics.uci.edu/databases/kddcup99/task.htmlAccessedonSeptember,2016, 1999.

[32] R. L. S. de Oliveira, C. M. Schweitzer, A. A. Shinoda, and L. R. Prete, "Using mininet for emulation and prototyping software-defined networks," in *Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on*, June 2014, pp. 1–6.

[33] L. Foundation, "Opendaylight: Open source sdn platform," [Online]. https://www.opendaylight.orgAccessedonSeptember,2016, 2016.