10

Correlation Analysis Between Honeypot Data and IDS Alerts Using One-class SVM

Jungsuk Song[1], Hiroki Takakura[2], Yasuo Okabe[3] and Yongjin Kwon[4]
[1]*National Institute of Information and Communications Technology (NICT)*
[2]*Information Technology Center, Nagoya University*
[3]*Academic Center for Computing and Media Studies, Kyoto University*
[4]*Information and Telecom. Eng., Korea Aerospace University*
[1,2,3]*Japan*
[4]*Korea*

1. Introduction

With the rapid development and proliferation of the Internet infrastructure and local networks, more and more security threats, *e.g.*, distributed denial of service (DDoS), computer viruses, Internet worms, trojan horses, sywares, adwares and bots, for computer systems and networks are also constantly emerging as well as evolving. There have been many efforts on fighting against these security threats in the last decade, which include cryptography, firewalls and intrusion detection systems (IDSs), etc. Especially, IDS(Base & Mell, 2001; Denning, 1987) is becoming increasingly significant to maintain high-level network security and to defend our crucial computer systems and networks from malicious attackers(Allen et al., 2000).

There are mainly two kinds of intrusion detection methods: misuse detection and anomaly detection. IDSs based on misuse detection method monitor network traffic to detect invalid incoming and/or outgoing accesses using predefined attack signatures(Bro, 2010; Snort, 2010). If they find any intrusion or suspicious activities which include the patterns of predefined signatures, they raise the corresponding alerts. In anomaly detection based IDSs, on the other hand, they use normal patterns to detect abnormal activities from observed data. They attempt to identify deviations from predefined normal patterns, and if such activities are observed over the network, then they are regarded as potential attacks. The former has capability to detect well-known attacks with a relatively high accuracy than that of the latter, but they have a fatal limitation in that they are unable to detect unforeseen attacks, *i.e.*, 0-day attacks, which are not included in the set of predefined signatures. While IDSs based on anomaly detection method have the potential capability of detecting unknown attacks, because their activities are different from normal ones.

During the last decade, many machine learning and data mining techniques have been applied to IDSs, so that their performance was significantly improved as well as they could be constructed with low cost and effort. Particularly, *unsupervised* anomaly detection techniques(Eskin et al., 2002; Guan et al., 2003; Laskov et al., 2004; Leung & Leckie, 2005; Li et al., 2003; Oldmeadow et al., 2004; Portnoy et al., 2001; Song et al., 2008a; 2009; Wang & Megalooikonomou, 2005) have received remarkable attention, because they are able to construct intrusion detection models without using any labeled training data (*i.e.*, with instances preclassified as being an attack or not) in an automated manner, and they also have intrinsic ability to detect 0-day attacks. Furthermore, considering labeled data or purely normal data cannot be obtained easily in practice, it is better to focus on applying unsupervised anomaly detection techniques to the construction of IDSs than supervised ones. Existing unsupervised anomaly detection techniques have been applied to mainly two types of data sources: raw traffic data and IDS alerts. In the case of approaches based on raw traffic data, they have a strong point compared with another that it is possible to detect all of cyber attacks which are being happened over our networks theoretically, while there is also a fatal problem in that they trigger an unmanageable amount of alerts. In fact, by some estimates, more than thousands of alerts are raised everyday(Manganaris et al., 2000), and about 99% of them is false positive(Julisch, 2003). This situation makes analyst impossible to inspect all of them in time and to identify which alerts are more dangerous. As a result, it is very difficult that IDS operators discover unknown and critical attacks from IDS alerts even if they contain such attacks.

Due to this impracticability of approaches based on raw traffic data, during the last few years, a lot of researchers have focused on mitigation of the amount of false alerts and detection of cyber attacks by analyzing IDS alerts(Bass, 2000; Clifton & Gengo, 2000; Giacinto et al., 2005; Manganaris et al., 2000; Treinen & Thurimella, 2006; Yu & Frincke, 2004; Zurutuza & Uribeetxeberria, 2004). Especially, by analyzing IDS alerts, it is possible to reveal invisible intrusions from them(Song et al., 2007; 2008b), because skillful attackers devise diverse artifice to hide their activities from recent security devices such as IDS, which leads to different combination and/or frequency of alerts from those of well-known attack activities. For example, attackers sometimes try to make IDSs trigger a large amount of alerts intentionally by sending well-crafted packets which have no malicious codes, but they are designed to match some signatures which are defined to detect an outdated attack. In this case, IDS operators are apt to misjudge such events as false positives or unimportant attacks. After that, real attacks are started to the targeted vulnerability because these attacks are no longer considered as suspect activities by IDS operators. Therefore, it is possible to identify something new activities by analyzing patterns of the tricked IDS alerts. Although those approaches based on IDS alerts have a shortcoming that they are only able to detect limited intrusions which could be observed from the IDS alerts, they enable us to discover hidden attacks which are undetectable in raw traffic data and to make the analysis task of IDS alerts more easy.

Considering the above two approaches (*i.e.*, those based on raw traffic data and IDS alerts) have advantages and disadvantage to each other, a hybrid approach for carrying out the correlation analysis between them is essential. In this chapter, we conduct correlation analysis between raw traffic data and IDS alerts using one-class SVM(Li et al., 2003; Schölkopf et al., 2001), focusing on evaluation of *unsupervised anomaly detection*, which is one of the most general and powerful unsupervised machine learning technique.

To this end, we first collected raw traffic data from our honeypots deployed in Kyoto University(Song et al., 2008c), and we extracted 14 statistical features(Benchmark Data, 2010; Song et al., 2009) from them. We also obtained IDS alerts that were recorded by Snort (ver. 4.9.1.4)(Snort, 2010) deployed in front of our honeypots. Snort is an open source network intrusion prevention and detection system (IDS/IPS) developed by Sourcefire(Sourcefire, 2010). Similar to honeypot data, we extracted 7 statistical features from IDS alerts(Song et al., 2008b). We then applied one-class SVM to two data sources, honeypot data and IDS alerts, and consequently obtained two intrusion detection models. With the two intrusion detection models, we investigated what each model detected from our evaluation data and carried out correlation analysis between the intrusions detected from them. Our experimental results for correlation analysis show that it is more useful and practical to integrate the detection results obtained from the two intrusion detection models.

The rest of this chapter is organized as follows. In Section 2, we give a brief description for one-class SVM and previous approaches based on raw traffic data and IDS alerts. In Section 3, we describe our experimental environment and benchmark data (*i.e.*, honeypot data and IDS alerts). In Section 4, we present experimental results obtained from each of two benchmark data and our correlation analysis elicited by combining them. Finally, we present concluding remarks and suggestions for future work in Section 5.

2. Related work

2.1 Intrusion detection using raw traffic data

The earlier methods for intrusion detection were based on intrusion detection rules, i.e., signatures, that are manually constructed by human experts(Sebring et al., 1988). However, since the amount of audit data increases rapidly, their methods consume huge amounts of cost and time to construct the signatures. In addition, these systems can detect only attacks that have been modeled beforehand. In order to cope with the problems, many researchers have applied data mining and machine learning techniques to intrusion detection(Amor et al., 2004; Bridges & Luo, 2000; Lee et al., 1998; 1999). However, there is also a problem that construction of intrusion detection models requires labeled training data, i.e., the data must be pre-classified as attack or not. In general, labeled data can not be obtained readily in real environment since it is very hard to guarantee that there are no intrusions when we are collecting network traffic. A survey of these methods is given in (Warrender et al., 1999).

Over the past few years, several studies to solving these problems have been made on anomaly detection using unsupervised learning techniques, called unsupervised anomaly detection, which are able to detect previously “unseen” attacks and do not require the labeled training data used in the training stage(Denning, 1987; Javitz & Valdes, 1993). A clustering method for detecting intrusions was first presented in (Portnoy et al., 2001), without being given any information about classifications of the training data. In (Eskin et al., 2002) Eskin, et al. presented a geometric framework for unsupervised intrusion detection. They evaluated their methods over both network records and system call traces, and showed that their algorithms were able to detect intrusions over the unlabeled data. In (Guan et al., 2003) Guan, et al. proposed a K-means based clustering algorithm, named Y-means, for intrusion detection. Y-means can overcome two shortcomings of the K-means: number of clusters dependency and degeneracy. In (Oldmeadow et al., 2004) Oldmeadow, et al. presented a solution that can automatically accommodate non-stationary traffic distributions, and demonstrated the effectiveness of feature weighting to improve detection accuracy against certain types of attack. In (Laskov et al., 2004) Laskov, et al. proposed a quarter-sphere SVM that is one variant of one-class SVM, with moderate success. In (Leung & Leckie, 2005) Leung, et al. proposed a new density-based and grid-based clustering algorithm, called fpMAFIA, that is suitable for unsupervised anomaly detection. In (Wang & Megalooikonomou, 2005) Wang, et al. proposed a new clustering algorithm, FCC, for intrusion detection based on the concept of fuzzy connectedness. In (Song et al., 2008a), Song, et al. proposed a K-means based clustering algorithm for intrusion detection. The proposed algorithm improves the detection accuracy and reduce the false positive rate by overcoming shortcomings of the K-means in intrusion detection. Also, Song, et al. proposed a new anomaly detection method based on clustering and multiple one-class SVM in order to improve the detection rate while maintaining a low false positive rate(Song et al., 2009).

2.2 Intrusion detection using IDS alerts

As IDS has played a central role as an appliance to effectively defend our crucial computer systems or networks, large organization and companies have deployed different models of IDS from different vendors. Nevertheless, there is a fatal weakness that they trigger an unmanageable amount of alerts. Inspecting thousands of alerts per day and sensor(Manganaris et al., 2000) is not feasible, specially if 99% of them are false positives(Julisch, 2003). Due to this impracticability, during the last few years a lot of researches have been proposed to reduce the amount of false alerts, by studying the cause of these false positives, creating a higher level view or scenario of the attacks, and finally providing a coherent response to attacks understanding the relationship between different alerts(Zurutuza & Uribeetxeberria, 2004).

T. Bass firstly introduced data fusion techniques in military applications for improving performance of next-generation IDS(Bass, 2000). In (Manganaris et al., 2000) Manganaris, et al. analyzed the alerts gathered by real-time intrusion detection systems by using data mining, and characterized the "normal" stream of alerts. In (Clifton & Gengo, 2000) Clifton, et al. also used data mining techniques to identify sequences of alerts that likely result from normal behavior, and then filtered out false positives from original alerts based on them. Yu, et al. proposed a framework for alert correlation and understanding in intrusion detection system. Their experimental results show that their method can reduce false positives and negatives, and provide better understanding of the intrusion progress by introducing confidence scores(Yu & Frincke, 2004). Giacinto, et al. performed alert clustering which produces unified description of attacks from multiple alerts to attain a high-level description of threats(Giacinto et al., 2005). In (Treinen & Thurimella, 2006) Treinen, et al. used meta-alarms to identify known attack patterns in alarm streams, and used association rule mining to shorten the training time.

As mentioned above, a number of approaches have been suggested to effectively manage IDS alerts, but their researches have been limited to reduction in the amount of IDS alerts mainly. However, since unusual behavior of intruders to evade modern well-managed security systems, in many cases it can be identified by analyzing IDS alerts. In (Song et al., 2007), Song, et al. suggested a data mining technique in order to extract unknown activities from IDS alerts. Also, Song, el al. proposed a generalized feature extraction scheme to detect serious and unknown cyber attacks in that new 7 features were extracted by using only the basic 6 features of IDS alerts; detection time, source address and port, destination address and port, and signature name(Song et al., 2008b).

3. Overview

Figure 1 shows the overall architecture of our correlation analysis. In our approach, we first collected traffic data from three different types of networks, *i.e.*, the original campus network of Kyoto University, QGPOP network and IPv6 network. QGPOP network is being provided by Kyushu GigaPOP Project(QGPOP, 2010) which aims to build a dedicated R&D Internet over Kyushu region in parallel with commodity Internet, focusing on Internet's end-to-end principle and new features like IPv6, multicasting, and Mobile IP. We also deployed Snort (ver. 4.9.1.4)(Snort, 2010) at the perimeter of the above three networks and stored IDS alerts recorded by it into our dedicated DB system. In order to decoy attackers into our networks, we deployed many types of honeypots in the internal network and we stored the traffic data observed on the honeypots into our dedicated DB system. We then construct two benchmark data from honeypot data and IDS alerts described in Sections 4.1 and 4.2. Finally, we apply two benchmark data to one-class SVM, respectively, evaluate their performance and carry out the correlation analysis between them.

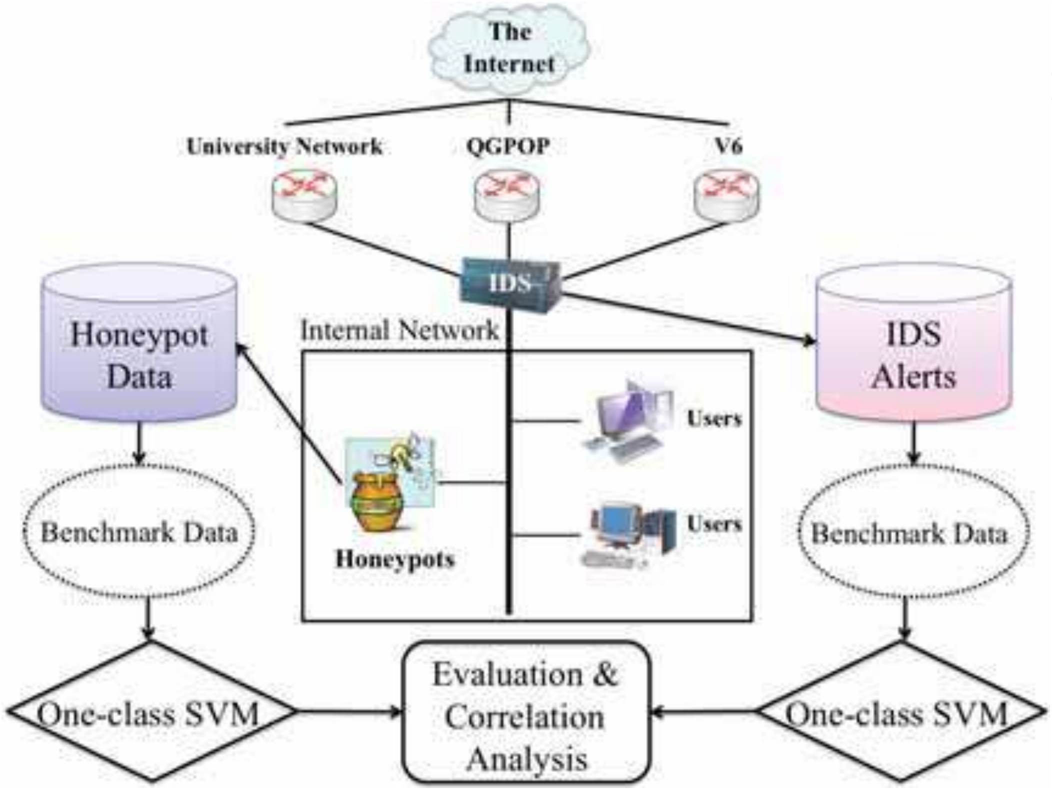


Fig. 1. The overall architecture of our correlation analysis.

4. Benchmark data

4.1 Honeypot data

In intrusion detection field, KDD Cup 1999 dataset(KDD Cup 99', 1999) has been used for a long time as benchmark data for evaluating performance of IDSs. However, there is a fatal drawback in that KDD Cup 1999 dataset is unable to reflect current network situations and latest attack trends, because it was generated by simulation over the virtual network more than 10 years ago, and thus its attack types are greatly old-fashioned. In spite of this drawback, researchers have used it as their evaluation data, because it is quite difficult to get high-quality evaluation data due to privacy and competitive issues: many organizations scarcely share their data with other institutions and researchers. To make matters worse, labeling traffic data as either normal or intrusion requires enormous amount of time for many human experts, even if real traffic data is available.

In order to provide more practical and useful evaluation results, it is needed to carry out our experiments using real traffic data. In (Song et al., 2008c), we deployed several types of honeypots over the 5 different networks which are inside and outside of Kyoto University: 1 class A and 4 class B networks. For example, there are some Windows base honeypots (*e.g.* with Windows XP with SP2, full patched Window XP, Windows XP without any patch, Windows Vista), and Symantec honeypot with Solaris, network printer, home appliance, *e.g.*, TV, Video Recorder and so on. In addition to traditional honeypots which only receive attacks, we deployed proactive systems which access to malicious web servers and join real botnets to get various types of commands. We collected all traffic data to/from our honeypots, and observed that most of them consist of attack data. In fact, for the collected traffic data, we carried out a deep inspection for every connection if there was a buffer overflow attack or not. In order to identify a shellcode and an exploit code from traffic data, we used the dedicated detection software(Ashula, 2010). We also used IDS alerts obtained from Snort (ver. 4.9.1.4)(Snort, 2010) and malware information obtained from ClamAV(Clamav, 2010) as extra information for inspecting traffic data. By using these diverse information, we thoroughly inspected the collected traffic data, and identified what happened on the networks.

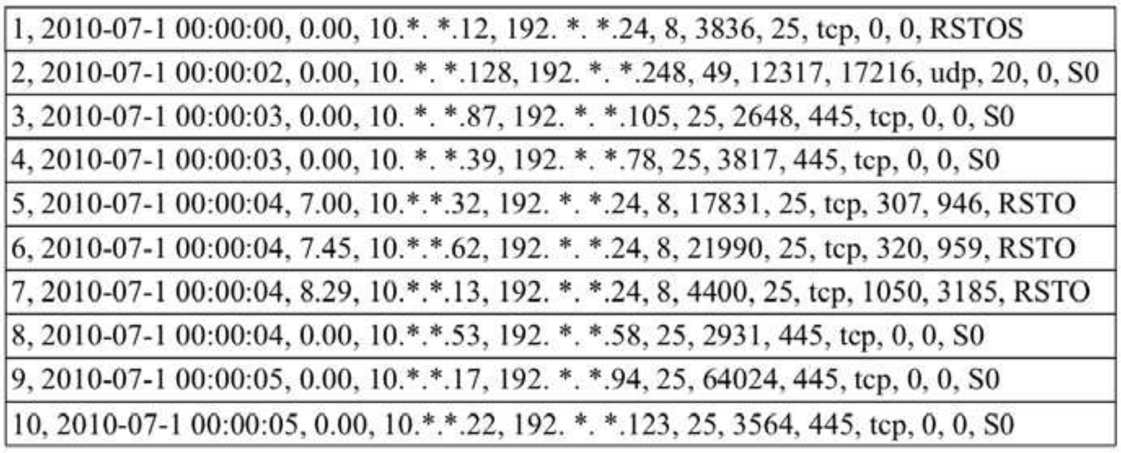
In spite of our effort for inspecting real attacks on the campus networks, there is still a possibility that unidentified attacks are being contained in the honeypot traffic data. However, in our investigation, we observed that most of honeypot traffic data captured in our honeypots are composed of attack data and there were few unidentified traffic data. Therefore, all the original honeypot traffic data are regarded as attack data in our benchmark data, because performance of one-class SVM is almost unaffected by a small amount of unidentified attack data or they can be treated as just noisy data.

On the other hand, since the most of the honeypot traffic data consist of attack data, we should prepare a large amount of normal data in order to evaluate performance of one-class SVM effectively. In order to generate normal traffic data, we deployed a mail server on the same network with honetpots, and regarded its traffic data as normal data. The mail server was also operated with several communication protocols, *e.g.*, ssh, http and https, for its management and also received various attacks. Although all of these activities were included with the traffic data, they do not affect to performance of machine learning techniques considered in our experiments due to their small amount.

4.1.1 Extracting 14 statistical features

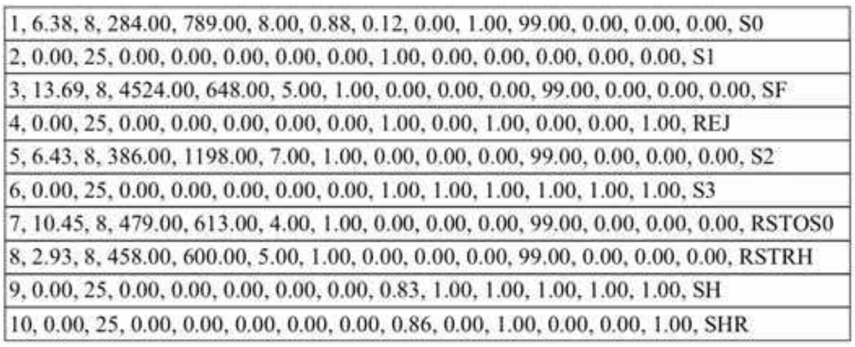
Among the 41 original features of KDD Cup 99 data set, we have extracted only 14 significant and essential features (*e.g.*, Figure 3) from traffic data (*e.g.*, Figure 2) of honeypots and a mail server, and we used 13 continuous features excluding one categorical feature (*i.e.*, “flag”) for our evaluation data. The reason why we extracted only the 14 statistical features is that among the original 41 features of the KDD Cup 99 dataset(KDD Cup 99’, 1999) there exist substantially redundant and insignificant features. Our benchmark data is open to the public at (Benchmark Data, 2010). Visit our web site for more detail.

1. **Duration**: the length (number of seconds) of the connection
2. **Service**: the connection’s service type, e.g., http, telnet, etc
3. **Source bytes**: the number of data bytes sent by the source IP address
4. **Destination bytes**: the number of data bytes sent by the destination IP address
5. **Count**: the number of connections whose source IP address and destination IP address are the same to those of the current connection in the past two seconds.
6. **Same_srv_rate**: % of connections to the same service in Count feature
7. **Serror_rate**: % of connections that have “SYN” errors in Count feature
8. **Srv_serror_rate**: % of connections that have “SYN” errors in Srv_count(the number of connections whose service type is the same to that of the current connection in the past two seconds) feature



```
1, 2010-07-1 00:00:00, 0.00, 10.*. *.12, 192. *. *.24, 8, 3836, 25, tcp, 0, 0, RSTOS
2, 2010-07-1 00:00:02, 0.00, 10. *. *.128, 192. *. *.248, 49, 12317, 17216, udp, 20, 0, S0
3, 2010-07-1 00:00:03, 0.00, 10. *. *.87, 192. *. *.105, 25, 2648, 445, tcp, 0, 0, S0
4, 2010-07-1 00:00:03, 0.00, 10. *. *.39, 192. *. *.78, 25, 3817, 445, tcp, 0, 0, S0
5, 2010-07-1 00:00:04, 7.00, 10.*.*.32, 192. *. *.24, 8, 17831, 25, tcp, 307, 946, RSTO
6, 2010-07-1 00:00:04, 7.45, 10.*.*.62, 192. *. *.24, 8, 21990, 25, tcp, 320, 959, RSTO
7, 2010-07-1 00:00:04, 8.29, 10.*.*.13, 192. *. *.24, 8, 4400, 25, tcp, 1050, 3185, RSTO
8, 2010-07-1 00:00:04, 0.00, 10.*.*.53, 192. *. *.58, 25, 2931, 445, tcp, 0, 0, S0
9, 2010-07-1 00:00:05, 0.00, 10.*.*.17, 192. *. *.94, 25, 64024, 445, tcp, 0, 0, S0
10, 2010-07-1 00:00:05, 0.00, 10.*.*.22, 192. *. *.123, 25, 3564, 445, tcp, 0, 0, S0
```

Fig. 2. Example of session data.



```
1, 6.38, 8, 284.00, 789.00, 8.00, 0.88, 0.12, 0.00, 1.00, 99.00, 0.00, 0.00, 0.00, S0
2, 0.00, 25, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00, 0.00, 0.00, S1
3, 13.69, 8, 4524.00, 648.00, 5.00, 1.00, 0.00, 0.00, 0.00, 99.00, 0.00, 0.00, 0.00, SF
4, 0.00, 25, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 1.00, 0.00, 0.00, 1.00, REJ
5, 6.43, 8, 386.00, 1198.00, 7.00, 1.00, 0.00, 0.00, 0.00, 99.00, 0.00, 0.00, 0.00, S2
6, 0.00, 25, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, S3
7, 10.45, 8, 479.00, 613.00, 4.00, 1.00, 0.00, 0.00, 0.00, 99.00, 0.00, 0.00, 0.00, RSTOS0
8, 2.93, 8, 458.00, 600.00, 5.00, 1.00, 0.00, 0.00, 0.00, 99.00, 0.00, 0.00, 0.00, RSTRH
9, 0.00, 25, 0.00, 0.00, 0.00, 0.00, 0.00, 0.83, 1.00, 1.00, 1.00, 1.00, 1.00, SH
10, 0.00, 25, 0.00, 0.00, 0.00, 0.00, 0.00, 0.86, 0.00, 1.00, 0.00, 0.00, 1.00, SHR
```

Fig. 3. Example of 14 statistical features.

9. **Dst_host_count**: among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose source IP address is also the same to that of the current connection.
10. **Dst_host_srv_count**: among the past 100 connections whose destination IP address is the same to that of the current connection, the number of connections whose service type is also the same to that of the current connection
11. **Dst_host_same_src_port_rate**: % of connections whose source port is the same to that of the current connection in Dst_host_count feature
12. **Dst_host_serror_rate**: % of connections that have "SYN" errors in Dst_host _count feature
13. **Dst_host_srv_serror_rate**: % of connections that "SYN" errors in Dst_host_ srv_count feature
14. **Flag**: the state of the connection at the time the summary was written (which is usually when the connection terminated). The different states are summarized in the below section.

4.1.2 Example of session data and their 14 statistical features

Figures 2 and 3 show examples of session data captured in our honeypots and their 14 statistical features, respectively. Note that we sanitized source and destination IP addresses because of secrecy of communication. 192.x.x.x indicates sanitized internal IP address and 10.x.x.x indicate sanitized external IP address. In Figure 2, each row indicates one session data, and it consists of 12 columns: ID, time, duration, source IP address, destination IP address, service type (*e.g.*, 8 represents HTTP), source port number, destination port number, protocol, source bytes, destination bytes, flag.

Feature name	Value
Duration	6.38 seconds
Service	8 (*i.e.*, HTTP)
Source bytes	284 bytes
Destination bytes	789 bytes
Count	8
Same_srv_rate	88%
Serror_rate	12 %
Srv_serror_rate	0%
Dst_host_count	1
Dst_host_srv_count	99
Dst_host_same_src_port_rate	0%
Dst_host_serror_rate	0%
Dst_host_srv_serror_rate	0%
Flag	S0

Table 1. Values of 14 statistical features in line 1.

In Figure 3, each row indicates one session data, and it consists of 15 columns, *i.e.*, one ID and 14 statistical features. For example, Table 1 shows the values of the 14 statistical features in line 1.

4.2 IDS alerts

In our previous work, we introduced a feature extraction scheme so that one can detect 0-day attack from IDS alerts(Song et al., 2007). In the feature extraction scheme, it uses "Incident ID" feature among the original features of IDS alerts that were recorded by SNS7160 IDS system(SNS7160, 2010). The Incident ID feature indicates a group of IDS alerts that are considered as correlated attack by SNS7160 IDS system. Hence, if two alerts contain the same Incident ID, then they become members of the same group. However, there is a problem that not all vendors provide the Incident ID feature in their IDS product. Furthermore, even if it provided, its building mechanism is different from each other.

On the other hand, in recent years, many organizations (*e.g.*, ISAC(REN-ISAC, 2010; TELECOM-ISAC, 2010)) are getting started to share their security information with others in order to improve network security. However, these organizations deploy various types of security devices such as IDSs, firewalls and so on. In addition, if we force them to utilize the same product, its weakness causes signs of many 0-day attacks invisible. Because of this, we need to devise a mechanism to integrate their information effectively. However, if we utilize only the common features of IDS such as IP address, port, detection time and so on, it is not enough to extract useful information from them. In order to satisfy these requirements, in (Song et al., 2008b) we extracted 7 statistical features (see subsection 4.2.1) using only the basic 6 features of IDS alerts: detection time, source address and port, destination address and port, and signature name.

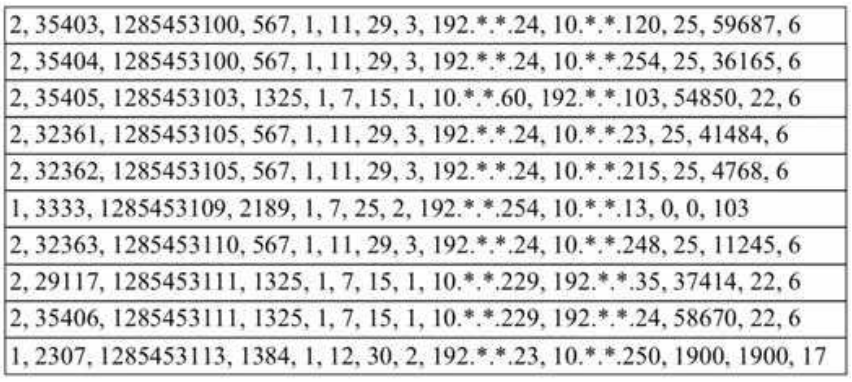
To obtain IDS alerts, we used Snort (ver. 4.9.1.4)(Snort, 2010) that is actually deployed at perimeter of Kyoto University. Snort is charged with protecting 2 class B and 4 class C networks. If Snort observes a suspicious session on the networks, it triggers a corresponding alert according to its detection engine. Note that we have obtained IDS alerts that were triggered by SNS7160 IDS system on our experimental network described in Figure 1 since 2006, and we have started to deploy Snort into our network since 2010. Thus, it is possible to extract the 7 statistical features from both IDS alerts (*i.e.*, Snort and SNS7160 IDS), because the 7 statistical features can be extracted from only the basic 6 features. Since Snort is a free software, we use its alerts as our evaluation data in this chapter.

4.2.1 Extracting 7 statistical features

From the IDS alerts of Snort, we extracted the 7 statistical features (Figures 4) from each alert (Figures 5). Note that the following features refer to the last N alerts whose source address and destination port number are the same to the current alert.

1. **NUM_SAME_SA_DA_DP**
   Among N alerts, the number of alerts whose destination address is the same to the current alert. We define them as n alerts.
2. **RATE_DIFF_ALERT_SAME_SA_DA_DP**
   Rate of the number of alerts whose alert types are different from the current alert to n.
3. **TIME_STDDEV_SAME_SA_DA_DP**
   Standard deviation of the time intervals between each instance of n alerts including the current alert.
4. **NUM_SAME_SA_DP_DIFF_DA**
   Among N alerts, the number of alerts whose destination address is different from the current alert; it becomes $(N - n)$.
5. **RATE_DIFF_ALERT_SAME_SA_DP_DIFF_DA**
   Rate of the number of alerts whose alert types are different from the current alert to $(N - n)$.
6. **TIME_STDDEV_SAME_SA_DP_DIFF_DA**
   Standard deviation of the time intervals between each instance of $(N - n)$ alerts including the current alert.
7. **RATE_REVERSE_SP_SAME_SA_DP**
   Rate of the number of the alerts whose source port is the same or larger than that of the current alert to N.

NUM_SAME_SA_DA_DP and NUM_SAME_SA_DP_DIFF_DA features represent that an attacker tries to exploit just one victim host and a lot of victim hosts, respectively. RATE_DIFF_ALERT_SAME_SA_DA_DP and RATE_DIFF_ALERT_SAME_SA_DP_DIFF_DA features indicate that if there is happening a real attack on the network, it sometimes raises several different kinds of IDS alerts. TIME_STDDEV_SAME_SA_DA_DP and TIME_STDDEV_SAME_SA_DP_DIFF_DA features are based on the fact that in the case of real attacks, since the time intervals between each alert triggered by IDS are very long and unpredictable, the values of these features will increase. RATE_REVERSE_SP_SAME_SA_DP feature means that if several successive sessions are made from a certain host, their source port numbers also increase automatically. Note that the values of the above 7 features have ‘0’, if the number of the corresponding IDS alerts does not exceed N.

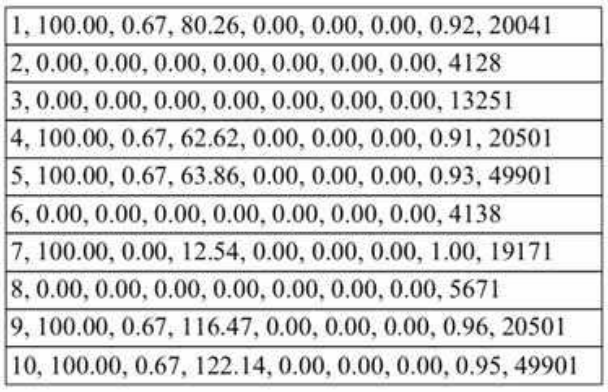


```
2, 35403, 1285453100, 567, 1, 11, 29, 3, 192.*.*.24, 10.*.*.120, 25, 59687, 6
2, 35404, 1285453100, 567, 1, 11, 29, 3, 192.*.*.24, 10.*.*.254, 25, 36165, 6
2, 35405, 1285453103, 1325, 1, 7, 15, 1, 10.*.*.60, 192.*.*.103, 54850, 22, 6
2, 32361, 1285453105, 567, 1, 11, 29, 3, 192.*.*.24, 10.*.*.23, 25, 41484, 6
2, 32362, 1285453105, 567, 1, 11, 29, 3, 192.*.*.24, 10.*.*.215, 25, 4768, 6
1, 3333, 1285453109, 2189, 1, 7, 25, 2, 192.*.*.254, 10.*.*.13, 0, 0, 103
2, 32363, 1285453110, 567, 1, 11, 29, 3, 192.*.*.24, 10.*.*.248, 25, 11245, 6
2, 29117, 1285453111, 1325, 1, 7, 15, 1, 10.*.*.229, 192.*.*.35, 37414, 22, 6
2, 35406, 1285453111, 1325, 1, 7, 15, 1, 10.*.*.229, 192.*.*.24, 58670, 22, 6
1, 2307, 1285453113, 1384, 1, 12, 30, 2, 192.*.*.23, 10.*.*.250, 1900, 1900, 17
```

Fig. 4. Examples of IDS Alerts

sensor_ID	Identification of each IDS
event_ID	Identification of each event
event_sec	UNIX time when the corresponding event was detected
sig_ID	Identification of each signature
gen_ID	Identification of each detection engine
rev	Revision of each signature
class	Attack types
priority	severity of each alert (1: high, 2: medium, 3: low)
src_address	source address (sanitized)
dst_address	destination address (sanitized)
src_port	source port number
dst_port	destination port number
ip_protocol	TCP, UDP, ICMP and so on

Table 2. Description of each column in IDS alerts.



```
1, 100.00, 0.67, 80.26, 0.00, 0.00, 0.00, 0.92, 20041
2, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 4128
3, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 13251
4, 100.00, 0.67, 62.62, 0.00, 0.00, 0.00, 0.91, 20501
5, 100.00, 0.67, 63.86, 0.00, 0.00, 0.00, 0.93, 49901
6, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 4138
7, 100.00, 0.00, 12.54, 0.00, 0.00, 0.00, 1.00, 19171
8, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 5671
9, 100.00, 0.67, 116.47, 0.00, 0.00, 0.00, 0.96, 20501
10, 100.00, 0.67, 122.14, 0.00, 0.00, 0.00, 0.95, 49901
```

Fig. 5. Example of the extracted 7 statistical features

4.2.2 Example of IDS alerts and their 7 statistical features

Figures 4 and 5 show examples of the alerts recorded by Snort and their 7 statistical features, respectively. Similar to the honeypot data, we sanitized source and destination IP addresses. 192.x.x.x indicates sanitized internal IP address and 10.x.x.x indicate sanitized external IP address. In Figure 4, each row indicates one alert, and it consists of 13 columns. The meaning of each column is described in Table 2.

Feature name	Value
NUM_SAME_SA_DA_DP	100
RATE_DIFF_ALERT_SAME_SA_DA_DP	67%
TIME_STDDEV_SAME_SA_DA_DP	80.26
NUM_SAME_SA_DP_DIFF_DA	0
RATE_DIFF_ALERT_SAME_SA_DP_DIFF_DA	0%
TIME_STDDEV_SAME_SA_DP_DIFF_DA	0%
RATE_REVERSE_SP_SAME_SA_DP	92%

Table 3. Values of 7 statistical features.

In Figure 5 each row indicates one alert, and it consists of 9 columns, *i.e.,*, one ID, 7 statistical features and alert type. For example, Table 3 shows the values of the 7 statistical features in line 1 in that we set N to 100. Note that 8th column represents identification of each alert.

5. One-class SVM

In our correlation analysis, we apply one-class SVM to two types of benchmark data, *i.e.*, honeypot data and IDS alerts, in order to detect cyber attacks from them. Support Vector Machines(SVM)(Vapnik, 1995; 1998) have received great interest and have been one of the most developed machine learning techniques. Some reasons why SVM has been succeeded in many applied applications are their good theoretical properties in generalization and convergence(Cristianini & Shawe-Yaylor, 2000). Another reason is their excellent performance in some hard problems(Dumais et al., 1998; Osuna et al., 1997).

One-class SVM(Schölkopf et al., 2001) is one of the extension of the binary SVM(Vapnik, 1995; 1998), which is based on unsupervised learning paradigms. Given the unlabeled l data points, $\{x_1, \ldots . x_l\}$ where $x_i \in R^n$; one-class SVM is to map the data points x_i into the feature space by using some non-linear mapping $\Phi(x_i)$, and to find a hypersphere which contains most of the data points in the feature space. Figure 6 shows the geometry of the hypersphere where it is formulated with the center c and the radius R in the feature space. Therefore, in intrusion detection field, the data points that belong to the outside of the hypersphere can be regarded as anomalies(*i.e.*. potential cyber attacks) because there are a few attacks in traffic data and IDS alerts, and most of them are usual false positives.

Mathematically, the problem of searching such a hypersphere is formulated as follows:

$$\begin{aligned} \min_{R\in\Re,\xi\in\Re^l,c\in\mathcal{F}} \quad & R^2 + \frac{1}{vl}\sum_{i=1}^{l}\xi_i, \\ \textbf{subject to}: & ||\Phi(x_i) - c|| \leq R^2 + \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l. \end{aligned} \tag{1}$$

The non-negative slack variables ξ_i allow that some points belong to the "wrong" side of the hypersphere. Also, the parameter $v \in [0, 1]$ determines the trade off between the radius of the hypersphere (*i.e.*, its size) and the number of the data points that belong to the hypersphere. When v is small, more data are put into the hypersphere. When v is larger, its size decreases. Since the center c belongs to the possibly high-dimensional feature space, it is difficult to solve the primal problem (1) directly. Instead of the primal problem (1), it is possible to the primal

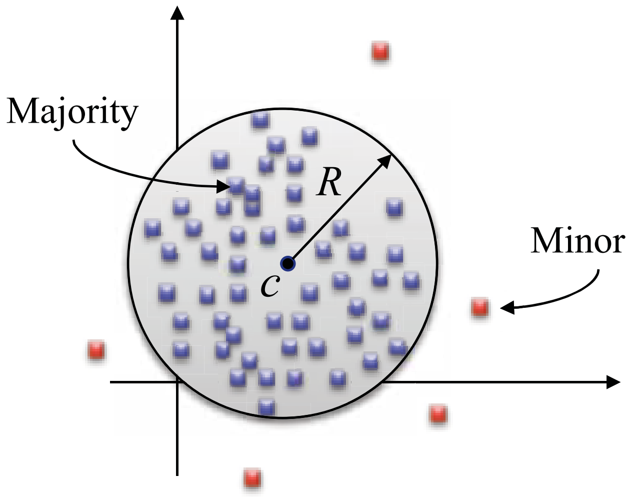


Fig. 6. The geometry of the sphere formulation of one-class SVM.

problem by its dual form with kernel functions, $k(x, y)$:

$$\begin{aligned} \min_{\alpha \in \Re^l} \quad & \sum_{i,j=1}^{l} \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^{l} \alpha_i k(x_i, x_i) \\ \textbf{subject to} : & \sum_{i=1}^{l} \alpha_i = 1, \\ 0 \leq \alpha_i \leq & \frac{1}{vl}, i = 1, \ldots, l. \end{aligned} \tag{2}$$

If we find a hypersphere from the training data, we can classify the testing data as either normal or attack using the hypersphere. In this classification, the following decision function, whether point x in the testing data is normal(*i.e.*, inside of the hypersphere), is used:

$$\begin{aligned} f(x) = \textbf{sgn}\Big(& R^2 - \sum_{i,j=1}^{l} \alpha_i \alpha_j k(x_i, x_j) \\ & + 2 \sum_{i} \alpha_i k(x_i, x) - k(x, x) \Big). \end{aligned} \tag{3}$$

The points with $f(x) = -1$ are considered to be anomalies because this means that they exist outside of the hypersphere. Otherwise they are considered to be normal, because they are members of the hypersphere. In our correlation analysis, we used LIBSVM library(Chang & Lin, 2001) to carry out the experiments with one-class SVM.

6. Experimental results and their analysis

6.1 Preprocessing and data preparation

In our experiments, we used the traffic data and IDS alerts of a day (Jul. 3rd, 2010) as our training data and they have contained 496,090 session data and 35,195 IDS alerts, respectively. Also, in the case of the traffic data, the ratio of attack data occupied 80% of the original traffic data. In case of real network, however, there are a few attack data or really dangerous attack data in its traffic data. Because of this, we adjusted the ratio of attack data to 1% and consequently we obtained 115,509 session data that were randomly and fairly selected from the original training data and regarded them as our training data. On the other hand, we regarded the original IDS alerts as our training data, because in IDS alerts, our goal is just to identify more serious and dangerous attacks from them. As the testing data, we used the traffic data and the IDS alerts of 4 days: Jul. 5th, 12th, 25th and 29th, 2010.

6.2 Evaluation process

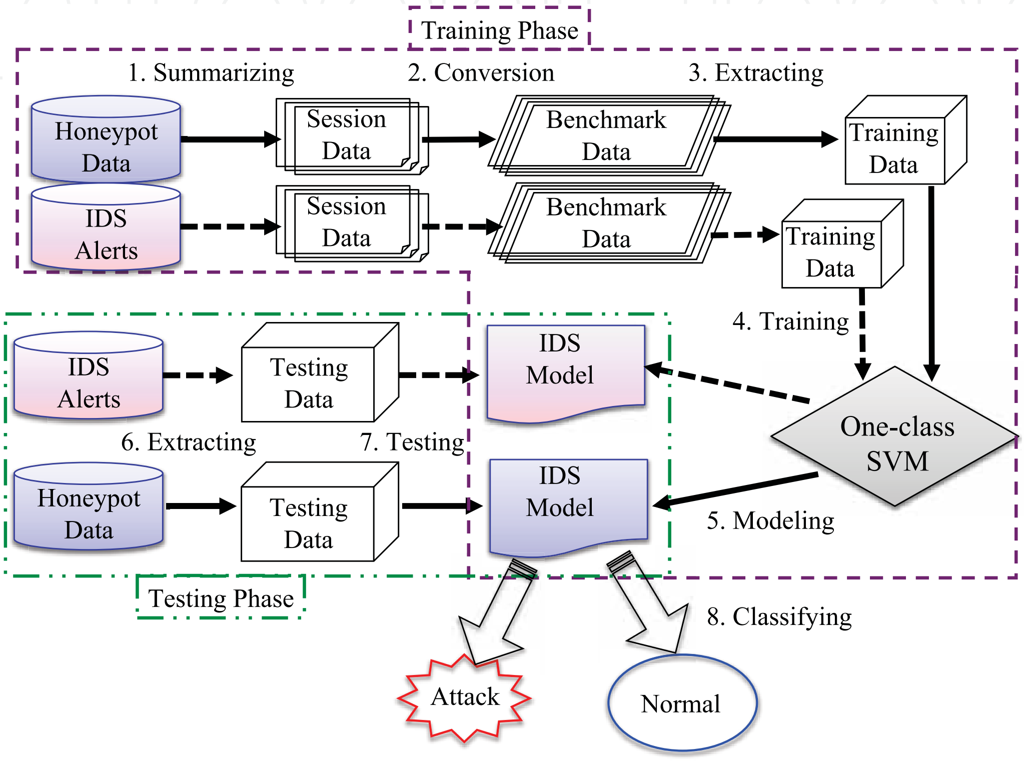


Fig. 7. The overall process of the training and testing phases.

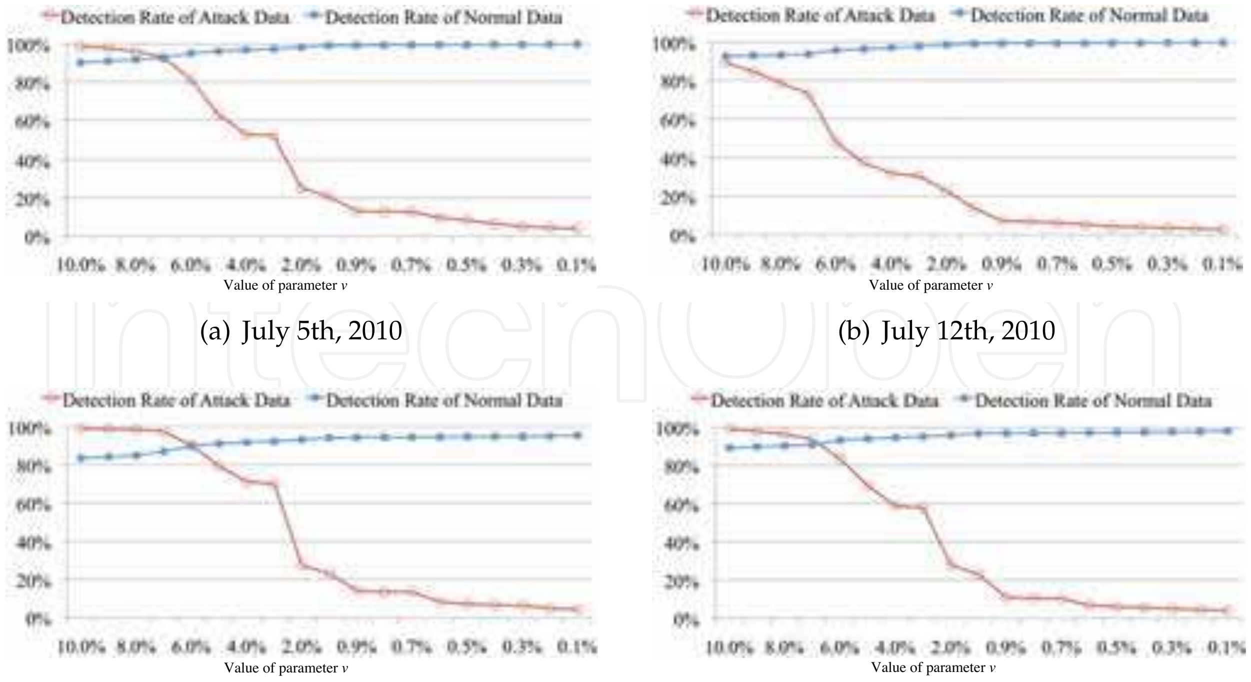
Figure 7 shows the overall process of correlation analysis between two types of evaluation data: honeypot data and IDS alerts. The evaluation process is composed of two phases: training phase and testing phase. The training phase is summarized as follows.

1. **Summarizing**: summarize collected raw traffic data and IDS alerts in session data as described in subsections 4.1.2 and 4.2.2. Especially, in the case of honeypot data, we used BroBro (2010) for this summarizing process.
2. **Conversion**: convert summarized session data of traffic data and IDS alerts into connection records which consist of 14 statistical features and 7 statistical features as described in subsections 4.1.1 and 4.2.1, respectively.
3. **Extracting**: build the training data from converted two types of connection records: honeypot data and IDS alerts. Note that the ratio of attack data to normal data is 1% in the training data of honeypot data and we set the parameter N, which is described in subsection 4.2.1, to 100.
4. **Training and Modeling**: apply two types of benchmark data to one-class SVM, and thus we obtain two IDS models.

In the testing phase, we applied the two processes, *i.e.*, 6 and 7 in Figure 7, which are the same to those of the training phase to the traffic data and IDS alerts of 4 days, and consequently obtained two types of connection records with 14 statistical features and 7 statistical features. After that, we fed two types of connection records of the testing data into the corresponding IDS model which was built in the training phase, and then we evaluated each IDS model according to their detection results.

6.3 Analysis results of honeypot data

In our experiments, we first evaluated performance of one-class SVM using honeypot data. In our performance evaluation, we varied the parameter, v, of one-class SVM. The parameter v represents the ratio of data points which are located outside of the hypersphere discovered by one-class SVM. In other words, if v is smaller (or larger), then number of data points which are inside the hypersphere increases (or decreases). Figure 8 shows the evaluation results of one-class SVM where we set the value of parameter v to 0.1% $\sim$ 10%. In Figure 8, x-axis indicates the values of the parameter v and y-axis indicates the detection rate and the false positive rate of one-class SVM. In our investigation, we observed that the optimized value of the parameter was 6% $\sim$ 10% for each testing data. Table 4 shows the best detection rate and the false positive rate. From Table 4, we can see that there are too many false positives: the number of false positives in four testing data is 7,833, 5,512, 74,463 and 6,772, even if the detection rate is around 90%. Note that in real traffic data, the number of false positives will be extremely larger than that of our honeypot data, because the ratio of attack data in our testing data was about 80%. In other words, since it is obvious that there are much more normal data in real traffic data, the number of false positives will also increase according to the amount of normal data. Therefore, it is needed to minimize those false positives so that network operators can identify more serious and dangerous attacks effectively.



(a) July 5th, 2010

(b) July 12th, 2010

(c) July 25th, 2010

(d) July 29th, 2010

Fig. 8. Performance of one-class SVM by honeypot data.

Date	v	True positive rate	False positive rate
Jul. 5th, 2010	7%	92.91% (316,134/340,235)	7.33% (7,833/106,983)
Jul. 12th, 2010	10%	89.39% (340,180/380,541)	7.41% (5,512/74,436)
Jul. 25th, 2010	6%	90.19% (268,881/298,124)	10.01% (11,653/116,496)
Jul. 29th, 2010	7%	93.54% (200,408/214,247)	9.10% (6,772/74,451)

Table 4. Best true positive rate and false positive rate.

6.4 Analysis results of IDS alerts

In this experiment, we evaluated performance of one-class SVM using IDS alerts. Since there is no label information, we cannot obtain ROC curve(Lippmann, 2000) like Figure 8. However, as mentioned in Section 1, it is possible to reveal unknown attacks by identifying unusual patterns of IDS alerts, even if most of them are false positives(Julisch, 2003), and we demonstrated it in our previous research(Song et al., 2007; 2008b). Therefore, in our evaluation, we call the IDS alerts detected by one-class SVM as "dubious" alerts, and the others as "trivial" alerts.

In our experiments, we first investigated how many real attack data and real normal data are included in the dubious alerts. Figure 9 shows the classification results. In Figure 9, x-axis indicates the values of the parameter v and y-axis indicates the number of real attack data and normal data which belong to the dubious alerts. From Figure 9, we can observe that the number of attack data and normal data has the similar distribution: if the number of attack data increases, the number of normal data also increases.

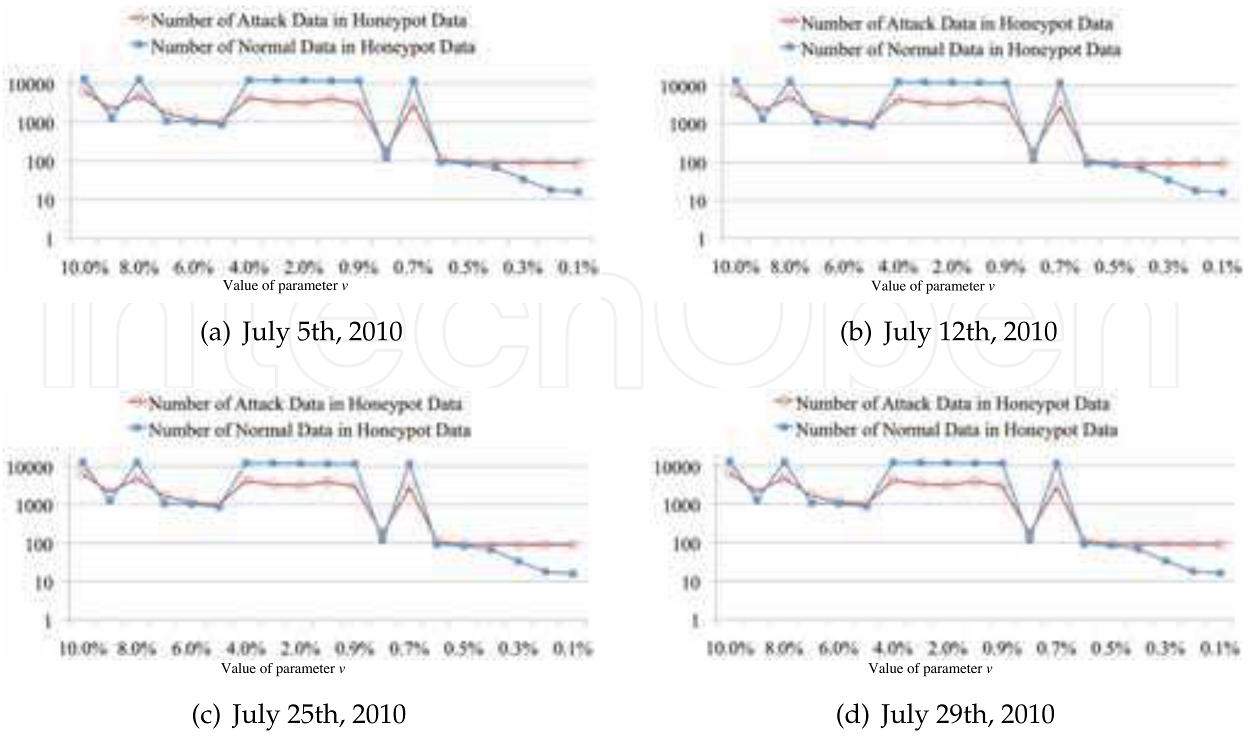
In general, there is a few attacks (less than 100 attacks in many cases) which are serious and dangerous on the certain organization network. From viewpoint of this, it could be said that the optimized value of the parameter v is 0.1%, because the number of the dubious alerts which were detected by one-class SVM is smallest. In fact, Table 5 shows the number of real attack data and normal data when v was 0.1%. However, it is obvious that we need to improve performance of one-class SVM, because there still exist some trivial alerts.

Date	v	Number of real attack data	Number of real normal data
Jul. 5th, 2010	0.1%	92	16
Jul. 12th, 2010	0.1%	195	15
Jul. 25th, 2010	0.1%	365	96
Jul. 29th, 2010	0.1%	308	7

Table 5. Number of real attack data and normal data when v is 0.1%.

6.5 Results of correlation analysis

In order to demonstrate the effectiveness and the necessity of correlation analysis between traffic data and IDS alerts, we conducted the following two experiments. Figure 10 shows the overall process of our correlation analysis. Firstly, we investigated the number of session data ('C' marked in Table 6) that are real attacks in the honeypot data and are not members of the IDS alerts (①). From Table 6, it is obvious that there are lots of real attacks which were not observed in the IDS alerts. For example, in the case of the testing data of Jul. 5th, 2010, it contained 447,217 session data which consist of 340,235 attack data and 106,983 normal data, and among 340,235 attack data, 323,559 attack data were not identified in the IDS alerts. This means that it is essential to analyze traffic data and it is not enough to analyze only the IDS alerts.



(a) July 5th, 2010

(b) July 12th, 2010

(c) July 25th, 2010

(d) July 29th, 2010

Fig. 9. Performance of one-class SVM by IDS alerts.

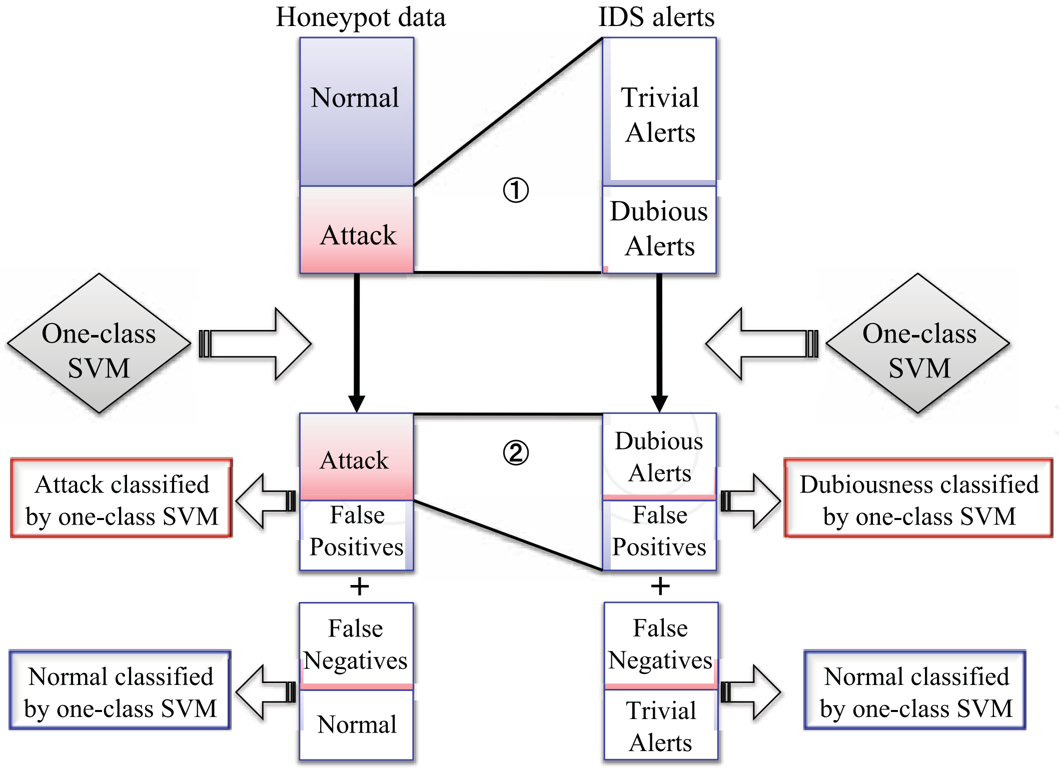


Fig. 10. The overall process of correlation analysis.

Secondly, we compared the real attacks detected from the honeypot data with the dubious alerts (②). In this experiment, we first counted the number of attack data ('D' marked in

Date	Total number of session data	Total number of attack data	C
Jul. 5th, 2010	447,218	340,235	323,559
Jul. 12th, 2010	454,977	380,541	363,831
Jul. 25th, 2010	414,620	298,124	272,938
Jul. 29th, 2010	288,698	214,247	200,311

Table 6. Results of correlation analysis between the original honeypot data and the original IDS alerts.

Table 7) which were detected from the two benchmark data at the same time. From Table 7, we can see that among 108 dubious alerts which were detected from the original IDS alerts, only 40 alerts were also observed from the real attacks detected from the honeypot data. This means that if we analyze only traffic data, it is unable to detect 68 real attacks which could be detected by analyzing the IDS alerts. After all, those results show that we need to analyze not only traffic data, but also IDS alerts, and to carry out correlation analysis between them so that network operators are able to identify more serious and dangerous cyber attack effectively.

Date	v	Number of the dubious alerts	D
Jul. 5th, 2010	7%	108	40

Table 7. Results of correlation analysis between the real attacks detected from the honeypot data and attack data detected from the IDS alerts.

7. Conclusion

In this chapter, we have carried out correlation analysis between honeypot data and IDS alerts. To this end, we first collected raw traffic data from our honeypots(Song et al., 2008c), and we extracted 14 statistical features(Benchmark Data, 2010; Song et al., 2009) from them as described in subsection 4.1. We also captured IDS alerts that were recorded by Snort (ver. 4.9.1.4)(Snort, 2010) deployed in front of our honeypots. Similar to honeypot data, we extracted 7 statistical features from IDS alerts(Song et al., 2008b) as described in subsection 4.2. We then applied one-class SVM to two benchmark data, *i.e.*, honeypot data and IDS alerts, and consequently obtained two intrusion detection models. With the two intrusion detection models, we evaluated each benchmark data, conducted correlation analysis between two benchmark data. Our experimental results show that it is more useful and practical to integrate the detection results obtained from the two intrusion detection models.

8. References

Allen, J.; Christie, A. & Fithen, W., (2000). State of the Practice of Intrusion Detection Technologies, Technical Report, CMU/SEI-99-TR-028, 2000.

Amor, N. B.; Benferhat, S. & Elouedi, Z., (2004). Naive Bayes vs decision trees in intrusion detection systems, *Proc. 2004 ACM Symp. on Applied Computing*, pp. 420-424, 2004.

Ashula, (2010). URL: *http://www.secure-ware.com/contents/product/ashula.html*

Base, R. & Mell, P., (2001). Intrusion Detection Systems, *NIST Special Publications*, November, 2001, SP 800-31.

Bass, T., (2000). Intrusion detection systems and multisensor data fusion, *Communications of the ACM*, ACM Press, pp. 99-105, New York, NY, USA, 2000.

Benchmark Data, (2010). URL: *http://www.takakura.com/Kyoto_data/*

Luo, J. & Bridges, S. M., (2000). Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection, *International Journal of Intelligent Systems*, pp. 687-703, 2000.

Bro, (2010). URL: *http://www.bro-ids.org/*

Chang, C.-C. & Lin, C.-J., (2001). Libsvm: a library for support vector machines, URL: *http://www.csie.ntu.edu.tw/ cjlin/libsvm*

Clamav, (2010). URL: *http://www.clamav.net/*

Clifton, C. & Gengo, G., (2000). Developing custom intrusion detection filters using data mining, *21st Century Military Communications Conference Proceedings(MILCOM2000)*, Vol. 1, pp. 440-443, Los Angeles, California, USA, 2000.

Cristianini, N., & Shawe-Taylor, J., (2000). An introduction to support vector machines, *Cam bridge University Press*, 2000.

Denning, D. E., (1987). An intrusion detection model, *IEEE Transactions on Software Engineering*, 1987, SE-13:222-232.

Dumais, S.; Platt, J.; Heckerman, D. & Sahami, M., (1998). Inductive learning algorithms and representations for text categorization, *7th International Conference on Information and Knowledge Management, ACM-CIKM98*, pp.148-155, 1998.

Eskin, E.; Arnold, A.; Prerau, M.; Portnoy, L. & Stolfo, S., (2002). A Geometric Framework for Unsupervised Anomaly Detection : Intrusion Detection in Unlabeled Data, In *Applications of Data Mining in Computer Security*, 2002.

Giacinto, G.; Perdisci, R. & Roli, F., (2005). Alarm Clustering for Intrusion Detection Systems in Computer Networks, *MLDM 2005*, LNAI 3587, pp. 184-193, 2005.

Guan, Y.; Ghorbani, A. & Belacel, N., (2003). Y-means : A Clustering Method for Intrusion Detection, In *IEEE Canadian Conference on Electrical and Computer Engineering, Proceedings*, 2003.

Javitz, H. S. & Valdes, A., (1993). The NIDES statistical component: description and justification, *Technical Report, Computer Science Laboratory, SRI International*, 1993.

Julisch, K., (2003). Clustering Intrusion Detection Alarms to Support Root Cause Analysis, *ACM Transactions on Information and System Security* 6(4), ACM Press, pp. 443-471, 2003.

KDD Cup 99’ dataset, (1999). The third international knowledge discovery and data mining tools competition dataset KDD99-Cup URL: *http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html*

Laskov, P.; Schäfer, C. & Kotenko, I., (2004). Intrusion detection in unlabeled data with quarter-sphere support vector machines, In: *Proc. DIMVA*, pp. 71-82, 2004.

Lee, W.; Stolfo, S. J. & Mok, K.W., (1998). Mining audit data to build intrusion detection models, *Proc. Int. Conf. Knowledge Discovery and Data Mining (KDD’98)*, pp.66-72, 1998.

Lee, W.; Stolfo, S. J. & Mok, K.W., (1998). A data mining framework for building intrusion detection model, *Proceedings of the IEEE Symposium on Security and Privacy*, pp.120-132, 1999.

Leung, K. & Leckie, C., (2005). Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters, *ACSC2005*, 2005.

Li, K.L.; Huang, H.K.; Tian, S.F. & Xu, W., (2003). Improving one-class SVM for anomaly detection, *International Conference on Machine Learning and Cybernetics*, Vol. 5, pp. 3077-3081, 2003.

Lippmann, R.P., (2000). Evaluating Intrusion Detection Systems: the 1998 DARPA Off-Line Intrusion Detection Evaluation, *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, Vol. 2.

Manganaris, S.; Christensen, M.; Zerkle, D. & Hermiz, K., (2000). A Data Mining Analysis of RTID Alarms, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 34, No. 4, pp. 571-577, 2000.

Oldmeadow, J.; Ravinutala, S. & Leckie, C., (2004). Adaptive Clustering for Network Intrusion Detection, In *Proceedings of the Third International Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2004.

Osuna, E.; Freund, R. & Girosi, F., (1997). Training support vector machines: an application to face detection, *International Conference on Computer Vision and Pattern Recognition(CVPR97)*, pp.30-136, 1997.

Portnoy, L.; Eskin, E. & Stolfo, S., (2001). Intrusion Detection with Unlabeled Data Using Clustering, In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security*, 2001.

QGPOP, (2010). URL: *http://www.qgpop.net/en/*

REN-ISAC, (2010). URL: *http://www.ren-isac.net/*

Schölkopf, B.; Platt, J.; Shawe-Taylor, J.; Smola, A. & Williamson, R., (2001). Estimating the support of a high-dimensional distribution, *Neural Computation*, 13(7):1443-1471, 2001.

Sebring, M. M.; Shellhouse, E.; Hanna, M. E. & Whitehurst, R. A., (1988). Expert systems in intrusion detection: A case study, *Proceedings of the 11th National Computer Security Conference*, pp.74-81, Baltimore, Maryland, October, 1988.

Snort, (2010). URL: *http://www.snort.org/*

SNS7160 IDS System, (2010). Symantec network security 7100 series

Sourcefire, (2010). URL: *http://www.sourcefire.com/*

Song, J.; Ohba, H.; Takakura, H.; Okabe, Y. & Kwon, Y., (2007). A Comprehensive Approach to Detect Unknown Attacks via Intrusion Detection Alerts, *ASIAN2007 Focusing on Computer and Network Security*, LNCS 4846, pp. 247-253, Doha Qatar, December 2007.

Song, J.; Ohira, K.; Takakura, H.; Okabe, Y. & Kwon, Y., (2008a). A Clustering Method for Improving Performance of Anomaly-based Intrusion Detection System, *IEICE Transactions on Information and Communication System Security*, Vol.E91-D, No.5, pp.1282-1291, May. 2008.

Song, J.; Takakura, H. & Kwon, Y., (2008b). A Generalized Feature Extraction Scheme to Detect 0-Day Attacks via IDS Alerts, *The 2008 International Symposium on Applications and the Internet(SAINT2008)*, The IEEE CS Press, 28 July - 1 Aug. 2008.

Song, J.; Takakura, H. & Okabe, Y., (2008c). Cooperation of intelligent honeypots to detect unknown malicious codes, *WOMBAT Workshop on Information Security Threat Data Exchange (WISTDE 2008)*, The IEEE CS Press, April, 2008.

Song, J.; Takakura, H.; Okabe, Y. & Kwon, Y., (2009). Unsupervised Anomaly Detection Based on Clustering and Multiple One-class SVM, *IEICE Transactions on Communications*, Vol. E92-B, No. 06, pp.1981-1990, Jun. 2009.

TELECOM-ISAC, (2010). URL: *https://www.telecom-isac.jp/index.html* (Japanes only)

Treinen, J. J. & Thurimella, R., (2006). A Framework for the Application of Association Rule Mining in Large Intrusion Detection Infrastructures, *RAID 2006*, LNCS 4219, pp. 1-18, 2006.

Yu, D. & Frincke, D., (2004). A Novel Framework for Alert Correlation and Understanding, *ACNS 2004*, LNCS 3089, pp. 452-466, 2004.

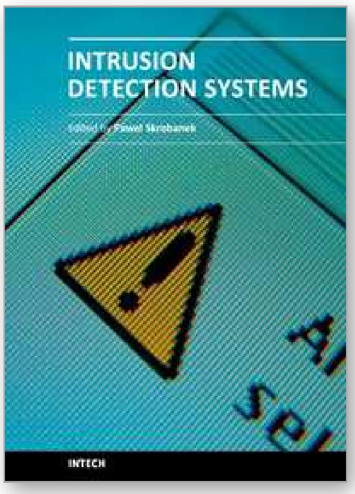
Vapnik, V., (1995). The nature of statistical learning theory, *Springer Verlag*, New York, 1995.

Vapnik, V., (1998). Statistical Learning Theory, *Wiley*, New York, 1998.

Wang, Q. & Megalooikonomou, V., (2005). A Clustering Algorithm for Intrusion Detection, In *SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, 2005.

Warrender, C.; Forrest, S. & Pearlmutter, B., (1999). Detecting intrusions using system calls: alternative data models, *1999 IEEE Symposium on Security and Privacy*, pp. 133-145, IEEE Computer Society, 1999.

Zurutuza, U. & Uribeetxeberria, R., (2004). Intrusion Detection Alarm Correlation: A Survey, In *Proceedings of the IADAT International Conference on Telecommunications and Computer Networks*, 1-3 December, 2004.



INTRUSION DETECTION SYSTEMS

Edited by Pawel Skrobanek

INTECH

Intrusion Detection Systems
Edited by Dr. Pawel Skrobanek

ISBN 978-953-307-167-1
Hard cover, 324 pages
Publisher InTech
Published online 22, March, 2011
Published in print edition March, 2011

The current structure of the chapters reflects the key aspects discussed in the papers but the papers themselves contain more additional interesting information: examples of a practical application and results obtained for existing networks as well as results of experiments confirming efficacy of a synergistic analysis of anomaly detection and signature detection, and application of interesting solutions, such as an analysis of the anomalies of user behaviors and many others.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821