ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Privacy Issues in Intrusion Detection System: Applications and Taxonomy

Jyoti
MCA, M. D. University, Rohtak

Abstract: Intrusion Detection Systems (IDSs) detect potential attacks by monitoring activities in computers and networks. This monitoring is carried out by collecting and analyzing data pertaining to users and organizations. The data is collected from various sources – such as system log files or network traffic– and may contain private information. Therefore, analysis of the data by an IDS can raise multiple privacy concerns. Recently, building IDSs that consider privacy issues in their design criteria in addition to classic design objectives (such as IDS' performance and precision) has become a priority. This article proposes a taxonomy of privacy issues in IDSs which is then utilized to identify new challenges and problems in the field. In this taxonomy, we classify privacy-sensitive IDS data as input, built-in and generated data. Research prototypes are then surveyed and compared using the taxonomy. The privacy techniques used in the surveyed systems are discussed and compared based on their effects on the performance and precision of the IDS. Finally, the taxonomy and the survey are used to point out a number of areas for future research.

Index Terms— Intrusion detection systems IDS, Privacy, Privacy preserving intrusion detection system, IDS privacy issues

I. INTRODUCTION

Intrusion Detection Systems (IDSs) are one of the most impor- tant defensive mechanisms in computer networks. These systems can detect and possibly prevent attacks and malicious activities which frontier security mechanisms, such as firewalls, often fail to catch.

Privacy issues in the field of computer security have been stud- ied extensively. However, there is no universal definition of pri- vacy. One of the most common definitions of privacy in information systems is: ''the claim of individuals, groups or institutions to de- termine for themselves when, how, and to what extent information about them is communicated to others'' [1].

Many countries have passed data protection and privacy laws which secure the right of privacy of their citizens. Such privacy laws include but are not limited to the US's HIPAA [2], the EU's data protection directive [3] and Canada's PIPEDEA [4]. It is imperative that new technologies and information systems not only consider users' demands and legal requirements for privacy, but also satisfy these requirements properly.

Privacy is very important to surveillance and monitoring sys- tems such as IDSs. To detect attacks in networks and computer systems, IDSs need to collect, store and analyze a wide scope of data. This data may contain private information and therefore the operation of an IDS may have privacy-related consequences.

In this article, we propose the first taxonomy, survey and future directions on privacy issues in IDSes.

1.1 Our contributions

A taxonomy of privacy issues in IDSs. We propose a taxon- omy which is based on three sources of private data: IDS input data (e.g. network traffic, log files), IDS built-in data (e.g. attack signatures or normal profiles) and IDS generated data (e.g. alerts or reports). For each source, we specify data fields which are concerns for privacy. Privacy issues for each data source are then discussed with illustrative examples. In this section, we present our proposed taxonomy of privacy issues in IDSs. This taxonomy is summarized in Fig. 1.

We observe that privacy-sensitive data in IDSs can be found in three different sources: IDS input data, IDS built-in data and IDS generated data. We present possible privacy-sensitive fields in each source. The privacy of these fields is determined by laws, the privacy policies and users' privacy demands.

Moreover, the privacy requirements for these fields can be divided into two main categories: preserving privacy of identifier fields such as user-names, IP addresses, etc., and preserving privacy of other non-identifier data such as URLs, time-stamps or even attack signatures. Discrimination between these two categories is useful in our taxonomy because different techniques are required to satisfy them.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Survey and classification of existing privacy preserving IDSs. We survey and classify existing privacy-preserving IDSs using our proposed taxonomy (Table 1). For each work, we examine the source of the private data, privacy sensitive fields and the tech- niques used to address the privacy requirements.

Comparison of privacy-preserving techniques. We also dis- cuss and compare different privacy-preserving techniques that can be used to ensure privacy in IDSs. Since these techniques address different privacy issues, it is important to know how they work and how they influence the IDS' performance and precision (false-positive and false-negative rates).

Future Directions. Finally, the taxonomy and survey are used to point towards a number of interesting areas for future research in the field.

1.2 Organization

This article is organized as follows: To better illustrate the issue of privacy in IDSs, in Section 2 we review a number of interesting cases in which preserving privacy is important. We present our proposed taxonomy of privacy issues in IDSs in Section 3. We then survey the related work based on the proposed taxonomy in Sec- tion 4, we discuss the privacy techniques used in surveyed systems and compare them based on their effects on IDS' performance and precision in Section 5, finally, challenges and future directions are presented in Section 6.

2. MOTIVATING SCENARIOS

One of the scenarios in which preserving privacy is of great importance is Collaborative Intrusion Detection (CID) [31–33]. Here, a number of IDSs wish to cooperate in order to detect distributed cyber attacks such as the spread of worms or denial-of-service attacks. Privacy concerns may arise since a single IDS's data – such as alert logs – have to be shared with other IDSs which may able to be accessed by unauthorized users. Sharing this data may risk exposing sensitive information such as intranet topology, network

In this section, we present our proposed taxonomy of privacy issues in IDSs. This taxonomy is summarized in Fig. 1.

We observe that privacy-sensitive data in IDSs can be found in three different sources: IDS input data, IDS built-in data and IDS generated data. We present possible privacy-sensitive fields in each source. The privacy of these fields is determined by laws, the privacy policies and users' privacy demands.

Moreover, the privacy requirements for these fields can be divided into two main categories: preserving privacy of identifier fields such as user-names, IP addresses, etc., and preserving privacy of other non-identifier data such as URLs, time-stamps or even attack signatures. Discrimination between these two categories is useful in our taxonomy because different techniques are required to satisfy them.

Table 1

Summary of surveyed papers (I.I.D., I.B.D. and I.G.D. stand for IDS input data, IDS built-in data and IDS generated data respectively).

Ref	Source of sensitive data	Privacy sensitive field	Privacy technique
[5]	I.I.D. (system logs)	User-IDs	Pseudonyms (simple mapping)
[6]	I.I.D. (system logs, network traffic)	User-IDs	Pseudonyms (Kerberos, MIX)
[7]	I.I.D. (firewall logs)	User-IDs, host names, IP addresses	Pseudonyms (simple mapping)
[8]	I.I.D. (system logs)	User-IDs	Pseudonyms (shamir secret sharing)
[9]	I.G.D. (IDS or firewall alerts)	IP addresses	Pseudonyms (hash functions)
[10]	I.G.D. (IDS alerts)	IP addresses	Bloom filters
[11]	I.G.D. (IDS alerts)	IP addresses, time	Statistical (concept hierarchies)
[12]	I.G.D. (IDS alerts)	IP addresses, ports, usernames, auth result	Hash, Paillier homomorphic encryption
[13]	I.G.D. (IDS alerts)	Packet payload	Statistical (Z-string)
[14]	I.I.D. (system logs)	User-IDs	Pseudonyms (Shamir secret sharing)
[15]	I.I.D. (system logs)	User-IDs, time	Homomorphic encryption
[16]	I.I.D. (network traffic)	IP addresses, port numbers, time intervals	Timeshifting, Pseudonyms, Statistical (perturbation)
[17]	I.G.D. (IDS alerts)	Packet payload, IP addresses	Bloom filters, Statistical (Z-string)
[18]	I.I.D. (network traffic)	Packet header, Packet payload	Two-tiered architecture with homomorphic encryption
[14]	I.I.D. (general audit data)	User-IDs, Time-stamps	Hidden time-stamp
[19]	I.G.D. (IDS alerts)	IP addresses	Cryptographic protocol
[20]	I.I.D and I.B.D (signatures)	Generic fields, Attack signatures	A 4-party priv. pres. architecture
[21]	I.G.D. (IDS alerts)	Policy driven	Removing, Pseudonyms
[22]	I.G.D. (IDS alerts)	IP Addresses	Bloom filter
[23]	I.I.D. (RFID tags)	RFID numbers	Cryptographic protocol
[24]	I.I.D. (network traffic)	Payload	HotItemID (sampling & hiding), multiset operation
[25]	I.G.D. (IDS alerts)	Packet header	Cryptographic protocol
[26]	I.G.D. (IDS log)	IP address	Statistical (I-Diversity)
[27]	I.G.D. (IDS alerts)	IP addresses, Packet payload (URL)	Secret sharing
[28]	I.I.D. (system log, network packet), I.B.D. (attack signatures)	Payload, Attack signatures	Oblivious deterministic finite automata
[29]	I.G.D. (IDS alerts)	Alert's attributes (e.g. IP address, time-stamp)	Homomorphic encryption
[30]	I.I.D., I.B.D	Files, Network signatures, IP addresses, Credentials.	Physical isolation and computational isolation

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

services, and security infrastructure to untrusted parties [34]. To mitigate this problem, sensitive information such as user identities or suspicious payloads in the IDS data should be obfuscated in such a way that the collaborative intrusion detection can be performed correctly while sensitive information is protected.

Another scenario in which IDSs cause a privacy risk is if a number of IDSs collaboratively generate attack signatures for widespread attacks (such as worms) by sharing their suspected packet payloads. Payloads can contain private information, and the signature generation should be performed in a way that privacy of collaborating IDSs is not compromised [20].

There is an increasing demand by organizations to outsource their security analysis to professional organizations. This involves security audit data and log files to be outsourced for security anal- ysis. However, there are plenty of privacy concerns about sharing sensitive information with third parties as it has been considered in [5,7,8,28].

3. A TAXONOMY OF PRIVACY ISSUES IN INTRUSION DETECTION SYSTEMS

In this section, we present our proposed taxonomy of privacy issues in IDSs. This taxonomy is summarized in Fig. 1.

We observe that privacy-sensitive data in IDSs can be found in three different sources: IDS input data, IDS built-in data and IDS generated data. We present possible privacy-sensitive fields in each source. The privacy of these fields is determined by laws, the privacy policies and users' privacy demands.

Moreover, the privacy requirements for these fields can be divided into two main categories: preserving privacy of identifier fields such as user-names, IP addresses, etc., and preserving privacy of other non-identifier data such as URLs, time-stamps or even attack signatures. Discrimination between these two categories is useful in our taxonomy because different techniques are required to satisfy them.

3.1 IDS input data

The IDS input data is any data – such as log files or network traffic – that is sent to or captured by an IDS for the purpose of intrusion detection. IDS input data may contain identifying fields such as IP addresses, usernames, host-names or any confidential information such as passwords, sensitive files that are private.

In host-based IDSs, input data come from system log files, while network-based IDSs capture input data from the network traffic. We discuss the privacy issues related to IDS input data.

3.1.1 Identifiers inside IDS input data

Identifiers inside input data are usually a source of privacy concerns. Due to privacy laws, the identities of users cannot be disclosed during intrusion analysis in certain situations. To better understand this issue, consider a sample log file of a web server which includes the HTTP requests made by users of a typical web application.

Suppose that this log file is sent to an IDS to be analyzed. During analysis, the IDS learns all user activities (e.g. which user- ID accessed which URL) which violates the users' privacy. Note that this violation occurs for both normal users as well as attackers. The question is how we can preserve the privacy of normal users while maintaining an effective intrusion analysis process? If each user- ID is replaced by a pseudonym before sending the log file to the IDS, then the IDS will not be able to observe user identities while still correctly can perform intrusion analysis operations. In this case, if an intrusion is detected, the IDS can report the attacker's pseudonym to the system administrator for re-identification of the pseudonym and recognition of the attacker's real identity. This technique, which is known as pseudonymization, has been used in a number of privacy-sensitive IDSs [5–9]. We will discuss the details of this technique in Section 5.1.

The identifiers inside IDS input data are not limited to the user- ID field. In network-based IDSs, identity fields could also be the source or destination IP addresses and it might be necessary to preserve the privacy of such fields during intrusion detection.

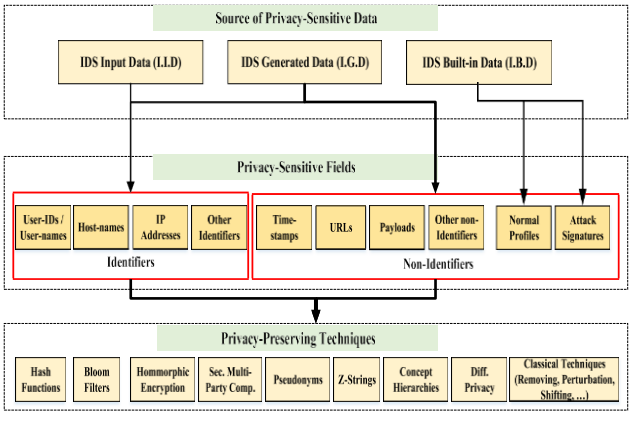

Fig. 1. A taxonomy of privacy issues in IDSs.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

3.1.2 Other privacy-sensitive information inside IDS input data

Sometimes private information can be found in non-identity fields such as files, URLs or any other information which is pro- cessed by the IDS. The URL field is an example of possibly pri- vate information. Consider the scenario in which an organization outsources its security analysis to a third-party IDS by sending log files and then receiving the security analysis result as it has been studied in a number of works [10,16,35]. There are privacy concerns about sharing sensitive information with third parties. URLs or other HTTP data may contain valuable information such as passwords or other personal information. Here, more elaborate techniques such as cryptographic protocols can be used to satisfy privacy requirements. Cryptographic techniques are discussed in Section 5.5.

3.2 IDS generated data

IDS generated data are the results of data analyzed by IDSs such as alerts. There exist some privacy-sensitive fields in IDS alerts. For example, when an IDS shares its alerts with other IDSs with the aim of collaborative intrusion detection or alert correlation, a large amount of information can be revealed to the other IDSs which could violate the privacy policies of many organizations.

3.2.1 Identifiers inside IDS generated data

When an IDS shares its alerts with other IDSs, a large amount of identity information can be revealed. Such identity information includes, but not limited to, user-IDs, IP addresses and host-names associated with a specific organization and its clients.

3.2.2. Other privacy-sensitive information inside IDS generated data

In some collaborative IDSs, non-identity fields such as attack payloads are shared among a number of IDSs in order to collab- oratively detect global attacks or to generate signatures of new attacks [13,24]. Some of the information inside an IDS alert includ- ing time-stamps, type of attack, and captured data is private. The privacy of these fields can be put at risk too. In these cases, more elaborate techniques are required to address privacy issues inside the IDS alerts.

3.3 IDS built-in data

IDS built-in data includes data such as attack signatures in misuse-based IDSs [36] which are used as a measure to detect intrusions. Normal profiles of users' behaviors in anomaly-based IDSs [37,38], which are learned by IDS during a training phase, are another example of IDS built-in data.

Considering the signatures in misuse-based IDSs, there are strong reasons to keep at least some signatures private. First of all, signatures can be analyzed by attackers who could potentially use them to learn their vulnerabilities and design their own exploits. This is especially the case for the signatures of zero-day exploits. IPS/IDS devices can become an accessible source of information for attackers who wish to learn and design new attacks.The following quote from the Zero Day Initiative (ZDI) [39] dis- closure policy demonstrates that security experts are well aware of the sensitivity of such information:''TippingPoint may share technical details of the vulnerability with other security vendors. Such a security vendor must show they are able to provide security protection for vulnerabilities, while at the same time not revealing the technical vulnerability details in their product updates.''Many security vendors consider their protection mechanisms proprietary and do not want to make them accessible to their competitors. However, ensuring that product updates do not reveal vulnerabilities and also hide the information considered propri- etary by the vendors can be a challenging task in the traditional setting.With the day-to-day discovery of new vulnerabilities, IDS ven- dors use a considerable amount of resources to generate new and effective signatures to patch new vulnerabilities. Therefore, many IDS vendors are not willing to disclose their new signatures to others. There are few research prototypes which address the problem of preserving the privacy of IDS built-in data [20,28,30] and many aspects of it have not yet been studied. We discuss a number of future directions for research in this area in Section 6.

4. A SURVEY AND COMPARISON

We survey and compare related works that address privacy issues in IDSs based on our proposed taxonomy and also present the pros and cons of each work. Table 1 lists the reviewed papers as well as their characteristics. This survey can help researchers fol- low the research trends in this field and guide them to open areas of research. For each work, the following aspects are specified:

- ***Sources of sensitive data:*** This column specifies the source of sensitive data for the corresponding work. I.I.D. stands for ''IDS Input Data'', I.B.D. stands for ''IDS Built-in Data'' and I.G.D. stands for ''IDS Generated Data''.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

- ***Privacy-sensitive field:*** This column specifies which fields of the data source are privacy-sensitive in the corresponding work.

- ***Privacy technique:*** This column specifies the privacy tech- nique used in the paper. A review and comparison of the privacy techniques is discussed in Section 5.

4.1 Works on privacy issues in IDS input data

The first works in this category focus on providing privacy for identifier fields. The ideas for providing privacy in this category was first proposed by Sobirey et al. [5] and it was later extended and completed by others. The main privacy technique is to re- place identifier fields with pseudonyms. In [5], the kernel of the Operating System is modified in order to replace user-IDs with pseudonyms during log generation. By this method, all log files are pseudonymized before being available to the IDS. The drawback with this method is that the kernel of the system should be totally trusted and in case of a system compromise or a backdoored kernel, the identity of users is compromised. To fix this, Pseudonyms can be created by a trusted third party instead of the operating system's kernel. In [6], a Kerberos-like protocol is proposed to address the anonymity issues.

When the IDS detects an intrusion, a re-identification process is used to reveal the real identity of the attacker to the system security administrator. In the early works, disclosure occurs imme- diately after recognizing an event as an attack, while in newer ones it occurs only if the number of alerts exceeds a threshold [8,14].

Sometimes there are privacy concerns over non-identifier fields in IDS input data. In these cases, privacy techniques other than pseudonyms have to be used. Various privacy techniques like hash- ing and homomorphic encryption are used to preserve the privacy of audit data – such as log files belonging to audit groups – which is sent to a central auditor for analysis [12].

Shamir's secret sharing scheme and time-shifting are used to preserve the privacy of Netflow logs [16]. However, the level of privacy provided by the proposed technique is not clear. Therefore, a distance-based timestamp comparison is suggested [40] in which a third party can compare time-stamps, but only if they are within a certain distance of a threshold. It is possible to compare events which are similar based on the time of their occurrence. The pro- posed method is only suitable for timestamps and is not applicable to other privacy-sensitve fields.

Park et al. proposed a privacy-preserving method in which homomorphic encryption is applied to encrypt and process log files [15]. However, this method is insecure as shown by Wagner et al. [41].

Niksefat et al. used a secure multi-party computation protocol to preserve the privacy of a suspicious payload extracted from a log file, or captured from a network [28]. The proposed system, called ZIDS, consists of both an IDS server that has a set of sen- sitive signatures for zero-day attacks and IDS clients that possess sensitive input data. They address the privacy of both input data and IDS signatures. The proposed technique is promising since it is a general cryptographic-based technique for protecting any kind of input data. However, the proposed technique is computationally intensive because of its underlying cryptographic building blocks.

4.1.1 Analysis and discussion

Works that focus on the privacy of identifier fields in IDS input data mainly use the pseudonymization approach to hide identi- fiers. Recent works in this category focus on threshold-based re- identification, and a decentralized approach for pseudonym gen- eration. The drawbacks with almost all methods in this category is that pseudonymization cannot be used to provide privacy for non- identifier fields such as URLs. This is due to the fact that the actual content of non-identifier fields is needed for intrusion analysis and using pseudonyms would change the semantic of data.

On the other hand, work on non-identifier fields use techniques that are more elaborate than the pseudonym technique, and of course more computation intensive. Specifically, homomorphic encryption and secure multi-party computation (SMC) protocols techniques are promising since they can do computations on en- crypted data.

4.2 Works on privacy issues in IDS generated data

As we discussed earlier, preserving privacy in alerts is important when a number of IDSs are willing to share their alerts with the goal of detecting distributed attacks. A set of data sanitization techniques that enable alert aggregation and correlation while maintaining anonymity for alert contributors is proposed in [9]. Releasing the alerts with delay is suggested as a way to defend against probe-response attacks while it fails in real-time

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

detecting. In another approach, several IDSs collaborate in order to catch distributed attacks by finding the intersection of their suspected IP addresses, called watch-lists [10,17]. Here the privacy challenge is that the organizations are not willing to expose their suspected IP watch-list to others. They use bloom filters, as explained in 5.3, to address the problem of secure intersection. The issue is that Bloom filter will get saturated dealing with huge amount of data and the accuracy is very low due to lots of false positives.

Secure set intersection problem was later extended in [19] by proposing a cryptographic protocol that allows participants to submit a set of IP addresses that are suspected of engaging in un- wanted activities, and then returns the set of IP addresses found in more of the suspected sets than allowed by a given threshold. This approach is exposed to communication and computation overhead of cryptographic protocols.

A concept-hierarchy-based approach is presented in [11] to generalize IP addresses to subnet masks. For example, an IP address like 192.168.1.5 is generalized by replacing it with a network address such as 192.168.1.0/24 and then it is shared with others. An entropy guided method is used to strike a balance between accuracy and privacy. However, alert correlation graph over gen- eralized identities is prone false positive.

Ulltveit et al. proposed a two-tired architecture to preserve the anonymity of alerts in an outsourcing scenario [21]. In this work, sanitized alerts are outsourced to first line analyzers. In the case of a real intrusion, un-sanitized alerts are sent to the second line analyzers, a group of trusted experts with permission to violate privacy, for further analysis and detection of the intruder.
Parekh et al. proposed an approach in which several IDSs in- tended to generate attack signatures collaboratively by sharing their alert payloads [13]. Because payloads can contain organi- zations' private information, a statistical transformation method named Z-String is used to convert the payloads to an approximate character distribution. The Z-string is shared with other IDSs. Since Z-string function is a one-way function, reproducing the original payload from a Z-string is not straight-forward so, the privacy of the payloads is preserved. However, to detect polymorphic worms, which also needs payload analysis, more sophisticated models are required.

This approach was completed later by Kim [24] who presented two techniques for privacy-preserving distributed-signature gen- eration are presented. The first technique is a sampling technique using statistical methods and the second one is a multiset opera- tion framework which relies on a semantically-secure homomor- phic cryptosystem.
Burkhart et al. utilized the idea of parallel secure multi-party computation for alert correlation using Shamir secret sharing to improve the performance of existing multi-party computation

frameworks in [27]. There are two types of entities: input peers and privacy peers. The input peers are the distributed IDSs that send shares of their alerts to the privacy peers who then do the required computation. The main weakness is that privacy peers should be trusted and if any two of them collude, the system is broken.

A privacy-preserving similarity-based technique is used in [29] to detect similar alerts. The intuition is that similar alerts tend to share similar attributes. To protect privacy, alerts are encrypted under a homomorphic encryption before being sent to a central server for aggregation. The proposed approach is theoretical and its feasibility in real-world has not been evaluated. Also, encrypted data should be stored on the server for a long time that requires a huge storage capacity.

A collaborative system in which IDSs communicate via P2P protocol and share their alerts in order to find the similar alerts and then detect global attacks is proposed in [22]. Also, locality feature is used to determine how far an alert can be distributed among the involved IDSs, i.e., it can only be shared with the IDSs in the same domain with the source IDS. However, privacy is not considered among IDSs in the same domain.

4.2.1. Analysis and discussion

The privacy-sensitive data in this category can be either iden- tifier or non-identifier information. Approaches like bloom-filters have been proposed to protect identity information [10] while they are vulnerable to saturation and might produce many false positives. For non-identifiers, statistical approaches like Z-string and cryptographic techniques like homomorphic encryption are appropriate. Hence the techniques used are either computational intensive or error-prone as discussed in Section 5.

4.3 Works on privacy issues in IDS built-in data

Lundin et al. proposed an anomaly-based IDS that can analyze the pseudonymized audit data while preserving

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

the anonymity of normal profiles [7]. This is done by generating the profiles based on pseudonymized audit data. The proposed technique is limited to pseudonymization of usernames, hostnames and IP addresses only and no solution is given for other privacy-sensitive data. Moreover, despite using pseudonymization, information can be inferred from known data, which is an example of a classical database problem. Trousset et al. proposed a secure collaborative detection ap- proach called SAX (Secure Algorithm eXecution), which ensures that private data and programs will not be disclosed during intru- sion detection [20]. In this approach, any program from the various collaborative sites can be executed without disclosing any infor- mation from the local IDS to the outside. The signatures of the IDS as well as its input data are considered private. The goal is that the IDS can submit its data to other IDSs for analysis while preserving the privacy of both the input data and IDS signatures. This work has a number of restricting assumptions. One assumption is that it needs at least four parties to perform private computations: two distinct and non-colluding agents, a control site and a processing site. Clearly this assumption limits the application of this approach in the absence of the trust. To address this issue, Niksefat et al. proposed a privacy-preserving signature-based IDS that converts signatures to a Deterministic Finite Automata (DFA) [28]. Then, private payloads are evaluated against the DFA through a special secure two-party computation protocol in such a way that the IDS client learns nothing about the zero-day signatures and the IDS server learns nothing about the input data and the analysis results. However, as we discussed in previous section, the proposed technique is computation intensive because the underlying cryp- tographic building blocks impose relatively heavy computational cost.

Table 2
Privacy techniques and comparison (F.P/F.N: False Positive/False Negative—Comp.: Computational).

Technique	F.P./F.N.?	Comp. overhead
Pseudonyms	No	Low
Hash functions	No	Low
Bloom filters	Yes	Medium
Homomorphic Enc.	No	High
S.M.C protocols	No	High
Z-strings	Yes	Medium
Concept hierarchies	Yes	Low
Differential privacy	Depends.. .	Depends.. .

4.3.1 Analysis and discussion

The type of data that is privacy-sensitive in this category is es- sentially different than the two other categories. Here the method focuses on providing the built-in data inside IDSes such as sig- nature and normal profiles, which makes the techniques more complicated and computational intensive.

5. PRIVACY-PRESERVING TECHNIQUES IN IDSS

Various techniques can be used to address the privacy re-quirements in intrusion detection systems. Each technique has its own advantages and drawbacks. For example, using some privacy- preserving techniques may increase IDS' false-positive and false- negative rates or others may affect the IDS' performance. In this section, we review each of these techniques, their applications for providing privacy and their effects on IDS' precision and perfor-mance are discussed. Table 2 lists and compares various techniques used to address the privacy requirements in intrusion detection systems.

5.1 Pseudonyms

The pseudonyms technique is one of the most common tech- niques to preserve privacy in IDS. This technique involves identi- fying features in log files or audit data and replacing them with pseudonyms before being sent to the IDS, and therefore the IDS does not have access to users' real identities during intrusion analysis. Various techniques have been proposed for pseudonym generation such as simple one-to-one string mapping [5,7], hash- ing [9] and Shamir secret sharing [14].

In some cases, pseudonyms are generated by the Operating System [5,8]. Moreover, a Kerberos like protocol, in which a trusted third party authenticates users and issues signed authentication tickets, was proposed in [6]. Each pseudonymized authenticated ticket can then be used by its associated user to log in to net- work services. The tickets contain information about the users' pseudonyms, but not their actual user-IDs.

Re-identification is the process of identifying malicious users given their pseudonyms [14]. The component that generates the pseudonyms might be responsible for re-identifying them if re- quested by the security manager [5–7]. In another approach, called ''Technical Purpose of Binding'', the user's identity can be auto- matically discovered by the IDS when the number of misuses by a particular user exceeds some threshold [8,14]. This

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

method has the advantage of a faster intrusion response time.

5.1.1 Scope of use

Although the pseudonym technique is rather simple and easy- to-use, it is suitable only for providing anonymity in IDS and it is not suitable for providing privacy of non-identity fields such as IDS signatures, URLs or time-stamps.

5.1.2 Effects on IDS precision and performance

Pseudonyms generally do not have a notable effect on IDS' precision because an IDS does not need to know the exact value of identity fields in order to detect an intrusion. For example, the value of the user-ID field is not important in the detection phase and the IDS can detect malicious requests using the URL field. Therefore, when the user-ID field is replaced with a pseudonym, the IDS can still detect the attacks.

In attack scenarios in which a series of actions are performed by the attacker, user pseudonyms must remain permanent in order to correctly detect the intrusion. In these cases, the use of a threshold scheme like [8] for pseudonym generation increases the likelihood of false-positives and false-negatives. We refer the readers to [14] for a detailed explanation on pseudonym effects on the IDS false- positive and false-negative rates.

IDS have some computation overhead when using pseudonyms due to the generation and substitution of pseudonyms in au- dit data. Moreover, when an attack occurs, the re-identification process may affect the system's performance. The experimental results in [14] show that this overhead is negligible, and using pseudonyms to provide anonymity in IDS is quite practical.

5.2 Hash functions

A hash function is a mathematical algorithm that maps data of arbitrary size to a bit string of a fixed size, which is designed to be a one-way function which means it is not invertible.

5.2.1 Scope of use

Hash functions can be used as an obfuscation method (e.g. in pseudonym generation) but can also be used to obliviously test the equivalence of two private values. In that case, the hashes of private values are exchanged and compared. If the hashed values are equal, one can conclude that the original values are equal as well. In a number of works [12,16] hash functions are used to provide privacy in IDS.

5.2.2 Effects on IDS precision and performance

While hashing is not a powerful privacy-preserving technique which can only be used in limited cases, hashing is considerably faster than other cryptographic techniques and therefore they have less of an effect on IDS' performance. Readers are referred to Sections 5.2.1 and 6 of [17] for more detailed information on the applications of hash functions in IDS.

5.3 Bloom filters

A Bloom filter is a one-way data structure that supports two operations: insertion and verification. This data structure is built in such a way that no information can be extracted from the data inserted into them, while it is possible to verify if a specific data has been inserted [10].

5.3.1 Scope of use

Application of Bloom filters is limited to insert/verify problems such as secure set intersection. Since Bloom filters are inherently similar to hash functions, they are efficient and helpful tools to address some of the privacy issues in IDS.

5.3.2 Effects on IDS precision and performance

Since the primitive building function in a bloom filter is a hash function, they are quite a fast technique. However, when the bloom filter is saturated, it returns false positives (because, for example, multiple data entries resolve to the same locations in the bit vector), but never gives false negatives. The false positives can be avoided through tuning and correlating against multiple alert lists [17].

5.4 Homomorphic encryption

Homomorphic encryption schemes are encryption functions that allow encrypted data to be operated on without knowledge of the decryption key introduced by Rivest et al. [42]. The claim is that these schemes can be applied to protect databases against eavesdropping by the system manager.

In 2009, Gentry introduced the first Fully Homomorphic En- cryption (FHE) scheme that allows any arbitrary function to be evaluated in an oblivious manner [43]. Before that, only a limited set of operations could be performed on encrypted data.

5.4.1 Scope of use

A limited number of works applied partially homomorphic schemes in order to address the IDS-privacy issues [12,15,44]. In [12], some misuse counters are added homomorphically and in [15] a small number of

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

arithmetic operations are performed on encrypted data to discover abnormal login hours. The goal is to allow the IDS to process privacy-sensitive data without being able to learn them. To achieve this goal, a homomorphic encryption function is used to encrypt privacy-sensitive data before sending the data to the IDS.

5.4.2 Effects on IDS precision and performance
Although homomorphic encryption is a powerful tool for pre- serving privacy, these schemes require computationally intensive algorithms. Existing schemes do not seem to be practical in time- sensitive applications, like those related to IDS.

5.5 Secure multi-party computation protocols
Certain cryptographic protocols which preserve the privacy of data in a distributed computation. Secure Multi-party Computa- tion (SMC) is the most popular scheme in this field [45]. In general, it points to computational systems in which multiple parties wish to jointly compute some value based on individually-held secret information, but do not wish to reveal their secrets to one another in the process.

5.5.1 Scope of use
Cryptographic protocols and specifically SMC protocols are a powerful technique which can preserve the privacy of non- identifiers (URLS, any arbitrary data) fields. There are few works that use cryptographic protocols with the aim of preserving privacy in IDS. Kerschbaum et al. proposed a protocol to detect anomaly patterns in RFID tags [23] and Ringberg et al. proposed a proto- col for threshold secure set intersection [19]. A secure two-party computation protocol for private detection of intrusions has been proposed in [28] that is suitable for private detection of zero-day attacks in private input data.

5.5.2 Effects on IDS precision and performance
Cryptographic protocols are generally designed in a way that satisfies strict security requirements and hence are generally ex- pensive in terms of computation and communication. Recently many efforts have been directed toward proposing more efficient protocols that are suitable for real-life applications.

5.6. Z-string
Z-string is a list of feature vectors from a histogram which shows the Zipf distribution of a given frequency distribution. The frequencies are arranged from highest to lowest and the cor- responding feature list is kept while the actual frequencies are thrown away.

5.6.1 Scope of use
Z-Strings can be used in collaborative IDSs to convert the sus- picious payloads to an approximate character distribution. The Z- string is then shared between collaborating IDS's, and since gen- eration of the original payload from a Z-string is impossible, the privacy of the payloads is preserved.

5.6.2 Effects on IDS precision and performance
Z-string is an efficient and fast technique but it may affect the IDS' precision as explained in Section 6.5.3 of [17].

5.7 Concept hierarchy
In the concept hierarchy technique, original attribute values are generalized to high-level concepts. For example, IP addresses are generalized to network addresses, and continuous attributes are generalized to intervals.

5.7.1 Scope of use
This technique can be used to protect the privacy of any data fields such as IP addresses or time-stamps that can be generalized.

5.7.2 Effects on IDS precision and performance
Generalization is a non-heavy computation task and can be done very efficiently. However, generalization always results in some sort of data loss which may affect the IDS precision and increase the IDS false positive/negative rates.

5.8 Differential privacy
Differential privacy is a rather new privacy technique that max- imizes the accuracy of queries from statistical databases while minimizing the chances of identifying its records. In other words, differential privacy is a framework for formalizing privacy in statis- tical databases in order to protect against deanonymization tech- niques [46].

5.8.1 Scope of use
Differential privacy is a promising approach for preserving pri- vacy in collaborative IDSs or an IDS with multiple sensors. Each IDS/sensor input data or alerts are seen as individual records of a statistical database and the goal is to perform a statistical query (e.g. whether an attack is in progress or not) on the whole database without leaking information about individual records. There are

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

few work that used this technique to preserve privacy in intrusion detection systems. [47].

5.8.1 Effects on IDS precision and performance

The effects of using the differential privacy technique on pre- cision and performance of IDSs is highly dependent on the un- derlying mechanisms. For example if noise is used to perturb the results, although it is quite fast, however it may increase the false positive/negative rates. If cryptographic building block are used, then the results are more accurate, however the calculations are less efficient.

5.9. Other classical techniques

There are numerous classical privacy-preserving techniques, but most of them can only be used for very simple privacy prob- lems [48]. We review several of these techniques: Blocking (Re- moval), Perturbation and Shifting.

Blocking removes sensitive data before they are released. Data removal and data reduction are interchangeably used for blocking. This technique is limited to data that are not required in the in- trusion detection process [6]. For example, removing the URL field

in an IDS input data would disrupt intrusion analysis. Although blocking is a simple technique for privacy protection, it affects the precision.

Perturbation is a technique that adds noise to privacy-sensitive data. In privacy-preserving data mining, sensitive data are per- turbed by adding noise to a statistical distribution [48]. This tech- nique works well when a large data set is provided and aggregated results such as summation or average are needed. This technique is not feasible in intrusion detection scenarios, like signature match- ing, in which the contents of each pattern for exact matching is needed, because it decreases the IDS' precision.

Shifting hides a value by shifting its bits. For example, all time- stamps in a log file can be shifted by a fixed random number. This way, the relative time intervals between events remains intact but the exact values of the time-stamps are hidden. The relative time information is enough to determine a number of attacks when correlating alerts from different sources [16].

6. FUTURE DIRECTIONS

In this section, we discuss a number of areas in which further research is needed and propose a number of interesting open problems in the field of IDS-privacy issues.

6.1 Quantifying privacy preservation in IDS

As discussed in Section 5, using privacy techniques to address the privacy issues in IDS may affect the performance and precision of the IDS. For example, cryptographic methods are very precise and powerful methods for providing privacy, but these schemes usually affect the IDS' performance since they require significant amounts of computation and communication. On the other hand, other techniques such as perturbation techniques have a good performance but are not as precise and powerful as needed. So it seems that the main challenge in providing privacy in IDS is to find a balance between the three conflicting objectives of performance, precision and privacy. In this regard, the issues of IDS' precision and performance and their relationship should be well-studied. IDS' precision is generally calculated based on IDS false-negative and false-positive rates. ROC curves [49] specifically help us in this matter by representing the balance between false-positive and false-negative rates. Parameters like response time or detection time are good measures for quantifying IDS' performance.

In order to quantify IDS-privacy preservation capabilities, one can use the measure of the level of privacy that an IDS provides. This can be achieved based on our proposed taxonomy given in Fig. 1. Quantification of privacy is important because it allows us to compare various IDS systems in terms of privacy preservation. Also it allows us to compare the power of various privacy techniques quantitatively.

Quantification requires an attacker model and can be measured depending on the application scenario. Namely, in case the goal is to protect the user privacy by pseudonyms, the attacker model would be to disclose user IDs. Therefore, the percentage of dis- closed user-IDs could be a metric for quantifying the privacy leak.

6.2 Providing privacy for non-identity data

Although there are numerous works which address the problem of preserving anonymity in IDS (i.e. replacing identity fields with pseudonyms), there are few works which address the privacy issues for non-identity fields such as content of files, or even the IDS signatures. To

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

provide privacy for such data, powerful cryp- tographic techniques for performing computations on encrypted data seem to be promising.

These techniques (and their applications in IDS-privacy issues) have not yet been well-studied. Unstudied areas include: Secure Multi-Party Computation (SMC) [45] and homomorphic encryp- tion [43]. These methods can provide strong privacy, but more efficient applications of these techniques need to be developed before they can be practical to be applied to these situations.

6.2.1 Fully homomorphic encryption (FHE)

Fully Homomorphic Encryption (FHE) allows any arbitrary pro- gram to be run on encrypted version of their inputs to produce an encrypted output [43]. Since the program never decrypts its input, it can be run by an untrusted party without revealing its inputs and internal state. This property makes FHE one of the most powerful techniques for providing privacy in computations. In the field of IDS-privacy issues, FHE schemes can provide privacy for both IDS and users.

Considering the privacy issues when intrusion analysis is out- sourced to an untrusted party, FHE schemes seem to be useful. Here, the IDS client encrypts the data using a FHE scheme and sends it to the IDS server, and the IDS server analyzes the data without being able to read it. The IDS server calculates the encrypted results which are then sent back to the IDS client for decryption. The main challenge with current schemes is that they require a significant amount of computations which can degrade IDS' performance. Further studies in this field can lead to schemes which establish a balance between security and efficiency.

We propose an open problem that can be solved using FHE schemes. In fact the following problem can be seen as outsourcing of a specific kind of computation (intrusion analysis) which is a common application of FHE schemes.

Open Problem 1 (Privacy Preserving Signature-Based Intrusion De- tection System). Suppose that Alice has a signature-based IDS (e.g. Snort) with a list of signatures for various computer attacks. Assume Bob has some data (e.g. a log file) that needs to be scanned for possible attacks using Alice's IDS. On one hand Bob does not trust Alice and wishes to preserve the privacy of his data from Alice. On the other hand, Alice's signatures are private and cannot be sent to Bob. How can Bob use Alice's IDS while neither of them disclose their private data to the other party?

Solving the above problem may solve situations where signature-based IDSs are not fully trusted. One case is when an organization needs to outsource the security analysis of some confidential data to a third-party company. In this case, because of confidentiality of data, the organization has privacy concerns because confidentiality of the data may be compromised during outsourcing. Another important reason for preserving privacy in signature-based IDSes is using third-party IDS products. Many organizations such as military bureaus which have secret informa- tion cannot trust commercial IDS products because of the potential of existence of backdoors in these products. Having an effective solution for this problem can help in building signature-based IDSs which are able to detect intrusions without needing to access data directly.

6.2.2 Using secure multi-party computation (SMC)

There are some cryptographic protocols which preserve the privacy of data in a distributed computation. Secure Multi-party Computations (SMC) are the most popular scheme in this field [45]. In general, it points to computational systems in which multiple parties wish to jointly compute some value based on individually- held secret bits of information, but do not wish to reveal their secrets to one another in the process. Using SMC schemes for addressing privacy issues seems po- tentially useful. However, current general SMC solutions that are able to compute any function securely have high computation

overhead. Therefore, SMC is not yet an efficient solution where performance is an important parameter. Efficient SMC schemes that are applicable to IDSs are one of the areas that have been studied in [28] but more studies are needed in order to build more applied systems.

The following open problem which deals with the privacy issue of distributed IDSs can be addressed using SMC schemes. Like the SMC model in the following scenarios a computation is performed using the secret input of each party.

Open Problem 2 (Privacy Preserving Distributed Signature-Based Intrusion Detection System). Suppose a number of signature-based IDSs, each of them has separate and private attack signatures. When data is sent to these IDSs for attack detection, they check it against their local signatures and also check this data by other IDSs signatures as well, in order to enhance their local

intrusion detection rate. How can these IDSs cooperate in intrusion detection while preserving the privacy of their local data, since the local data and local signatures of each IDS are private?

Problem 2 can be seen as a more general case of Problem 1 but with additional privacy requirements. It is a common scenario in today's anti-cybercrime strategies to cooperate with others. Using other IDS signatures may help to better detect novel attacks and reduce the false negative rate of local intrusion detection. But privacy concerns are a blockade for organizations to share their data. A solution to this problem may help organizations to cooperate for better and more accurate intrusion detection without compromising their valuable data.

CONCLUSION

We discussed the important issue of preserving privacy in IDS. We explained that in some intrusion detection scenarios it is required that identities or other privacy-sensitive data not be disclosed to parties involved in the intrusion detection process. We then identified and classified the sources of privacy-sensitive data in IDS as ''IDS input data'', ''IDS built-in data'' and ''IDS generated data''. Based on this classification, we proposed a taxonomy of privacy requirements in IDS. For each data source in the proposed taxonomy, we discussed a number of privacy-sensitive fields and their respective techniques.

Based on our proposed taxonomy, we then surveyed and com- pared a number of important research works which address the privacy issues of IDS. We also surveyed the privacy techniques used to provide privacy in IDS. For each privacy technique, we discussed its scope of use as well as its effect on IDS performance and precision.
Finally, we discussed a number of areas that require further research before they can be used to provide efficient and secure privacy for IDSs.

REFERENCES

[1] Alan F. Westin, Privacy and freedom, Washington Lee Law Rev. 25 (1) (1968) 166.

[2] Health Information Privacy, http://www.hhs.gov/hipaa/ (last access: 06.09.17).

[3] Protection of Personal Data, http://ec.europa.eu/justice/data-protection/ (last access: 06.09.17).

[4] Justice Laws Website, http://laws-lois.justice.gc.ca/eng/acts/p-8.6/ (last ac- cess: 06.09.17).

[5] Michael Sobirey, Simone Fischer-Hubner, Kai Rannenberg, Pseudonymous Audit for Privacy Enhanced Intrusion Detection, Springer, 1997.

[6] Roland Büschkes, Dogan Kesdogan, Privacy enhanced intrusion detection, in: Multilateral Security in Communications, Information Security, 1999, pp. 187–204.

[7] Emilie Lundin, Erland Jonsson, Anomaly-based intrusion detection: privacy concerns and other problems, Comput. Netw. 34 (4) (2000) 623–640.

[8] Joachim Biskup, Ulrich Flegel, Transaction-based pseudonyms in audit data for privacy respecting intrusion detection, in: Recent Advances in Intrusion Detection, Springer, 2000, pp. 28–48.

[9] Patrick Lincoln, PhillipA. Porras, Vitaly Shmatikov, Privacy-preserving sharing and correlation of security alerts, in: USENIX Security Symposium, 2004, pp. 239–254.

[10] Michael E. Locasto, Janak J. Parekh, Angelos D. Keromytis, Salvatore J. Stolfo, Towards collaborative security and p2p intrusion detection, in: IEEE SMC Information Assurance Workshop, (IAW'05), IEEE, 2005, pp. 333–339.

[11] Dingbang Xu, Peng Ning, Privacy-preserving alert correlation: a concept hi- erarchy based approach, in: Computer Security Applications Conference, 21st Annual, IEEE, 2005, p. 10.

[12] Adam J. Lee, Parisa Tabriz, Nikita Borisov, A privacy-preserving interdomain audit framework, in: 5th ACM Workshop on Privacy in Electronic Society, ACM, 2006, pp. 99–108.

[13] Janak J. Parekh, Ke Wang, Salvatore J. Stolfo, Privacy-preserving payload-based correlation for accurate malicious traffic detection, in: Proceedings of the 2006 SIGCOMM Workshop on Large-Scale Attack Defense, ACM, 2006, pp. 99–106.

[14] Ulrich Flegel, Privacy-respecting Intrusion Detection, Vol.35, Springer Science & Business Media, 2007.

[15] Hyun-A. Park, Dong Hoon Lee, Jongin Lim, Sang Hyun Cho, PPIDS: privacy pre- serving intrusion detection system, in: Intelligence and Security Informatics, Springer, 2007, pp. 269–274.

[16] Jianqing Zhang, Nikita Borisov, William Yurcik, Outsourcing security analysis with anonymized logs, in: Securecomm and Workshops, IEEE, 2006, pp. 1–9.

[17] Janak J. Parekh, Privacy-preserving distributed event corroboration. Ph.D. the- sis, Columbia University, 2007.

[18] Giuseppe Bianchi, Elisa Boschi, Dimitra I. Kaklamani, E.A. Koutsoloukas, Geor- gios V. Lioudakis, Francesco Oppedisano, Martin Petraschek, Fabio Ricciato, Carsten Schmoll, Towards privacy-preserving network monitoring: Issues and challenges, in: International Symposium on Personal, Indoor and Mobile Radio Communications, IEEE, 2007, pp. 1–5.

[19] Haakon Andreas Ringberg, Jennifer Rexford, Privacy-preserving collaborative anomaly detection. Ph.D. thesis, Princeton University, 2009.

[20] François Trousset, Pascal Poncelet, Florent Masseglia, SAX: a privacy pre- serving general purpose method applied to detection of intrusions, in: First International Workshop on Privacy and Anonymity for Very Large Databases, ACM, 2009, pp. 17–24.

[21] Nils Ulltveit-Moe, Vladimir Oleshchuk, Two tiered privacy enhanced intrusion detection system architecture, in: International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IEEE, 2009.

[22] Emmanouil Vasilomanolakis, Matthias Krügl, Carlos Garcia Cordero, Max Mühlhäuser, Mathias Fischer, SkipMon: A locality-aware collaborative in- trusion detection system, in: Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance, IEEE, 2015, pp. 1–8.

[23] Florian Kerschbaum, Nina Oertel, Privacy-preserving pattern matching for anomaly detection in RFID anti-counterfeiting, in: Radio Frequency Identifi- cation: Security and Privacy Issues, Springer, 2010, pp. 124–137.

[24] Hyang-Ah Kim, Privacy-preserving distributed, automated signature-based detection of new Internet worms. Ph.D. thesis, Carnegie Mellon University, 2010.

[25] Martin Burkhart, Mario Strasser, Dilip Many, Xenofontas Dimitropoulos, SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics, Network 1 (2010) 101101.

[26] Hayretdin Bahsi, Albert Levi, Preserving organizational privacy in intrusion detection log

sharing, in: International Conference on Cyber Conflict, IEEE, 2011, pp. 1–14.

[27] Martin Burkhart, Xenofontas Dimitropoulos, Privacy-preserving distributed network troubleshootingbridging the gap between theory and practice, ACM Trans. Inf. Syst. Secur. (TISSEC) 14 (4) (2011) 31.

[28] Salman Niksefat, Babak Sadeghiyan, Payman Mohassel, Saeed Sadeghian, Zids: A privacy-preserving intrusion detection system using secure two-party com- putation protocols, Comput. J. (2013) 1–16.

[29] Hoang Giang Do, Wee Keong Ng, Privacy-preserving approach for sharing and processing intrusion alert data, in: International Conference on Intelligent Sensors, Sensor Networks and Information Processing, (ISSNIP), IEEE, 2015, pp. 1–6.

[30] B. Michael Thomas, Neal L. Ziring, Using classified intelligence to defend un- classified networks, in: 2015 48th Hawaii International Conference on System Sciences, (HICSS), IEEE, 2015, pp. 2298–2307.

[31] Chenfeng Vincent Zhou, Christopher Leckie, Shanika Karunasekera, A survey of coordinated attacks and collaborative intrusion detection, Comput. Secur. 29 (1) (2010) 124–140.

[32] Emmanouil Vasilomanolakis, Shankar Karuppayah, Max Mühlhäuser, Mathias Fischer, Taxonomy and survey of collaborative intrusion detection, ACM Com- put. Surv. (CSUR) 47 (4) (2015) 55.

[33] Rainer Bye, Seyit Ahmet Camtepe, Sahin Albayrak, Collaborative intrusion detection framework: Characteristics, adversarial opportunities and counter- measures, in: International Conference on Collaborative Methods for Security and Privacy, CollSec, 2010.

[34] Vashek Matyas, Jiri Kur, Conflicts between intrusion detection and privacy mechanisms for wireless sensor networks, IEEE Secur. Privacy 5 (11) (2013) 73–76.

[35] Sudhir N. Dhage, B.B. Meshram, Intrusion detection system in cloud computing environment, Int. J. Cloud Comput. 1 (2–3) (2012) 261–282.

[36] Martin Roesch, et al., Snort: Lightweight intrusion detection for networks, LISA 99 (1) (1999) 229–238.

[37] Debra Anderson, Teresa F. Lunt, Harold Javitz, Ann Tamaru, Alfonso Valdes, et al. Detecting unusual program behavior using the statistical component of the Next-generation Intrusion Detection Expert System (NIDES), SRI International, Computer Science Laboratory, 1995.

[38] Parisa Kaghazgaran, Babak Sadeghiyan, Masquerade detection using GUI events in windows systems, Int. J. Inf. Commun. Technol. (IJICT) (2011).

[39] Tippingpoint, Zero day initiative. 2016. http://www.zerodayinitiative.com/ (Last Access: 06.09.17).

[40] Florian Kerschbaum, Distance-preserving pseudonymization for timestamps and spatial data, in: ACM Workshop on Privacy in Electronic Society, ACM, 2007, pp. 68–71.

[41] David Wagner, Cryptanalysis of an algebraic privacy homomorphism, in: Information Security, Springer, 2003, pp. 234–239.

[42] Ronald L. Rivest, Len Adleman, Michael L. Dertouzos, On data banks and privacy homomorphisms, Found. Secure Comput. 4 (11) (1978) 169–180.

[43] Craig Gentry, et al. Fully homomorphic encryption using ideal lattices, in: STOC, Vol. 9, 2009, pp. 169–178.

[44] Parisa Kaghazgaran, Babak Sadeghiyan, Secure two party comparison over en- crypted data, in: Information and Communication Technologies (WICT) World Congress, IEEE, 2011, pp. 1123–1126.

[45] Oded Goldreich, Foundations of Cryptography: Volume 2, Basic Applications, Cambridge University Press, 2004.

[46] Wikipedia. Differential privacy — Wikipedia, the free encyclopedia, 2017. h ttps://en.wikipedia.org/w/index.php?title=Differe ntial_privacy&oldid=77342 1488 [Online; accessed 28.05.17].

[47] Jason Reed, Adam J. Aviv, Daniel Wagner, Andreas Haeberlen, Benjamin C. Pierce, Jonathan M. Smith, Differential privacy for collaborative security, in: Proceedings of the Third European Workshop on System Security, ACM, 2010, pp. 1–7.

[48] Charu C. Aggarwal, S. Yu Philip, A general survey of privacy-preserving data mining models and algorithms, in: Privacy-Preserving Data Mining, Springer, 2008, pp. 11–52.

[49] Stefan Axelsson, The base-rate fallacy and the difficulty of intrusion detection, ACM Trans. Inf. Syst. Secur. (TISSEC) 3 (3) (2000) 186–205.