COMPUTERS & SECURITY XXX (2013) 1–15

Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Creditability-based weighted voting for reducing false positives and negatives in intrusion detection

Ying-Dar Lin [a], Yuan-Cheng Lai [b,*], Cheng-Yuan Ho [c], Wei-Hsuan Tai [a]

[a] Department of Computer Science, National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu 300, Taiwan
[b] Department of Information Management, National Taiwan University of Science and Technology, No. 43, Sec. 4, Keelung Road, Taipei 106, Taiwan
[c] Advanced Research Institute, Institute for Information Industry, 1F., No. 133, Sec. 4, Minsheng E. Rd., Taipei 105, Taiwan

ARTICLE INFO

Article history:
Received 18 December 2012
Received in revised form
9 September 2013
Accepted 29 September 2013

Keywords:
Intrusion detection
False positives/negatives
Weighted voting
Majority voting
Creditability

ABSTRACT

False positives (FPs) and false negatives (FNs) happen in every Intrusion Detection System (IDS). How often they occur is regarded as a measurement of the *accuracy* of the system. Frequent occurrences of FPs not only reduce the throughput of an IDS as FPs block the normal traffic and also degrade its trustworthiness. It is also difficult to eradicate all FNs from an IDS. One way to overcome the shortcomings of a *single* IDS is to employ *multiple* IDSs in its place and leverage the different capabilities and domain knowledge of these systems. Nonetheless, making a correct intrusion decision based on the outcomes of multiple IDSs has been a challenging task, as different IDSs may respond differently to the same packet trace. In this paper, we propose a method to reduce FPs and FNs by applying a *creditability-based weighted voting* (CWV) scheme to the outcomes of multiple IDSs. First, the CWV scheme evaluates the *creditability* of each individual IDS by monitoring its response to a large collection of pre-recorded packet traces containing various types of intrusions. For each IDS, our scheme then assigns different *weights* to each intrusion type according to its FP and FN ratios. Later, after their operations, the outcomes of individual IDSs are merged using a *weighted voting* scheme. In benchmarking tests, our CWV-based multiple IDSs demonstrated significant improvement in *accuracy* and *efficiency* when compared with multiple IDSs employing an ordinary majority voting (MV) scheme. The accuracy is the percentage of whole traces that are determined accurately, while the efficiency indicates that the voting algorithm performs better on reducing both FP and FN ratios. The CWV scheme achieved 95% accuracy and 94% efficiency while the MV scheme produced only 66% accuracy and 41% efficiency; the average percentages of FP/FN reduction were 21% and 58% respectively.

* *Corresponding author*. Tel.: +886 2 27376794; fax: +886 2 27376777.
E-mail addresses: ydlin@cs.nctu.edu.tw (Y.-D. Lin), laiyc@cs.ntust.edu.tw (Y.-C. Lai), tommyho@iii.org.tw (C.-Y. Ho), weihsuantai@gmail.com (W.-H. Tai).

0167-4048/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cose.2013.09.010

1. Introduction

Intrusion Detection Systems (IDSs) protect computer networks against intrusions. Among various techniques, the signature-based approach has been a popular one. It characterizes intrusions according to their specific attack patterns, known as signatures, and tries to detect malicious activities by comparing these signatures against the actual traffic using *pattern matching*. For a signature-based IDS to operate correctly, it must maintain a signature database of all known intrusions. There are some major challenges in implementing an effective signature-based defense mechanism. The first one lies with the balance between generic and precise signature specification. Generic signatures can be used to specify wide ranges of attacks, but they can be mismatched with benign traffic and thus produce *false positives* (FPs). On the other hand, signatures that are too specific may miss attacks that differ only slightly from some known attacks, and produce *false negatives* (FNs). The second challenge lies with the difficulty of implementing an efficient IDS runtime engine. In order to maintain a reasonable throughput, an IDS engine cannot afford to analyze the complete context of all network activities in real-time. As a result, intrusions that differ only slightly from normal activities or known attacks may not be detected because the IDS engine only analyzes the partial context of potential intrusions. Lastly, the on-going variation of malicious traffic makes it difficult to maintain an up-to-date signature database. A well-known example is the maintenance of the Snort signature database (Sourcefire).

As figures of merit, the frequencies of FP and FN occurrences are often quite high among IDSs and render most of them unsatisfactory in their performance. To illustrate the severity of FP and FN impacts, let us examine the issues from two different angles. From the vendor's viewpoint, FPs create a heavy workload for the IDS analysis engine, while FNs cause the IDS engine to fail in generating the signatures of new attacks. From the user's viewpoint, frequent alerts generated by FPs disrupt the system administrators' work and cause them to distrust the system, while FNs imply that malicious intrusions have not been detected in a protected network. All these are serious issues. Thus, the reduction of FP/FN occurrences remains a paramount task for all IDS vendors and users.

In order to reduce FPs and FNs, an analyst *manually* post-processes all the alerts produced by the IDS to determine whether the alerts are *true positives* (TPs) or *false positives* (FPs) (Pietraszek, 2006). Nevertheless, an IDS can do only so much by itself, partly because the analyst can only examine the TP/FP alerts produced by the IDS, but cannot investigate the FNs of the IDS. In addition, if there are many FPs, an analyst would need considerable time to analyze the alerts. According to alert management (Pietraszek, 2006), which means an analyst post-processes all the alerts to further improve the signature design, the use of a single IDS as the only source of alerts would be a major handicap. To overcome the problem and limitations of a single IDS, multiple IDSs are used because each has its own private and independent signature design. Based on the different domain knowledge among the IDSs, traffic can be more accurately recognized by leveraging the detection capability of the IDSs. Therefore, malicious activities which cannot be detected by some IDSs are detected by the others.

However, the detection results among multiple IDSs may be in conflict. To resolve such conflicts, the *Majority Voting* (MV) algorithm (Chen et al., June 2009) was proposed. MV first finds *potential* FPs (P-FPs) and *potential* FNs (P-FNs) by comparing the alerts. If a few IDSs generate alerts but most IDSs do not when they process the same traffic, these traces are P-FPs of the few IDSs. In contrast, if few IDSs do not generate alerts but most IDSs do, these are P-FNs of the few IDSs. Then, P-FPs and P-FNs are analyzed to verify they are indeed FPs and FNs. However, in (Latif-shabgahi et al., 2004; Parham, 2002), the authors found MV often leads to incorrect decisions, although they did not focus on the results of alert handling. We also found MV is not *efficient* in our experiments: MV disregards the different domain knowledge among IDSs, which results in *low* percentages of P-FPs/P-FNs being *true* FPs/FNs.

In this work, we propose a *Creditability-based Weighted Voting* (CWV) scheme to leverage the different domain knowledge among multiple IDSs, reduce FPs and FNs, and increase the efficiency of alert post-processing. There are four components in our algorithm: *Creditability Modeling* (CM), *Authority Selection* (AS), *Voter Exclusion* (VE), and *Weighted Voting* (WV). First, CM identifies the IDSs' detection capability for different types of traffic traces and determines the IDSs' corresponding creditability by investigating their *past* detection experience. In order to investigate the detection capability based on one or both of the two factors composing an alert, i.e., protocol type and malicious type, the creditability is therefore constructed at two levels, *Protocol* level and *Alert* level. For instance, "HTTP" is at the Protocol level. "HTTP: Attempt to Read Password File" is an alert of HTTP at the Alert level. If the creditability of some IDSs exceeds decision criteria for a certain type of trace, AS will choose them to be *authorities*. The reason is that these IDSs may have lower FP and FN incidences for a certain type of trace. On the other hand, if no IDS can be an authority, VE will exclude IDSs that perform poorly in detection because these IDSs would nullify the correct decisions. Finally, WV assigns *weights* to either the IDSs chosen by AS or the existing IDSs excluded by VE and uses these weights to determine a trace as malicious or benign. Accordingly, compared with MV, CWV considers the different domain knowledge among IDSs, and thus reduces FPs and FNs.

This work makes the following contributions. First, it uses multiple IDSs simultaneously, rather than a single IDS, to detect intrusions. Second, it uses the CWV scheme to leverage domain knowledge among multiple IDSs, reducing not only the number of FPs, but also the number of FNs. CWV determines the creditability of each IDS at the protocol level and at the alert level, rather than at the single level that is used in MV. Third, several experiments are performed to demonstrate that CWV is much better than MV on accuracy and efficiency.

The rest of this paper is organized as follows. Section 2 presents the background and related works. Section 3 defines terminologies and problem statements. Section 4 describes the design and solution ideas of our algorithm. Section

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

5 displays the evaluation of our solutions. Finally, Section 6 concludes this work and discusses the future work.

2. Background

This section describes alert post-processing and its related methods, followed by the generation FP/FN datasets.

2.1. *Methods of alert post-processing*

Currently, IDSs have become necessary components to network security. However, IDSs easily produce many alerts, so it is necessary to find an efficient way to reduce the number of alerts and provide a more succinct and high-level view of security events. To address this issue, alert post-processing (APP) has been proposed. APP uses alerts as an input and processes them to improve their accuracy. According to the corresponding goal, APP can be classified into three categories: alert correlation, alert causing, and alert classification. Note that some previous approaches may belong to multiple categories because they have more than one goal. A systematic illustration of these categories is as follows.

First, alert correlation finds the causal relationships between alerts in order to construct high-level attack scenarios from the isolated alerts (Valeur et al., 2004; Xu et al., 2008; Ning et al., 2002; Valdes and Skinner, 2001; Maggi et al., 2009; Yu and Frincku, 2005; Porras et al., 2002; Sadoddin and Ghorbani, 2006; Ning and Xu, 2003; Ning et al., 2004). There are three types of alert correlation techniques: multi-step, fusion-based, and filter-based (Xu et al., 2008). For example, Ning et al. (2002) detected multi-step attacks with an alert correlation approach which correlates alerts based on pre-conditions and post-conditions. Two alerts are correlated when the pre-condition of a later attack is satisfied by the post-condition of an earlier attack. The fusion-based alert correlation technology utilizes the alert similarity metric to fuse the correlative alerts. For example, Valdes and Skinner (2001) presented a correlation process utilizing an alert similarity metric. Maggi et al. (2009) used fuzzy measures and fuzzy sets to deign an alert fusion model. Yu and Frincku (2005) proposed improving and assessing alert accuracy by incorporating an algorithm based on the exponentially weighted Dempster-Shafer theory of evidence. The filter-based correlation technology seeks to identify the most important alerts in the alert stream. For example, Porras et al. (2002) discussed a mission-impact-based approach to prioritize alerts. Alert correlation actually offers a more high-level view on the security events raised by the IDS. However, Sadoddin and Ghorbani (2006) found that alert correlation may not have a significant effect on reducing the number of total alerts and the number of FPs. The reason is that the goal of alert correlation is to provide an abstract view of attacks, rather than reducing the number of FPs and FNs, despite that it sometimes does reduce the number of FPs.

Second, alert causing (Julisch, 2003a, 2001, 2003b) studies the causes of FPs and identifies root causes that create an IDS alert. It groups the alerts with similar root causes. For instance, Julisch (2003a) defined six attributes for an alert: source and destination IP addresses, source and destination ports, alert types, and timestamps. The alerts with the same six attributes are categorized into the same group, called an *alert cluster*. Thus, the alerts in the same alert group may have the same root cause. According to the root causes, a system administrator may reduce the number of FPs in an IDS.

Third, alert classification classifies alerts into TPs and FPs for reducing the number of FPs in IDSs (Chen et al., June 2009; Pietraszek, 2004; Viinikka et al., 2009; Treinen and Thurimella, 2006; Clifton and Gengo, 2000; Long et al., 2006; Vaarandi, 2009; Vaarandi and Podins, 2010; Zhang et al., 2012; Gupta et al., 2012). The Adaptive Learner for Alert Classification (ALAC) is an adaptive alert classifier based on the feedback of an intrusion detection analysis and a machine-learning technique (Pietraszek, 2004). It has a recommender mode and an agent mode. In the recommender mode, all the alerts are labeled as TPs or FPs and passed to the analyst; in the agent mode, some alerts are processed automatically. Therefore, ALAC could intuitively reduce the number of FPs in the IDS. However, although the agent mode reduces the analyst's workload, the recommender mode still creates a heavy workload for the analyst. Viinikka et al. (2009) used time series modeling for modeling regularities in large alert volumes. This study is based on the observation that flows consisting of alerts related to normal system behavior can contain strong regularities. Thus it models these regularities using non-stationary autoregressive models. Once modeled, the regularities can filtered out to reduce the number of FPs.

To reduce the number of FPs in IDSs, some approaches with alert log mining have been proposed (Treinen and Thurimella, 2006; Clifton and Gengo, 2000; Long et al., 2006; Vaarandi, 2009; Vaarandi and Podins, 2010; Zhang et al., 2012; Gupta et al., 2012). Treinen and Thurimella (2006) have investigated the application of association rule mining for the detection of rules for new attack types. Using a similar approach, frequent alert sequences were investigated to construct IDS alert filters in (Clifton and Gengo, 2000). Long et al. (2006) have suggested a supervised clustering algorithm for distinguishing Snort IDS true alerts from FPs. An unsupervised data mining based approach for IDS alert classification was further proposed in (Vaarandi, 2009). This algorithm first employs frequent itemset mining to detect patterns that describe frequently occurring redundant alerts. It then extracts signature IDs from detected patterns and finds frequent endpoint sets which describe strong associations between alert attribute values for each ID. With this approach, knowledge is mined from IDS logs and processed in an automated way to build an alert classifier. However, frequent endpoint sets used in (Vaarandi, 2009) cannot capture all strong associations because an unrealistic assumption that all associations for a given signature ID are of the same type was made. Thus, Vaarandi and Podins (2010) further proposed data clustering techniques to find fine-grained subpatterns for each detected pattern. Decision support classification (DSC) was proposed for alert classification (Zhang et al., 2012). DSC first collects alert transactions in an attack-free environment. Accordingly, all alerts are treated as FPs in this environment, and the frequent patterns to be mined in this case are treated as patterns of normal behaviors, or called patterns of FPs. DSC then uses these patterns to remove FPs. Gupta et al. (Gupta et al., 2012) proposed a post-processor for IDS alerts using knowledge-based evaluation, a system that uses background

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

Table 1 – Comparison of methods of alert classification.

Reference	Techniques	No. of IDSs	Goal	Drawbacks
(Pietraszek, 2004)	Machine-learning	One	Reduce FPs	High overheads in recommender mode Manual knowledge acquisition
(Viinikka et al., 2009)	Time series modeling	One	Reduce FPs	High overheads
(Treinen and Thurimella, 2006)	Association rule mining	Multiple	Reduce FPs	High overheads
(Clifton and Gengo, 2000)	Frequent alert sequences finding	One	Reduce FPs	High overheads
(Long et al., 2006)	Supervised clustering	One	Reduce FPs	High overheads
(Vaarandi, 2009)	Unsupervised data mining	One	Reduce FPs	High overheads
(Vaarandi and Podins, 2010)	Unsupervised data mining and data clustering	One	Reduce FPs	High overheads
(Zhang et al., 2012)	Decision support	Multiple	Reduce FPs	High overheads
(Gupta et al., 2012)	Knowledge-based evaluation	One	Reduce FPs	High overheads
(Chen et al., 2009)	Majority Voting (MV)	Multiple	Reduce FPs/FNs	Low accuracy
This work	Creditability-based Weighted Voting (CWV)	Multiple	Reduce FPs/FNs	

information about the hosts present on the network and the vulnerability exploited to generate a score for each alert. The score is measure of the importance of the alert. A simple binary classifier then classifies the alert as FP or TP based on value of score threshold.

However, the previous studies belonging to the category of alert classification will pay a lot of overhead because of data mining efforts. Also the processed alerts usually come from only one IDS, so these studies can only process FP cases but cannot investigate FN cases. Hence, they cannot reduce the number of FNs due to the limitation posed by a single IDS. Therefore, attention has turned to the use of multiple IDSs. For instance, Chen et al. (2009) presented a particular method of APP, *Majority Voting* algorithm (MV), to deal with alerts produced by multiple IDSs and reduce the number of FPs and FNs. The idea of MV is to resolve the conflicts among the outcomes of multiple IDSs. MV finds FPs and FNs by comparing the alerts of multiple IDSs. If few IDSs produce alerts from specific traffic traces, the trace is likely to be an FP case of the few IDSs. If few IDSs do not produce alerts, it is likely to be an FN case of the few IDSs. However, Parham (2002) showed that MV is not absolutely correct in many cases, and it would often lead to an incorrect decision. Furthermore, the main cause for the inefficiency of MV is that it disregards the different domain knowledge among multiple IDSs.

Creditability-based Weighted Voting (CWV) algorithm reduces both the number of FPs and FNs and increases the efficiency of APP by leveraging different domain knowledge among multiple IDSs. CWV can investigate the detection creditability of multiple IDSs to overcome the limitation of a single IDS, as well as reduce the number of FPs and FNs to decrease the heavy workload of the analyst. Table 1 summarizes the goals and methods of the different approaches. In this paper, we will focus on a comparison of MV and CWV and evaluate the efficiency of these two algorithms.

2.2. *Generation methods of FP/FN datasets*

In order to evaluate the detection capability of the IDSs, one must pay special attention to the choice of test traces. Some studies used real-world traffic traces to evaluate FPs and FNs and measure the accuracy of the IDSs (Chen et al., 2009; Wang, 2010).

As shown in Fig. 1, Wang designed an Active Trace Collection (ATC) (Wang, 2010) to actively extract and classify suspicious traces from real-world traffic captured in the NCTU Beta Site (Lin et al., 2010). First, in the extraction module, it uses a traffic replay tool to replay the captured traffic to multiple IDSs. If an IDS detects a specific behavior in the traffic, it will trigger an alert. Based on the IDSs' alerts, the ATC

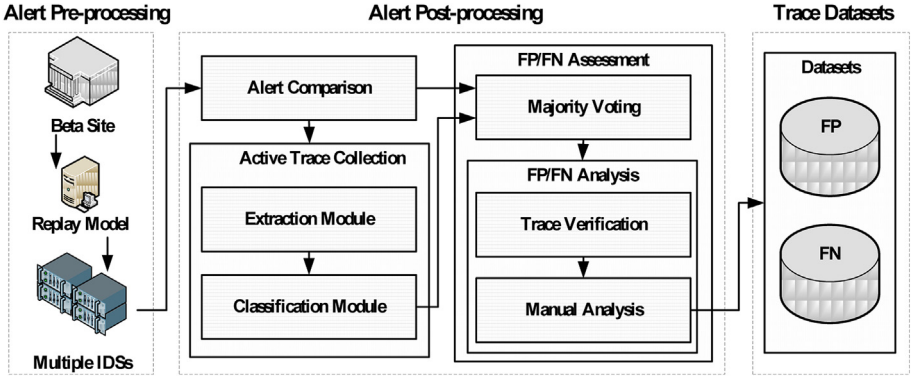


Fig. 1 – Generation method of FP/FN datasets.

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

finds the anchor packets that trigger the alerts by comparing five fields, i.e., source and destination IP addresses, source and destination ports, and protocols. Then it processes the packet and connection association to extract each session into the packet traces. Second, in the classification module, the ATC uses alerts to classify traces into different categories by keywords. Ten categories, listed in Table 2, have been established, including *Web, File Transfer, Remote Access* and others. Each category uses corresponding protocol names as its keywords (Wang, 2010). As an example, the Web category uses HTTP as its keyword. Currently, numerous suspicious classified traces have been collected.

The detection of IDSs may be incorrect due to FPs and FNs. Ho et al. proposed a FP/FN Assessment (FPNA) (Ho et al., 2012) which analyzes the FP and FN cases and investigates the causes of FPs and FNs. First, it finds potential FPs and FNs of the IDSs by using a voting algorithm (e.g., majority voting). Next, in FP/FN analysis, it replays the corresponding extracted traces based on the IDS alerts. This step verifies whether the traces are reproducible to the original IDSs. Then, the reproducible traces are manually analyzed to confirm which cases are correct FPs or FNs. The confirmed FP and FN cases and their causes are recorded to generate the FP/FN datasets. This study further uses the traces and the causes behind the FPs and FNs to investigate the creditability of the IDSs.

In the followings, two case studies of the FP/FN analysis are used as examples to show why the benign traces were detected as malicious ones and the malicious traces were not detected by IDSs. The investigation on the FP/FN analysis is illustrated by the description of activity, the corresponding signature, and the cause of FP/FN, as shown in the respective fields in Fig. 2(a) and Fig. 2(b). The description of the malicious activity refers to Common Vulnerabilities and Exposures (CVE) (Common Vulnerabilities an, 1999). The corresponding signature of the malicious activity refers to the Snort signature database (Sourcefire) as examples. The cause of FP/FN explains why FP/FN occurs.

1) Fig. 2(a) illustrates a false positive case, "WEB-CGI csh access", and a detailed analysis of the packet content using Wireshark (Wireshark et al., 1998) is shown in Fig. 2(b). The execution of the csh interpreter in the cgi-bin directory on a WWW site is detected by simply matching the "/csh" content in the requested URI field. It often results in FP because the signature design is too general and simplistic.
2) Fig. 3(a) illustrates a false negative case, "SQL Worm propagation attempt", and a detailed analysis of the packet content is shown in Fig. 3(b). The SQL Worm would result in buffer overflow in the Microsoft Windows server service. The worm loads Kernel32.dll and WS2_32.dll and then calls GetTickCount to continuously send 376 bytes UDP packet of exploit and propagation codes across port 1434 until the SQL Server process shuts down. However, it sometimes results in FN since some IDSs lack the signature to detect it.

Table 2 – Classification of traces and their representative keywords.

Category	Keywords
Web	HTTP
Email	POP3, SMTP, IMAP
FileTransfer	FTP, SMB, TFTP
RemoteAccess	Telnet, SSH, RDP, VNC
Encryption	SSL, FTPS, HTTPS
Chat	IRC, ICQ, Yahoo Messenger, MSN, AIM, Skype, Google talk
FileSharing	Bittorrent, eDonkey, Gnutella, Pando, SoulSeek, Winny, Xunlei
Streaming	PPLive, QuickTime, Octoshape, Orb, Slingbox
VoIP	SIP
Network	NetBIOS, DNS, SNMP, Socks, STUN

3. Problem statement

3.1. Terminologies

Table 3 defines a confusion matrix to represent the types of trace datasets detected by IDSs. The elements are actual trace behaviors (malicious or benign) and detected alarms (alert or non-alert). According to the corresponding relation between the elements, there are four types of traces: *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), and *False Negative* (FN). TP and FP represent the alerts produced by the IDS for malicious and normal activities, respectively. Similarly, TN and FN mean the IDS does not produce an alert for a normal and a malicious activity, respectively.

Table 4 defines the notations used in this algorithm. M and $\neg M$ respectively denote malicious and benign traces. X represents the number of IDSs involved in detection, and the X IDSs form a set V. Whether all X IDSs have the voting rights depends on the voting algorithm, such as MV and CWV. According to the detection results, one of the four types, TP, FP, TN, or FN, would occur. Furthermore, there are different Y_P^j kinds of alerts produced by the j-th IDS under the protocol P, and $A_{P,k}^j$ records the k-th kind of alerts. After the recording, in order to investigate its creditability, the probability, $R_{P,type}^j$, is used to denote the rate of the trace type generated by the j-th IDS under the protocol P. However, it is possible that some IDSs are not creditable, so two thresholds τ_d and τ_a are used to choose the IDSs with suitable creditability. The set of the chosen IDSs is a subset of V and is denoted as VR. τ_d is the detection threshold, whereas τ_a is the abnormality threshold. Then, according to intrusion detection alarm (alert or non-alert) produced by the j-th IDS for the i-th trace, denoted as d_i^j, a corresponding weight w_i^j is determined. Based on the above notations and definitions, CMD_i can be calculated for malicious tendency of the i-th trace.

3.2. Problem description

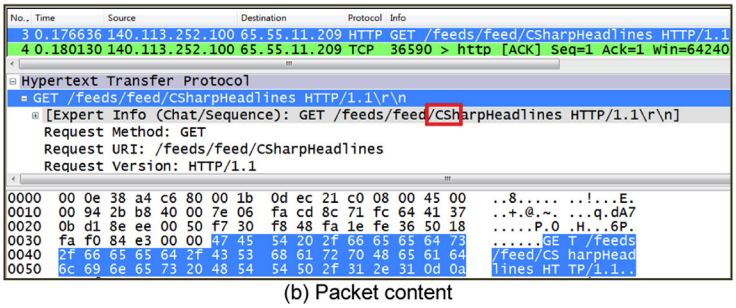
In APP, the efficiency is low when alerts come from only one IDS, as explained in Section 1. On the other hand, when alerts come from multiple IDSs, the efficiency may also be low if APP disregards the different domain knowledge among multiple IDSs. Moreover, different IDSs may have different detection results for the same traffic trace due to their different domain knowledge. How to efficiently use these results to make a good decision on the processed traffic trace is thus a problem.

The above description can be formulated as follows.

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

Description	Perl, sh, csh, or other shell interpreters are installed in the cgi-bin directory on a WWW site, which allows remote attackers to execute arbitrary commands. (reference: CVE, 1999-0509)
Signature	alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI csh access"; flow:to_server,established; uricontent:"/csh"; nocase; ...)
Cause	GET /feeds/feed/CSharpHeadlines HTTP/1.1

(a) Explanation



(b) Packet content

Fig. 2 – A False Positive case study – WEB-CGI csh access.

Given: (1) *X* IDSs, (2) some detected traces for training, (3) alerts produced by *X* IDSs to these traces, and (4) some undetected traces.

Objectives: correctly determine each undetected trace as a malicious trace or a benign one to efficiently reduce FPs and FNs.

4. Creditability-based Weighted Voting

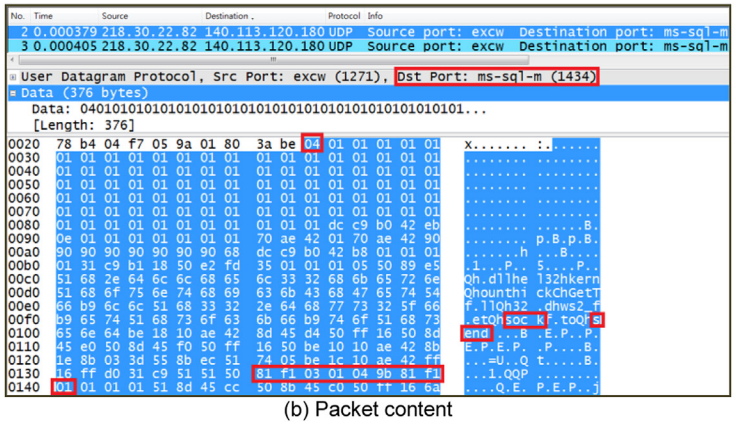
This section details CWV and its four components. The first component is *Creditability Modeling*, which investigates and models the IDSs' creditability according to the past detections. Second, *Authority Selection* determines authorities of detection. Third, *Voter Exclusion* excludes IDSs that perform poorly in the past detections. Last, *Weighted Voting* determines where a trace belongs.

4.1. Overview

The goal of this work is to increase the efficiency of alert post-processing when alerts come from multiple IDSs; that is, to increase the accuracy of the corresponding processed traces which actually belong to TP, FP, TN, or FN cases. The generated TP/FP/TN/FN datasets can thus be used by IDS vendors to improve their signature designs, as well as used to accumulate general knowledge on alerts.

Description	Buffer overflow in the Server Service in Microsoft Windows 2000 SP4, XP SP1 and SP2, and Server 2003 SP1 allows remote attackers, including anonymous users, to execute arbitrary code via a crafted RPC message. (reference: CVE, 2002-0649)
Signature	alert udp $EXTERNAL_NET any -> $HOME_NET 1434 (msg:"SQL Worm propagation attempt"; flow:to_server; content:"|04|"; depth:1; content:"|81 F1 03 01 04 9B 81 F1 01|"; fast_pattern:only; content:"sock"; content:"send"; ...)
Cause	Signature content doesn't exist

(a) Explanation



(b) Packet content

Fig. 3 – A False Negative case study – SQL worm propagation attempt.

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

Table 3 – Confusion matrix definition.

		Detected	
		Alert	Non-alert
Actual	Malicious	True Positive (TP)	False Negative (FN)
	Benign	False Positive (FP)	True Negative (TN)

As shown in Fig. 4, the Active Trace Collection collects and classifies suspicious traffic traces, which are replayed to multiple IDSs, by comparing the alerts produced by different IDSs. Since the detection of IDSs could be incorrect, the FP/FN Analysis investigates the causes of FP/FN using the collected traces and records the confirmed TP/FP/TN/FN traces into the Datasets. Based on the Datasets and the accumulated knowledge of alerts, this work proposes the CWV scheme for making a more accurate decision on suspicious traffic traces.

The main concept in using CWV to increase the efficiency of alert post-processing, i.e., to classify the traces more accurately, is by investigating the IDSs' creditability because an IDS could not perform well on all types of traces, and merging the IDSs' detection results based on their corresponding creditability. Therefore, as shown in Fig. 5, to investigate the IDSs' creditability, *Creditability Modeling* (CM) selects significant types of traces from the Datasets to set up the *Training Data* and uses the *Two-level Modeling* to model the IDSs' corresponding creditability for different types of traces. For the integration of the IDSs' detection results, *Authority Selection* (AS) first selects the IDSs with high detection capabilities to be authorities. If AS does not select an authority, *Voter Exclusion* (VE) excludes the IDSs that cannot usually perform well, i.e., usually produce FPs and FNs. Finally, *Weighted Voting* (WV) uses the IDSs selected by either AS or VE and their corresponding creditability to weight and classify the traces, i.e., malicious, benign, or unknown.

Table 4 – Notations used in creditability-based weighted voting.

Notations	Descriptions
$M(\neg M)$	Malicious trace (benign trace)
X	Number of IDSs.
V:{IDS_1,IDS_2,…,IDS_X}	Set of IDSs.
Type:{*TP, FP, TN, FN*}	Types of the trace dataset.
Y_P^j	Number of kinds of alerts produced by the j-th IDS under the protocol P.
$A_{P,k}^j$	Under the protocol P, the k-th kind of alerts produced by the j-th IDS.
$A_P^j(\neg A_P^j)$	Under the protocol P, any(no) alert produced by the j-th IDS.
P:{*HTTP, FTP,*⋯}	Used protocol of classified traces.
$R_{P,type}^j$	Under the protocol P, rate of the trace type generated by the j-th IDS.
VR	Set of the IDSs which are allowed to vote.
Z	Number of elements of VR.
τ_d	Detection threshold for measuring the correctness of detection.
τ_a	Abnormality threshold for measuring the abnormality of alert frequency.
d_i^j	Intrusion detection alarm produced by the j-th IDS for the i-th trace.
w_i^j	Weight of the j-th IDS for the i-th trace, which is assigned according to the creditability.
CMD_i	The creditability of the malicious decision for the i-th trace.

4.2. CM: Creditability Modeling

An alert is a description of a suspicious activity in a signature. It comprises either one or both factors: protocols and malicious types, and these two factors are thus considered in investigating an alert. Furthermore, a protocol is defined as the first level in this article because it is the most common categorization. Similarly, malicious types belong to the second level due to its detailed description of an alert. Therefore, CM is designed using two-levels to investigate and model the detection capability of IDSs for different types of traffic. As shown in Fig. 5, CM includes two components: *Training Data* (TD) and *Two-level Modeling* (TLM).

According to the TP/FP/TN/FN traces confirmed by the FP/FN Analysis, CM selects significant types of traces to set up the TD. The selection policies are based on the proportion of appearances in traffic and the number of corresponding defined signatures. If both of them are high, the CM will identify the types of traces as significant ones and select them into TD.

Based on the TD, the TLM calculates two detection capabilities for an IDS. One is for the *Alert level* (AL) and the other is for the *Protocol level* (PL). As the name implies, each AL's detection capability depends on the accuracy of an alert and the detection capability of PL is based on certain protocols. Therefore, the conditional probability of each element of the confusion matrix can be calculated as follows.

First, in AL, to know the detection capability of an alert $A_{P,k}^j$, the accuracy rate of this alert, $P(M|A_{P,k}^j)$, analyzed by the FP/FN Analysis, can be calculated as

$$P\left(M|A_{P,k}^j\right) = \frac{C\left(A_{P,k}^j\right)}{T\left(A_{P,k}^j\right)}, \quad 1 \le j \le X, \tag{1}$$

where $T(A_{P,k}^j)$ and $C(A_{P,k}^j)$ are the total number of $A_{P,k}^j$ and the number of $A_{P,k}^j$ that detect the correct intrusion, respectively. Notably, the alert generated by an IDS about an attack, $A_{P,k}^j$, is used only to find the values of $T(A_{P,k}^j)$ and $C(A_{P,k}^j)$ for that IDS. Although different alerts are generated in different IDSs, the alert that is generated by an IDS can be clearly recognized by that IDS. Accordingly, the capability of the IDS to detect the alert can be calculated without standardizing the alert format.

Second, to determine the detection capability of an IDS at generating alerts, we define the successful detection rate as $P(M|A_P^j)$ to mean the probability of correctly detected malicious traces with the protocol P if alerts are generated by the j-th IDS. Based on (1), the successful detection rate is calculated as

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

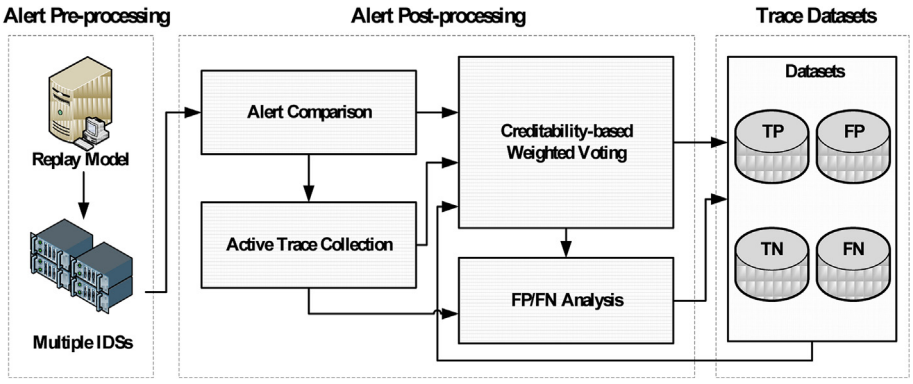


Fig. 4 – Architecture of the proposed system.

$$P\left(M|A_P^j\right) = \frac{\sum_{k=1}^{Y_P^j} C\left(A_{P,k}^j\right)}{\sum_{k=1}^{Y_P^j} T\left(A_{P,k}^j\right)}, \quad 1 \le j \le X, \tag{2}$$

where Y_P^j is the number of kinds of alerts produced by the j-th IDS under the protocol P.

Third, to perceive the detection capability of an IDS at keeping silent, we define the successful omission rate as $P(\neg M|\neg A_P^j)$ to mean the probability of correctly detected benign traces with the protocol P if no any alert is generated by the j-th IDS. According to the Bayes' theorem, the successful omission rate is calculated as

$$P\left(\neg M|\neg A_P^j\right) = \frac{P(\neg M) \times R_{P,TN}^j}{P(\neg M) \times R_{P,TN}^j + P(M) \times R_{P,FN}^j}, \quad 1 \le j \le N. \tag{3}$$

As a result, each IDS has a creditability table which comprises three vectors, i.e., $P(M|A_{P,k}^j)$, $P(M|A_P^j)$ and $P(\neg M|\neg A_P^j)$.

4.3. AS: Authority Selection

Based on the investigation of detection capability of IDSs for different types of traces, AS finds that sometimes some IDSs have a much higher creditability than others. This is because lower FP and FN rates result in higher creditability. Therefore, if the creditability of some IDSs exceeds a decision criterion for a certain type of trace, AS selects these IDSs to be authorities for detecting the specific trace.

AS comprises three steps. First, for each type of trace, AS sorts the FP and FN rates of each IDS from high to low. Then, AS separately calculates the average values of FP and FN rates of the IDSs, denoted as L_1 and L_2, respectively, listed after three-quarters of all IDSs. Third, IDSs whose FP and FN rates are both lower than L_1 and L_2 are selected to be the authorities of detection by AS.

After three steps, there are three possible outcomes: no authority, one authority, or multiple authorities. If no authority occurs, CWV will enter VE and then WV. When there is one authority, the traces will be decided directly by that authority. Otherwise, CWV will enter WV and the traces will be decided by multiple authorities.

4.4. VE: Voter Exclusion

VE is designed to exclude IDSs which usually perform poorly in detection. IDSs are excluded based on two conditions: TP/FP rates and alert frequency.

According to the TP and FP rates, VE excludes IDSs whose TP is less than the detection threshold τ_d. The reason is that some IDSs produce more incorrect than correct detections. VE assumes that such IDSs are not strong and excludes them.

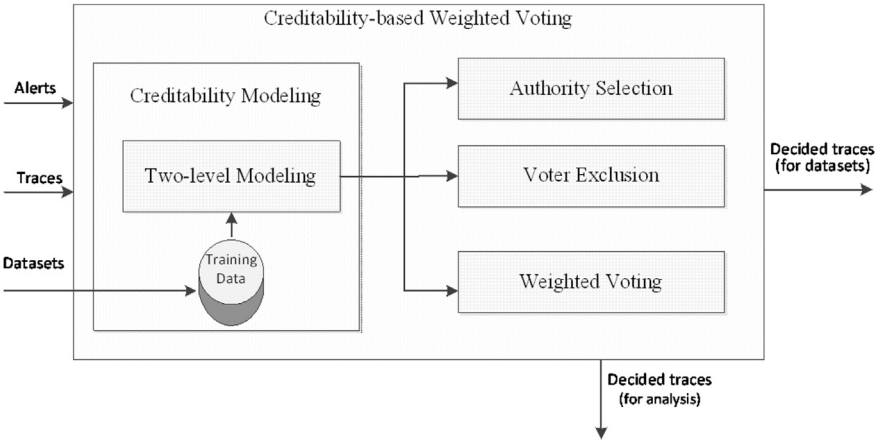


Fig. 5 – Architecture of creditability-based weighted voting.

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

Table 5 – Two-level creditability results of example run.

Creditabilities	IDS1	IDS2	IDS3	IDS4	IDS5	IDS6	IDS7
$P(M\|A^j_{HTTP})$	–	0.46	0.03	–	1.00	–	0.51
$P(\neg M\|\neg A^j_{HTTP})$	0.71	0.78	0.52	0.71	0.75	0.71	0.80
w_{87}	N/A	0.83	N/A	N/A	N/A	N/A	0.20

Based on the alert frequency, VE assumes that IDSs with an abnormal alert frequency are unusual, because some IDSs always or never produce alerts when detecting a specific type of trace. For example, when processing the same type of trace, some IDSs do not produce any alert while others do. Moreover, when an IDS with the detection function for a certain type of trace does not produce any alert, its corresponding signature design is doubted. Thus, in processing the same type of trace, if either the alert rate or the non-alert rate is more than τ_a, i.e., abnormal alert frequency, the IDS is excluded by the VE.

4.5. WV: Weighted Voting

After either multiple authorities are chosen from AS or VE excludes some voters, CWV will enter the last component, WV. First, WV assigns weights to existing voters according to their creditability. Then, when the WV processes the traces one by one, the value of CMD is used to calculate the degree of tendency towards malicious activities. First, the weight of the j-th IDS for the i-th trace, w^j_i, is calculated as

$$w^j_i = \begin{cases} P\left(M|A^j_{P,k}\right), & \text{if } \left(d^j_i = A^j_{P,k}\right) \\ P\left(M|A^j_P\right), & \text{if } \forall k, \ \left(d^j_i \neq A^j_{P,k}\right) \\ 1 - P\left(\neg M|\neg A^j_P\right), & \text{if } \left(d^j_i = \text{NULL}\right) \end{cases} \quad \text{and } d^j_i \neq \text{NULL}, \tag{4}$$

where d^j_i is intrusion detection alarm (alert or non-alert) produced by the j-th IDS for the i-th trace and it is set as NULL when no any alert produced. In (4), there are three conditions to calculate w^j_i. The first condition is that the j-th IDS produces an alert and this alert belongs to the alerts generated for the training data. It can belong to AL with $P(M|A^j_{P,k})$. The second condition is that the j-th IDS produces an alert but the alert does not belong to the previous alerts. It can only be calculated in PL with $P(M|A^j_P)$. The last condition is that the j-th IDS does not produce an alert. It is calculated in PL with $1 - P(\neg M|\neg A^j_P)$.

Then WV calculates the overall creditability of the malicious decision for the i-th trace by

$$CMD_i = \frac{1}{Z}\sum_{j=1}^{Z} w^j_i, \tag{5}$$

where Z is the number of voters.

Finally, WV makes a decision on the i-th trace using $CMDi$ to decide whether the trace is malicious, benign or unknown. The i-th trace is determined as malicious if its $CMDi$ is more than α, while it is benign if the $CMDi$ is less than β. When the value ranges between α and β, the malicious or benign tendency is not very obvious. Hence, this trace is classified into the unknown one.

Notably, most IDSs not only report an alert by text, but also report a number that specifies the generated alert. For such an IDS, the alert number is utilized to determine which alert and to find rapidly its weight of the capability for detecting the alert with a constant time complexity. For an ID that reports the alert only by text, the reported text is compared with the alert texts that were generated by this IDS and stored in the database. A hashing function is utilized to determine whether they match, so the time complexity is constant. Thus, CWV has a low workload and a low computational complexity in the detection phase.

4.6. Example of creditability-based Weighted Voting

Assume there are seven IDSs (i.e., $X = 7$) which detect the same traffic and produce corresponding alerts. By comparing the alerts, the HTTP traces can be collected and are given as examples here. After the FP/FN Analysis, the TP/FP/TN/FN datasets can be set up. Then, CM sets up the TD according to the datasets and calculates $R^j_{HTTP,TP}$, $R^j_{HTTP,FP}$, $R^j_{HTTP,TN}$, and $R^j_{HTTP,FN}$. Next, CM uses the TLM to model the seven IDSs' corresponding creditability. First, in AL, the accuracy rate of the k-th alert is calculated as $P(M|A^j_{HTTP,k})$ and thus the accuracy rates of all alerts can be obtained. Next, PL calculates the successful detection rate $P(M|A^j_{HTTP})$ and successful omission rate $P(\neg M|\neg A^j_{HTTP})$, which are shown in Table 5.

After CM, the other three components of CWV can process HTTP traces with a two-level creditability. First, in AS, the L_1 and L_2 are 0 and 0.51 respectively. By comparing each IDS's $R^j_{HTTP,FP}$ and $R^j_{HTTP,FN}$ with L_1 and L_2 respectively, there is no authority for the detection. Next, in VE, the 3rd IDS is excluded according to the TP/FP rates. The 1st, 4th, 5th and 6th IDSs are excluded according to the abnormal alert frequency. Hence, the remaining voters are the 2nd and 7th IDSs. Finally, in WV, when processing the 87th trace, the 2nd IDS produces an alert "IBM Lotus Domino Accept-Language Buffer Overflow" which belongs to the previous alerts in the training data, while the 7th IDS does not produce any alert. The weight of the 2nd IDS of the alert in AL, $P(M|d^2_{87})$, is 0.83. The weight of the 7th IDS of the non-alert in PL, $1 - P(\neg M|\neg A^7_{HTTP})$, is 0.20. Therefore, the CMD_{87} is calculated as $(0.83 + 0.20)/2$, which is 0.52. Because the value is larger than 0.5 ($\alpha = 0.5$), the 87th trace is decided as a malicious one.

5. Evaluation and observation

In this section, the detection capability of multiple IDSs and the performance of CWV are evaluated. First, the IDSs' corresponding creditability of different types of traffic traces modeled by the CM are illustrated. Second, the Accuracy, TPR, TNR and Efficiency are used to evaluate the voting algorithms.

5.1. Trace selection and experiment environment

As mentioned in Section 4.2, trace selection policies are based on the rate of appearance in traffic and the rate of number of

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

Table 6 – Investigation result of trace selection.

Category	Web	File sharing	Chat	File transfer	Network	Remote access	VoIP	Encryption	Email	Streaming
% Of traffic	35.86	32.69	8.82	7.07	4.84	4.05	3.14	2.79	0.49	0.22
% Of signature	81.78	0.14	0.57	2.13	8.80	0.66	1.41	0.00	4.48	0.04

corresponding signatures. If both are significant for a certain type of trace, the trace will be selected. According to the ten categories classified by ATC (Wang, 2010), we investigated the traffic in the NCTU Beta Site (Common Vulnerabilities an, 1999) during the period from September 1, 2010 to February 1, 2011 to understand the frequently appearing categories in traffic. Second, we took the rule version 2.9 of Snort as an example to investigate signature classification and distribution. The investigation results are shown in Table 6. Initially we chose five types, Web, File Transfer, Network, Remote Access, and VoIP, because they have more signatures and could generate more alerts. Then VoIP is skipped because we are not very familiar with it. From Web, File Transfer, Network and Remote Access, we selected the most popular protocol, respectively. Hence, the four types of traces were decided: HTTP, FTP, NetBIOS and TELNET.

The real-world traffic captured from the NCTU Beta Site (Lin et al., 2010), during the period from September 1, 2010 to February 1, 2011, occupies 10Tbytes. We then used a traffic replay tool (e.g., tcpreplay) to replay the captured raw traffic to multiple IDSs. Seven IDSs are involved in the classification, as shown in Table 7. Table 8 presents the number of the four selected types of traces, where a trace means a flow. The size of the traffic set is large while the numbers of traces in Table 8 are not large. The reason is that the traces shown in Table 8 only include the replayed traffic which triggers at least one alert in seven IDSs. Since Snort may classify inaccurately, it is used for rough traffic classification to determine the percentage of each traffic type and the percentage of the signature in each category, as indicated in Table 6. A manual analysis was carried out as an extra precaution to obtain the number of traffic traces in Table 8. That is, whether the traces are malicious or benign were confirmed by some experts.

Since Snort may classify inaccurately, it is used for rough traffic classification to determine the percentage of each traffic type and the percentage of the signature in each category, as indicated in Table 6. A manual analysis was carried out as an extra precaution to obtain the number of traffic traces, as shown in Table 8. Hence, experts were able to confirm whether the traces are malicious or benign.

The ratio of malicious traces to benign ones is about 4–6. The rate of benign traces is lower than anticipated since we expected to avoid a flood of benign traces from dominating the results in this experiment. During the period, the two dominant types of traces were HTTP and NetBIOS. In the HTTP traffic, 39% of the traces were malicious, meaning HTTP applications are frequently exploited. In the NetBIOS traffic, 62% of the traces were malicious, meaning the vulnerabilities of NetBIOS are usually targeted by attackers. Here, we choose the traces collected in the first two months to be the training data and the traces of the latter three months to be the processing data. The former served as input for the CM to set up the TD while the latter served as input for measuring the accuracy of the CWV.

The parameters in CWV were set as follows. In the VE, the detection threshold τ_d was set at 0.5 while the abnormality threshold τ_a was set at 0.9. In WV, the values of α and β were both set at 0.5. The parameters are discussed in detail in Section 5.3.

5.2. *Experiment results of investigation of creditabilities*

In the CM evaluation, this work takes seven IDSs, which are called IDS1, IDS2, …, and IDS7, respectively, as examples to represent the IDSs' corresponding creditability for different types of traffic traces at two levels.

As mentioned in Section 4.2, the successful detection rate and successful omission rate are defined as $P(M|A_P^j)$ and $P(\neg M|\neg A_P^j)$, respectively, to represent the detection capability for PL. As shown in Table 9, when the value of detection rate is '–', it means that it is uncalculated; that is, the IDS does not produce any alert for the type of traces. Where the value of detection rate is 0.00, it means that the alerts result from commonly used commands, i.e., the traffic is always benign. For example, some alerts produced by IDS5 for FTP traces result from a commonly used FTP command. Where the value of detection or omission rates is 1.00, it means that the definition of the signature for the type of trace is more precise. For instance, only one type of alerts is produced by IDS5 for TELNET traces and after our investigation, this detection result is correct. Notably, the IDSs' detection capability for different protocols is different. In our investigation, for HTTP, IDS2, IDS5 and IDS7 have a higher creditability. For FTP, IDS5, IDS6 and IDS7 have a higher creditability. For NetBIOS, IDS1, IDS4, IDS5 and IDS6 have a higher creditability. For TELNET, IDS3 and IDS5 have a higher creditability. Generally, IDS5 achieves satisfied successful rates in each protocol.

5.3. *Accuracy, TPR, TNR, and Efficiency of voting algorithms*

Let TP_{traces} be the number of malicious traces which are correctly determined, FN_{traces} be the number of malicious traces which are not determined, TN_{traces} be the number of

Table 7 – Vender and device names on seven IDSs.

Vendor name	BroadWeb	D-link	Fortinet	McAfee	Tipping-point	Trend micro	ZyXEL
Device name	NetKeeper7K	DFL-1600	FortiGate-110c	M-1250	5000E	TDA2	ZyWALL USG 1000

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

Table 8 – Statistics for number of traffic traces.

Type	Malicious	Benign	Total
(A) Training data			
HTTP	46	72	118
FTP	22	74	96
NetBIOS	66	47	113
TELNET	4	31	35
Total	138	224	362
(B) Processing data			
HTTP	57	86	143
FTP	29	77	106
NetBIOS	87	46	133
TELNET	5	42	47
Total	178	251	429

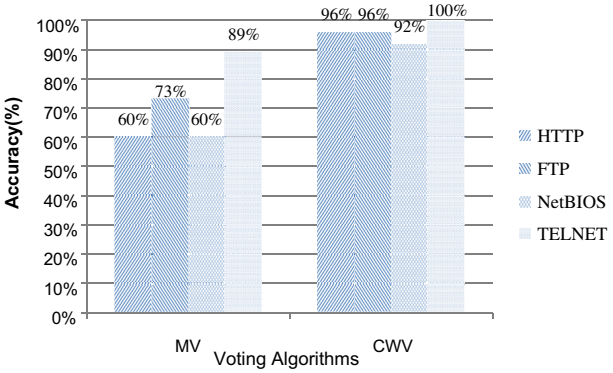


Fig. 6 – Accuracy of the voting algorithms.

benign traces which are correctly classified, FP_{traces} be the number of benign traces which are incorrectly determined as malicious ones.

This work uses the *Accuracy*, *TPR*, and *TNR* metrics (Wu and Banzhaf, 2010) for the voting algorithm in the evaluation. *Accuracy* is evaluated with the percentage of whole traces that are determined accurately. This is a commonly used metric for an overall view of evaluation.

$$Accuracy = \frac{TP_{traces} + TN_{traces}}{TP_{traces} + FP_{traces} + TN_{traces} + FN_{traces}} \times 100\%.$$

TPR is evaluated with the percentage of malicious traces that are correctly determined as malicious ones, while the *TNR* is evaluated with the percentage of benign traces that are correctly determined as benign ones.

$$TPR = \frac{TP_{traces}}{TP_{traces} + FN_{traces}} \times 100\%$$
$$TNR = \frac{TN_{traces}}{TN_{traces} + FP_{traces}} \times 100\%.$$

There is a tradeoff between TPR and TNR, so when evaluating the performance of a voting algorithm, we have to consider both TPR and TNR. Similar to the F1 score (Rijsbergen, 1979) which is a measure of a test's accuracy, this work defines a similar measure for the efficiency of the voting algorithm. Efficiency takes the harmonic mean of TPR and TNR, given by:

$$Efficiency = \frac{2}{\frac{1}{TPR} + \frac{1}{TNR}} \times 100\%.$$

A higher value of *Efficiency* indicates that the voting algorithm performs better on not only *TPR*, but also *TNR*.

5.3.1. *Experimental evaluation results*

Fig. 6 shows the complete accuracy results of CWV and MV. It is observed that each accuracy of CWV is higher than that of MV. The overall accuracy of CWV and MV are 95% and 66%, respectively, by calculating them for all traces. It is observed that CWV is improved by about 1.4 times as that of MV. The results demonstrate that the weights of IDSs should be different for leveraging the different domain knowledge among the IDSs when multiple IDSs are involved in the detection.

Figs. 7 and 8 compare the TPR and TNR of CWV with MV. The results mainly demonstrate the effect of a two-level creditability modeling. The average TPR of CWV and MV are 93% and 14%, respectively, meaning that CWV has a lower FN rate. The reason is that the FNs of some IDSs could be avoided by leveraging other IDSs' correct detection using the corresponding creditability. The average TNR of CWV and MV are 98% and 93%, respectively, meaning that the CWV has a lower FP. The main reason is that, in CWV, the FPs of some IDSs could be filtered by the creditability, especially in AL. In CWV, the TNR is higher than the TPR because the correctness of the alert itself is investigated in AL. Thus, an alert with a frequent FP would be filtered. In addition, the TPR and TNR of MV for HTTP, FTP, and TELNET are 0% and 100%, respectively. The reason is that only a few IDSs produce alerts, which means either FNs occur in most IDSs or FPs occur in few IDSs.

Table 9 – Experiment results of successful detection and omission rates in protocol level for each IDS.

IDSs	Types							
	HTTP		FTP		NetBIOS		TELNET	
	$P(M\|A_p^j)$	$P(\neg M\|\neg A_p^j)$	$P(M\|A_p^j)$	$P(\neg M\|\neg A_p^j)$	$P(M\|A_p^j)$	$P(\neg M\|\neg A_p^j)$	$P(M\|A_p^j)$	$P(\neg M\|\neg A_p^j)$
IDS1	–	0.71	–	0.92	0.95	0.69	0.28	0.99
IDS2	0.46	0.78	–	0.92	–	0.33	–	0.98
IDS3	0.03	0.52	0.00	0.78	0.66	0.33	1.00	0.99
IDS4	–	0.71	–	0.92	0.68	1.00	–	0.98
IDS5	1.00	0.75	0.74	0.98	0.88	0.83	1.00	0.99
IDS6	–	0.71	0.69	1.00	0.67	0.68	0.00	0.98
IDS7	0.51	0.80	0.70	0.95	0.41	0.31	0.01	0.72

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

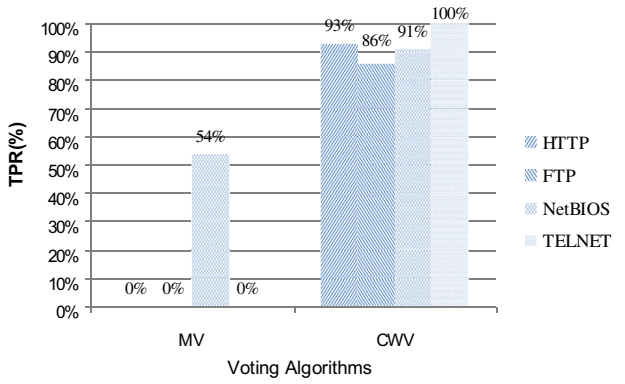


Fig. 7 – TPR of the voting algorithms.

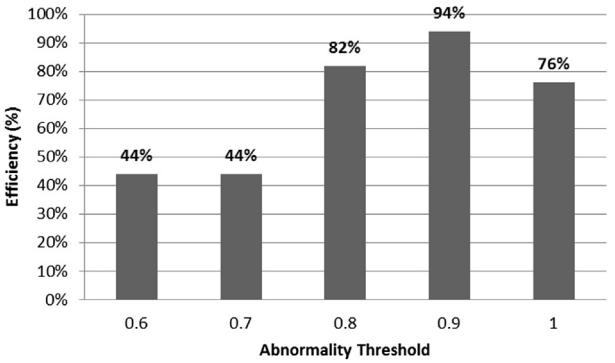


Fig. 9 – Efficiency of various abnormality thresholds of CWV.

Regardless of the situation, MV makes a classification directly from the same weighted IDSs; i.e., it may underestimate the judgments of some IDSs which have a noticeably high creditability. Accordingly, MV could make incorrect classifications, especially in the case that most of them are FNs.

From TPR and TNR showed in Figs. 7 and 8, respectively, the efficiency of MV is 41%, while that of CWV is as high as 94%. CWV can maintain an efficiency at about 2.3 times higher than that of MV. This means CWV can maintain both TPR and TNR well.

5.3.2. *Discussion of important parameters in CWV*

In VE, the detection threshold τ_d is set at 0.5, which means half of the detection is correct, i.e., the probability of intuition is 50%. Based on this value, we experiment with various abnormality threshold values between 0.6 and 1.0, and the results are shown in Fig. 9. Values smaller than 0.5 are not used in this experiment because they result in no voters being used. It does not make sense when there are no voters in a voting. As can be seen in Fig. 9, when the abnormality threshold is 0.9, CWV has the highest efficiency. Therefore, we use this value in all the experiments of this article.

Similarly, in WV, we changed α and β from 0.1 to 0.9 and found that the accuracy can be 100% when the values of α and β are 0.7 and 0.1, respectively. However, the range between α and β is large, 0.6 (= 0.7–0.1), so the number of unknown cases is up to 55% of total processed traffic traces. Hence, α and β can be tuned as a result of a tradeoff between the accuracy of the decided traces and the number of unknown traces that need to be analyzed manually.

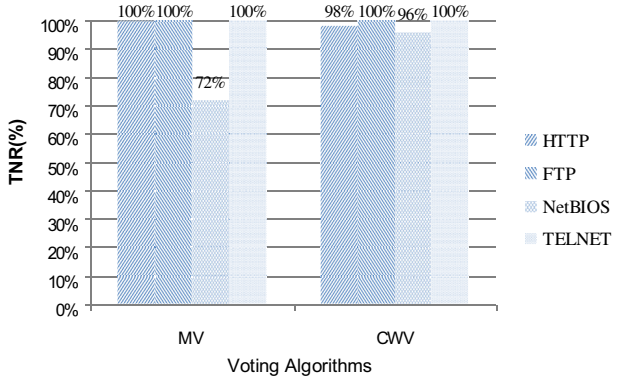


Fig. 8 – TNR of the voting algorithms.

5.4. *Differences between CWV and each IDS in percentages at FP and FN*

Table 10 shows the percentages of FP and FN of CWV and each IDS for different types of traces. Some IDSs have FP and FN values with 0% and 100% since these IDSs do not produce alerts for this type of traces. This also means these IDSs lack the respective signatures. Notably, the FP of IDS3 is 100% because the alerts result from commonly used commands, such as the "FTP GET command". For IDS5, both the FP and FN are 0%. The reason is the IDS5 produced only one type of alerts, and the alert is "SOLARIS.TELNETD.AUTHENTICATION.EXP", which is a precise signature in our investigation, i.e., the creditability of the alert is 1.0. In the evaluation, the traces that generate this alert are always malicious ones. For NetBIOS traces, most IDSs produce many alerts that result in more FPs for each IDS, while for other types of traces, many IDSs do not produce alerts, resulting in more FNs.

The differences between CWV and each IDS in the percentage of FP and FN are shown in Table 11. Some detection results of IDSs are partially better than those of the CWV because the percentage of FP or FN is negative, but no IDS can individually detect well in both FP and FN. CWV performs well in most cases for all types of traces by leveraging the different detection capability among the IDSs, i.e., different percentages of FP and FN. It is demonstrated that the average percentage of FP and FN reduction for CWV and for each IDS is 21% and 58%, respectively.

5.5. *Case studies*

In this section, two case studies in the experiment are given as examples to show the TP case in CWV and FN in MV, and the TN case in CWV and FP in MV.

5.5.1. *Case study I: TP case in CWV and FN in MV*

Alerts and the corresponding creditability are shown in Fig. 10(a), while the trace content is illustrated in Fig. 10(b). It is

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

Table 10 – Percentages of FP and FN of CWV and each IDS.

	HTTP		FTP		NetBIOS		TELNET	
	FP	*FN*	*FP*	*FN*	*FP*	*FN*	*FP*	*FN*
IDS1	0	100	0	100	63.04	5.17	0	100
IDS2	26.74	94.74	0	100	0	100	0	100
IDS3	63.95	100	100	86.21	80.43	1.15	2.38	40.00
IDS4	0	100	0	100	25.53	7.58	0	100
IDS5	0	98.25	0	48.28	52.17	9.20	0	0
IDS6	0	100	0	13.79	67.39	1.15	0	100
IDS7	9.30	5.26	0	48.28	39.13	82.76	97.62	80.00
CWV	2.33	7.02	0	13.79	4.35	9.20	0	0

observed that the attacker uses the command "USER –fadm" as the argument injection via the USER environment variable in the environment option to attempt to bypass the authentication. More information about Telnet environment option can be found in (Alexander, 1994). Furthermore, the malicious content in hexadecimal is "ff fa 27 00 00 55 53 45 52 01 2d 66 61 64 6d". However, this malicious trace can be correctly determined by CWV because of the high creditability in AL, while it is missed by MV because only a few voters can detect it.

5.5.2. *Case study II: TN case in CWV and FP in MV*

The alerts and the corresponding creditability are shown in Fig. 11(a), while the trace content is illustrated in Fig. 11(b). It is observed that the signature designs are not specific. Hence, the general signature is easily matched in the payload even though the payload is benign. Obviously, logon/login failure is a general outcome that often occurs in normal activities. Our investigation demonstrated that the corresponding creditability of this trace is low. Therefore, this benign trace can be correctly determined as a benign one by CWV, while it is incorrectly classified as a malicious one by MV because most voters detected it.

6. Conclusions and future work

This work proposes Creditability-based Weighted Voting (CWV) to reduce both FPs and FNs and increase the efficiency of alert post-processing by using multiple IDSs. Creditability Modeling (CM) leverages the domain knowledge among multiple IDSs by investigating the detection capability of all the IDSs and models the corresponding creditability for them. From the experiment results, we demonstrate the different IDSs' detection capability by their creditability. We observed that the signature design is the main factor in the correctness of detection. Some IDSs have more *specific* signatures that result in fewer numbers of alerts and FPs, while some IDSs have more *generic* signatures that result in larger numbers of alerts and FPs. On the other hand, some IDSs lack certain signatures, which results in FNs.

This work uses accuracy, TPR and TNR, and defines efficiency to evaluate two voting algorithms, CWV and MV. CWV can achieve an accuracy and efficiency up to 95% and 94%, respectively, which are much higher than MV in comparison. Between CWV and each IDS, CWV performs well for most cases of all types of traffic traces. It is demonstrated that the average percentage of FP and FN reduction for CWV and each IDS are 21% and 58%, respectively.

However, CWV may make an incorrect decision in some situations. For example, when processing a trace which triggers a new alert in some IDSs, CWV can only use the corresponding creditability of the IDs at the Protocol level (PL), but not at the Alert level (AL), to determine the trace, which leads to an incorrect decision. In addition, if an IDS significantly updates or modifies its signature database, which means the detection capability changes greatly, the corresponding creditability would be almost useless; in this case, CWV may make an incorrect decision on the trace. Hence, the frequency and the duration of updating the training data are issues to resolve in the future. Currently, this work does not distinguish between host-based and network-based alerts. Clearly, they should be distinguished to adjust the IDSs' creditability further. Such work shall be performed in the future.

Another goal in the future is the automation of the analysis because it could increase the productivity and practicability of the proposed scheme. In the *foreground*, CWV keeps

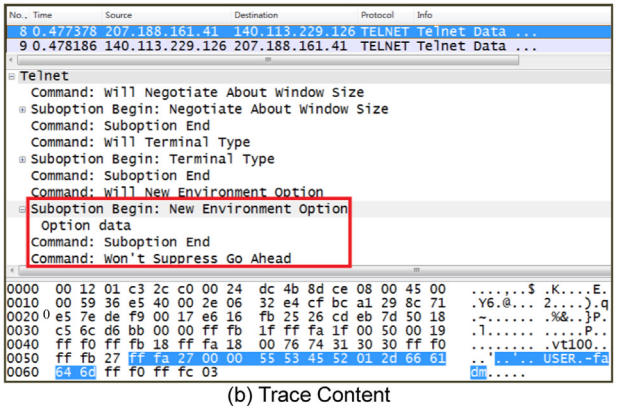
Table 11 – Differences between CWV and IDS in percentages of FP and FN.

	HTTP		FTP		NetBIOS		TELNET	
	FP	*FN*	*FP*	*FN*	*FP*	*FN*	*FP*	*FN*
IDS1	–2.33	92.98	0	86.21	58.69	–4.08	0	100
IDS2	24.41	87.82	0	86.21	–4.35	90.8	0	100
IDS3	61.62	92.98	100	72.42	76.08	–8.05	2.38	40.00
IDS4	–2.33	92.98	0	86.21	21.18	–1.62	0	100
IDS5	–2.33	91.23	0	34.49	47.82	0	0	0
IDS6	–2.33	92.98	0	0	63.04	–8.05	0	100
IDS7	6.97	–1.76	0	34.49	34.78	73.56	97.62	80.00
Average	11.95	78.44	14.29	57.15	42.46	20.37	14.29	74.29

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010

Alert	$P(M \mid d_i^j)$
SOLARIS.TELNETD.AUTHENTICATION.EXP	1.00
Solaris Telnetd Authentication Bypass Vulnerability	0.80

(a) Alerts and Corresponding Creditability



(b) Trace Content

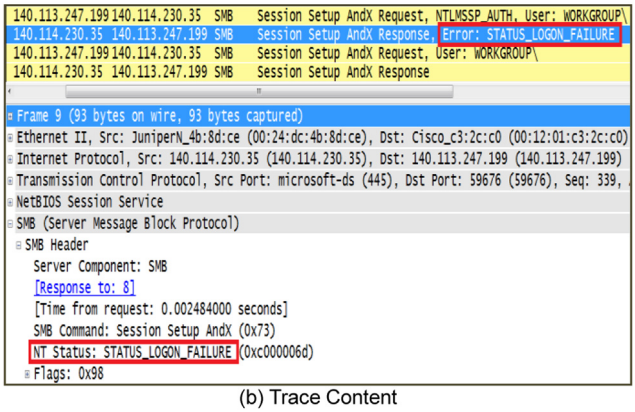
Fig. 10 – Case study i.

processing traffic traces one by one, while in the *background*, the creditability table for each IDS is updated while considering the above issues to maintain the reliance on creditability.

Finally, the ATC was used to extract actively and classify suspicious traces from real-world traffic that were captured on the NCTU Beta Site. Based on the collected dataset, CWV can significantly reduce the numbers of false positives and false negatives. However, the dataset used affects the evaluation results. A large training dataset is required for CWV to obtain more correct weights of IDSs. Recent work has been conducted to collect more complete and typical datasets (Creech and Hu, 2013; Vasudevan et al., 2011). These newer datasets should be used in the future to validate the applicability of CWV.

Alert	$P(M \mid d_i^j)$
netbios: SMB.Login.Failure	0.26
SMB: Windows Logon Failure	0.12
EXPLOIT Server Service Remote Code attack	0.62
NETBIOS-SS: NULL Credentials Login	0.50

(a) Alerts and Corresponding Creditability



(b) Trace Content

Fig. 11 – Case study II.

Acknowledgments

This work was supported in part by National Science Council and Institute of Information Industry in Taiwan.

REFERENCES

IETF RFC 1572. In: Alexander S, editor. Telnet environment option. Lachman Technology, Inc; January 1994.

Chen I-W, Lin P-C, Luo C-C, Cheng T-H, Lin Y-D, Lai Y-C, et al. Extracting attack sessions from real traffic with intrusion prevention systems. In: Proceeding of IEEE international conference on communications (ICC) June 2009.

Clifton C, Gengo G. Developing custom intrusion detection filters using data mining. In: Proceeding of military communications symposium, vol. 1; 2000. p. 440–3.

Common vulnerabilities and exposures (CVE) [Online]. Available: http://cve.mitre.org/; 1999.

Creech G, Hu J. Generation of a new IDS test dataset: time to retire the KDD collection. In: IEEE wireless communications and networking conference April 2013. p. 4487–92.

Gupta D, Joshi PS, Bhattacharjee AK, Mundada RS. IDS alerts classification using knowledge-based evaluation. In: International conference on communication systems and networks January 2012. p. 1–8.

Ho C-Y, Lai Y-C, Chen I-W, Wang F-Y, Tai W-H. Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems. IEEE Commun Mag March 2012;vol. 50(3):146–54.

Julisch K. Mining alarm clusters to improve alarm handling efficiency. In: Proceeding of the 17th annual computer security applications conference 2001. p. 12.

Julisch K. Clustering intrusion detection alarms to support root cause analysis. ACM Trans Info Sys Security (TISSEC) 2003a;vol. 6(4):443–71.

Julisch K. Using root cause analysis to handle intrusion detection alarms. Ph.D. dissertation. University of Dortmund; 2003b.

Latif-shabgahi G, Bass JM, Bennett S. A taxonomy for software voting algorithm used in safety-critical systems. IEEE Trans Reliability 2004;vol. 53(3):319–28.

Lin Y-D, Chen I-W, Lin P-C, Chen C-S, Hsu C-H. On campus beta site: architecture designs, operational experience, and top product defects. IEEE Commun Mag 2010;vol. 48:83–91.

Long J, Schwartz D, Stoecklin S. Distinguishing false from true alerts in snort by data mining patterns of alerts. In: Proceeding of SPIE defense and security symposium, vol. 6241; April 2006. 62410B-1–62410B-10.

Maggi F, Matteucci M, Zanero S. Reducing false positives in anomaly detectors through fuzzy alert aggregation. Info Fusion J October 2009;10(4):300–11.

Ning P, Xu D. Learning attack strategies from intrusion alert. In: Proceeding of the 10th ACM conference on computer and communications security October 2003. Washington D.C., USA.

Ning P, Xu D, Healey C, St. Amant R. Building attack scenarios through integration of complementary alert correlation methods. In: Proceeding of the 11th annual network and distributed system security symposium February 2004.

Ning P, Cui Y, Reeves Douglas S. Constructing attack scenarios through correlation of intrusion alerts. In: Proceeding of the 9th ACM conference on computer and communications security November 2002. p. 245–54. Washington, DC, USA.

Parham B. Voting algorithms. IEEE Trans Reliability 2002;vol. 43(4):617–29.

Pietraszek T. Using adaptive alert classification to reduce false positives in intrusion detection; 2004. p. 102–24. Lecture Notes In Computer Science.
Pietraszek T. Alert classification to reduce false positives in intrusion detection. Ph.D. dissertation. Germany: Albert-Ludwigs-Universität Freiburg im Breisgau; 2006.
Porras P, Fong M, Valdes A. A mission-impact-based approach to INFOSEC alarm correlation. In: International symposium on the recent advances in intrusion detection, vol. 2516; 2002. p. 95–114.
Rijsbergen CJ. Information retrieval. London: Butterworths; 1979.
Sadoddin R, Ghorbani A. Alert correlation survey: framework and technique. In: Proceeding 2006 international conference on privacy, security and trust: bridge the gap between PST technologies and business services, vol. 380; November 2006.
Sourcefire. Snort: an open source network intrusion prevention and detection System, [Online]. Available: http://www.snort.org/vrt.
Treinen JJ, Thurimella R. A framework for the application of association rule mining in large intrusion detection infrastructures. In: International symposium on the recent advances in intrusion detection, vol. 4219; 2006. p. 1–18.
Vaarandi R. Real-time classification of IDS alerts with data mining techniques. In: Proceeding of military communications symposium 2009. p. 1–7.
Vaarandi R, Podins K. Network IDS alert classification with frequent itemset mining and data clustering. In: International conference on network and service management October 2010. p. 451–6.
Valdes A, Skinner K. Probabilistic alert correlation. In: International symposium on the recent advances in intrusion detection, vol. 2212; 2001. p. 54–68.
Valeur F, Vigna G, Kruegel C, Kemmerer RA. A comprehensive approach to intrusion detection alert correlation. IEEE Trans Depend Sec Comp September 2004;vol. 1(3):146–69.
Vasudevan AR, Harshini E, Selvakumar S. SSENet-2011: a network intrusion detection system dataset and its comparison with KDD CUP 99 dataset. In: Second Asian Himalayas international conference on internet March 2011.
Viinikka J, Debar H, Me L, Lehikoinen A, Tarvainen M. Processing intrusion detection alert aggregates with time series modeling. Info Fusion J October 2009;10(4):312–24.
Wang S-H. Extracting, classifying and anonymizing packet traces with case studies on false positives/negatives assessment. M.S. thesis. Taiwan: Department of Computer Science, National Chiao Tung University; 2010.
Wireshark [Online]. Available: http://www.wireshark.org/; 1998.
Wu SX, Banzhaf W. The use of computational intelligence in intrusion detection systems: a review. App Soft Comp 2010;vol. 10:1–35.
Xu Ming, Wu Ting, Tang Jing Fan. An IDS alert fusion approach based on happened before relation. In: International conference on wireless communications, networking and mobile computing October 2008. p. 1–4.
Yu D, Frincku D. Alert confidence fusion in intrusion detection systems with extended Dempster–Shafer theory. In: Proceeding of the 43rd annual southeast regional conference, vol. 2; 2005. p. 142–7.
Zhang Y, Huang S, Wang Y. IDS alert classification model construction using decision support techniques. In: International conference on computer science and electronics engineering March 2012. p. 301–5.

Ying-Dar Lin is Professor of Computer Science at National Chiao Tung University (NCTU) in Taiwan. He received his Ph.D. in Computer Science from UCLA in 1993. Since 2002, he has been the founder and director of Network Benchmarking Lab (NBL, www.nbl.org.tw), which reviews network products with real traffic. He also cofounded L7 Networks Inc. in 2002, which was later acquired by D-Link Corp. His research interests include network security, deep packet inspection, P2P networking, and embedded hardware/software co-design. His work on "multi-hop cellular" has been cited over 500 times. He is an IEEE fellow and on the editorial boards of IEEE Transactions on Computers, IEEE Network, IEEE Communications Magazine Network Testing Series, IEEE Communications Surveys and Tutorials, IEEE Communications Letters, Computer Communications, and Computer Networks. He published a textbook "Computer Networks: An Open Source Approach" with Ren-Hung Hwang and Fred Baker through McGraw-Hill in February 2011.

Yuan-Cheng Lai received the Ph.D. degree in Computer Science from National Chiao Tung University in 1997. He joined the faculty of the Department of Information Management at National Taiwan University of Science and Technology in 2001 and has been a professor since 2008. His research interests include wireless networks, network performance evaluation, network security, and content networking.

Cheng-Yuan Ho is a R&D manager in the Advanced Research Institute, Institute for Information Industry, Taipei, Taiwan. His research interests include the design, analysis, and modeling of the congestion control algorithms, real-flow test, high speed networking, embedded hardware-software co-design, quality of service, and mobile and wireless networks. Ho has a PhD in Computer Science from National Chiao Tung University, Hsinchu, Taiwan.

Wei-Hsuan Tai received his M.S. degree in Computer Science from National Chiao Tung University in 2011. His advisor is Prof. Ying-Dar Lin and mentor is Dr. Cheng-Yuan Ho.

Please cite this article in press as: Lin Y-D, et al., Creditability-based weighted voting for reducing false positives and negatives in intrusion detection, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.010