SnortView: Visualization System of Snort Logs

Hideki Koike
Graduate School of Information Systems
University of Electro-Communications
1-5-1, Chofugaoka, Chofu
Tokyo 182-8585, Japan
koike@acm.org

Kazuhiro Ohno
Graduate School of Information Systems
University of Electro-Communications
1-5-1, Chofugaoka, Chofu
Tokyo 182-8585, Japan
ohno@vogue.is.uec.ac.jp

ABSTRACT

False detection is a major issue in deploying and maintaining Network-based Intrusion Detection Systems (NIDS). Traditionally, it is recommended to customize its signature database (DB) to reduce false detections. However, it requires quite deep knowledge and skills to appropriately customize the signature DB. Inappropriate customization causes the increase of false negatives as well as false positives. In this paper, we propose a visualization system of a NIDS log, named SnortView, which supports administrators in analyzing NIDS alerts much faster and much more easily. Instead of customizing the signature DB, we propose to utilize visualization to recognize not only each alert but also false detections. The system is based on a 2-D time diagram and alerts are shown as icons with different styles and colors. In addition, the system introduces some visualization techniques such as overlayed statistical information, source-destination matrix, and so on. The system was used to detect real attacks while recognizing some false detections.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Security

Keywords

Intrusion detection, Visualization

1. INTRODUCTION

In recent years, cyber attacks, such as Internet Worms, DDoS attacks, or other unauthorized access, are increasing more and more. As one of the solutions, Network-based Intrusion Detection Systems (NIDS), which can detect such attacks in real-time, are widely used. The NIDS use the database (DB) called signature DB which contains the signatures of known attacks. When the network packet matches one of its signatures, the NIDS produces an alert.

However, a huge number of false detections is a major issue when maintaining the NIDS. There are two kinds of false detection, false positive and false negative. False positive judges normal traffic as an attack, while false negative fails to produce an alarm for a real attack. There are trade-offs between false positive and false negative detections. If the administrator reduces the number of signature rules in order to reduce false positives, the NIDS would miss the attacks which are not in the signature DB. If the administrator adds rules to the DB so as not to miss the attacks, the possibilities of false positives are increased.

In order to solve this false detection issue, it is generally recommended that the signature DB should be appropriately customized depending on the monitored environments. However, appropriate customization of the signature DB requires quite deep knowledge about the signatures and skills for operating NIDS. Since it is difficult to automate this process, it must be done manually by the administrators. Unfortunately, there not many administrators have such deep knowledge and skills.

The NIDS also requires deep knowledge and skills in investigating its logs. The administrators have to read the logs line by line and have to find the sign of real attacks by using their knowledge and skills. However, the NIDS logs are generally huge in size and it is a time-consuming task to analyze them.

This paper describes a NIDS log visualization system which supports administrators in analyzing NIDS alerts much faster and much more easily. Instead of customizing the signature DB *the most* appropriately, we propose to use a moderately customized signature DB and appropriately visualize all the information in the NIDS log which would contain false detections. To summarize, the traditional approach is to customize the signature DB in order to reduce the false positives and to reduce the load of administrators. However, it would cause the false negatives issue. Our approach, on the other hand, is not to make such too much customization in order to prevent the false negatives. Instead, we use visualization effectively to recognize the false positives in the logs.

This paper is organized as follows. The next section discusses which information should be visualized. Section 3 describes the visualization system we developed. Section 4 shows example analysis using our system. In Section 5, we discuss advantages and limitations of our system. Then, we conclude the paper.

VizSEC/DMSEC'04, October 29,2004,Washington,DC,USA.

2. INFORMATION TO BE VISUALIZED

For a number of years, we have been using Snort [6] to monitor several networks, including two campus networks and two company networks. Through the experience of investigating their logs, we identified which information is essential for investigating NIDS logs; we also identified which information is necessary for recognizing false detections.

2.1 Essential Information

First of all, we identified the minimum information which should be visualized. The following information is essential in investigating NIDS logs and should be visualized *simultaneously*.

- Time of access

  When the event occurred. Time information is crucial in intrusion detection. Not only the time of each event but also the order of the events and the time interval of related events are important.

- Type of access

  What type of access was detected. This information includes the classification of services (e.g., Web, Mail, FTP, DNS, etc.) and the classification of protocols (e.g., TCP, UDP, ICMP, etc.).

- Source of access

  From where the attack was executed. By using this information, the name of the organization where this source IP belongs to may be obtained.

- Destination of access

  To where the attack was sent. This information is used to decide priorities of investigation. For example, The access to important systems, such as Web servers or Mail servers, should be investigated as soon as possible. On the other hand, the access to client systems might be investigated later.

- Details of access

  This is necessary because the information which should be focused on is different depending on the access. Also, if the attack is targeted for services which are not provided, administrators may skip to check this attack.

2.2 Heuristics

By comparing logs in different environments and by analyzing detailed information in each log, the following alarms are more likely to be found to be false detections.

- Alarms which appear consecutively

  In our experience, the alarms which appear consecutively are possibly false positives.

- Alarms which appear many times

  In the same way, the alarms which appear many times in the entire log are more likely to be false positives.

- Alarms which conflict with provided services

  If an alarm for the attack on Windows IIS server is found in a network which does not provide such services, the alarm would be false positive.

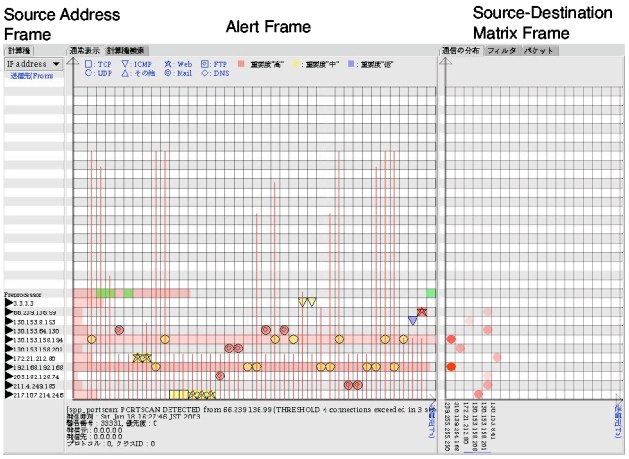

Figure 1: Snapshot of SnortView.

- Alarms about other networks

  If an alarm for a network which is not monitored is found, it would be false positive.

It should be noted that these cases are just possibilities. There are real attacks even though they match to these criteria. An important point is that our visualization approach will not filter out the alarms which match these criteria, but will still display them on screen to enable administrators to judge for themselves.

3. SNORTVIEW

3.1 Overview

Based on the discussions held in the previous section, we developed a visualization system for a NIDS log, named SnortView, for real-time monitoring of attacks as well as for recognition of false detections. The system is composed of two modules: the log analysis module and the visualization module. The log analysis module reads syslog and Snort alert log every two minutes for almost real-time monitoring. Next, these two logs are integrated into one medium format where each event is sorted by its time. The syslog is widely used by various programs to record the system information. The syslog contains evidential information of malicious activities, such as unusual termination of server programs. When we examine whether or not a particular NIDS alert is false detection, it is helpful to examine the syslog events which occur before and after the NIDS alert. Since the NIDS log and syslog are separated and located in the different directories, it is rather difficult to compare each log. Integrating these two logs based on time makes it much easier to analyze NIDS alerts.

3.2 Visualization

Figure 1 shows a snapshot of SnortView. The application window is mainly separated into the following three frames:

3.2.1 Source Address frame

At the left of Figure 1, there is a Source Address frame where the source IPs detected by NIDS are sorted and listed vertically.

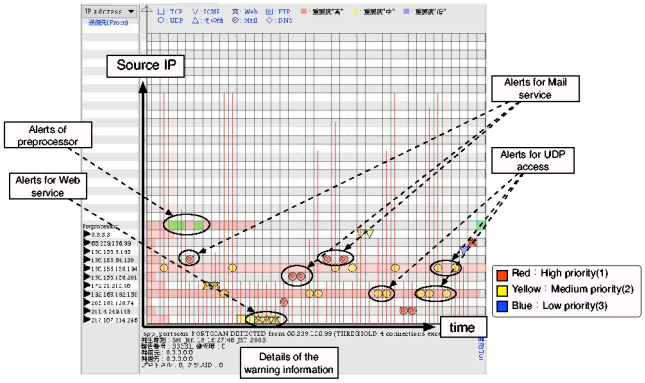

Figure 2: Alert panel

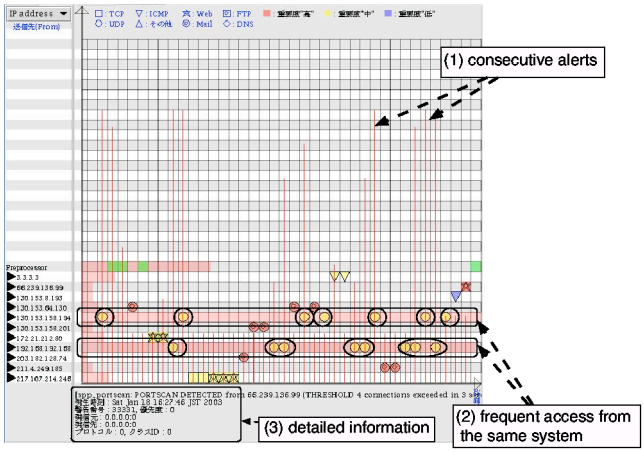

Figure 3: Statistical data are overlayed.

Table 1: Shapes and meanings of a symbol.

Classification of the service(1)		
Service Name	Shapes	Port number
Web	★	80, 3128, 8080
Mail	◎	25, 110, 143
FTP	⊡	20, 21
DNS	♢	53
Message of system		
System name	Shapes	Port number
Preprocessor	Green □	53
Syslog	⊠	none
Classification of protocol		
Protocol name	Shapes	Port number
TCP	□	Except defines with (1).
UDP	○	
ICMP	▽	
the others	△	

3.2.2 *Alert frame*

The middle of the application window shows an alert frame. In this frame, the vertical axis represents a list of source IPs as described above, and the horizontal axis represents time. Each NIDS alert is displayed as a colored icon as shown in Figure 2. The color represents the priority information of the Snort alert. That is, red, yellow, and blue mean priority 1, 2, and 3, respectively. Different shapes are assigned to each icon depending on the type of attacks as shown in Table 1.

In NIDS log, the same alert often appears repeatedly. If such alerts are simply visualized in the time diagram, the administrators would see only the series of the same icon and could obtain useful information from the visualization. We solved this problem by visualizing statistical information simultaneously such as shown in Figure 3. As shown in Figure 3(1), a vertical red line is overlayed on each icon. This line represents the number of consecutive alarms. This statistical visualization is useful to prevent redundant consecutive display of the same icons. On the other hand, as shown in Figure 3(2), horizontal red bars are also displayed in each source IP column. These bars show the total number of appearances of the source IP in entire Snort log.

At the bottom of the alert frame, the detail of each alarm is shown (Figure 3(3)). This is useful to recognize the conflict between the provided network services and the attack.

3.2.3 *Source-Destination Matrix frame*

The right of Figure 1 is the Source-Destination Matrix frame. In this matrix representation, a red circle represents communication between a particular source and a particular destination. The source IP is found by moving the focus to the left. The destination IP is found by moving the focus to the bottom.

When a user clicks a symbol in the alert frame, the communication path is highlighted as two blue lines (i.e., vertical and perpendicular) as shown in Figure 4. This helps the user to recognize the source IP and the destination IP of the communication. At the same time, the detailed information on the alarm is displayed in the detail window. The detailed information includes the alert message, time of the event, alert number, priority, source IP and port number, destination IP and port number, protocol, alert ID defined in Snort.

In general, a two-dimensional time diagram displays the transition of a set of particular pieces of information such as source IPs in this diagram. It is difficult to simultaneously show the source IPs and the destination IPs in the two-dimensional time diagram. We solve this problem by adding the matrix visualization.

4. EXAMPLE ANALYSIS

4.1 Detection of Exceptional Alert

In Figure 5, a particular source is periodically sending ICMP packets as indicated by ▽. It is often the case that such periodically continuing alerts are false positives. However, as we can see in this figure, another alert (i.e., □) exceptionally appears in the series of the same alert. When the administrator investigates the textual log, such an exceptional alert is hidden in the huge amount of the same alert and he/she cannot recognize this exceptional alert. However, the exceptional alert comes up in the visualization, and the administrator is successful in finding the alert.

4.2 Detection of Hidden Alert

Figure 6, shows two series of the same alerts (i.e., yellow ★). In our visualization, the same consecutive alerts are visualized as one symbol and an overlayed vertical line. In this

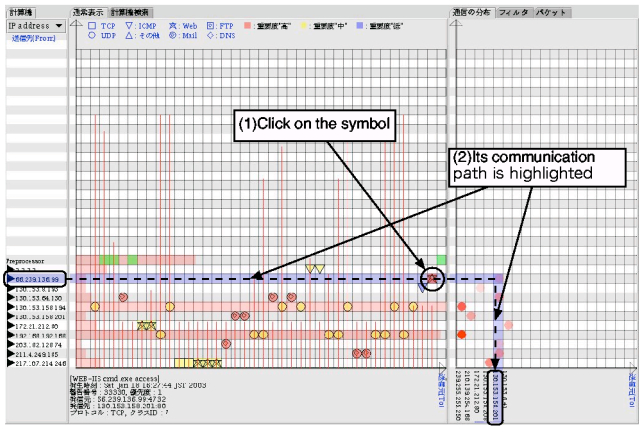

Figure 4: Communication path

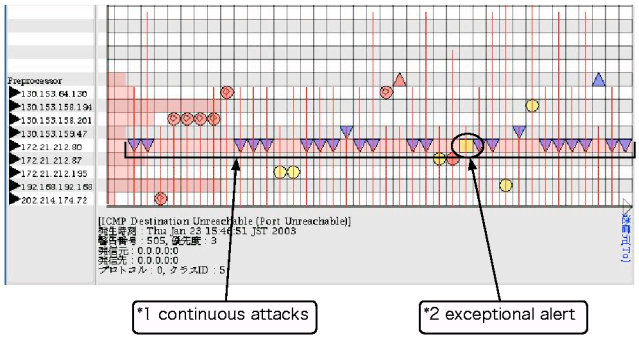

Figure 5: Detection of exceptional alert

example, the administrator can easily find an alert (purple ★) between two series of the same alerts.

4.3 Detection of Sequence of Attack

Figure 7 shows that a very small number of packets were sent in every fifteen minutes from a host in an outer network as indicated by ▽. Then the host finally executed an attack to the Web server (as indicated by ★)). Script kiddies often use automated tools which produce a number of alerts in a short period of time. These alerts are relatively easier to find. On the other hand, advanced attackers use this method to probe their target system. It is difficult to find correlation between these time separated attacks in textual log. By using the visualization, it is much easier to understand the correlation between probe activities and the attack.

4.4 Detection using Syslog Information

Figure 8 shows an example analysis using syslog information. In the figure, there is a symbol (★) which represents an attack on the Web service. This is an alert that a Web server on the monitoring network was attacked by CodeRed worm. Just after, there is another symbol (⊠) which represents a syslog event and its message is such as:

```
httpd: GET /scripts/root.exe?/c+dir HTTP/1.0 404 208
```

The error code 404 means that the specified address does not exist. In this case, the Web server did not halt and sent the

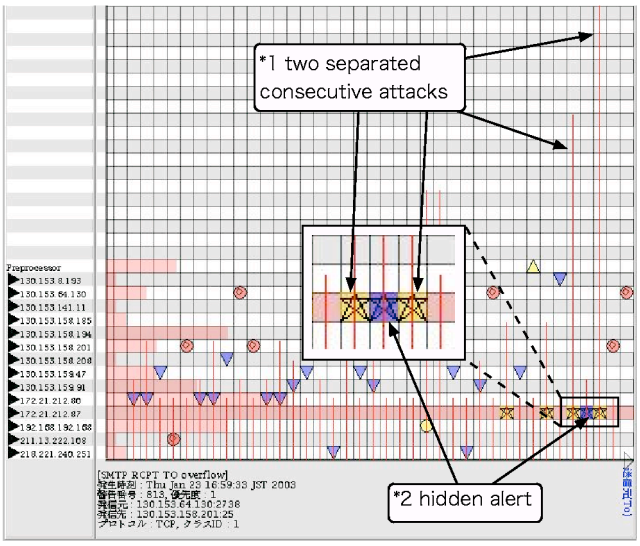

Figure 6: Detection of hidden alert

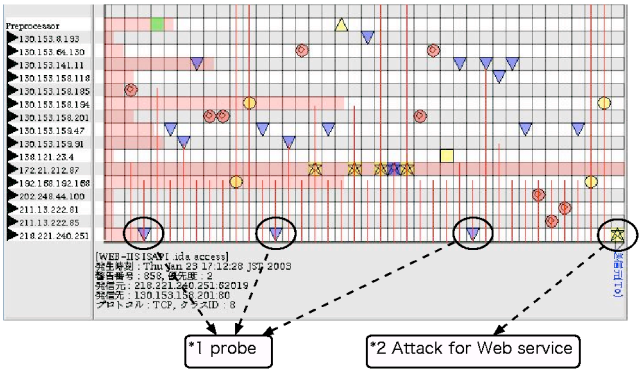

Figure 7: Detection of Sequence of attacks

error message appropriately. Therefore, the administrator knows that the attack did not succeed. By showing the NIDS alarm and syslog message simultaneously, it becomes easier to find the false detection.

5. DISCUSSIONS

5.1 Advantages

The first advantage of using our system is real-time monitoring capability. In general, a huge number of alerts are continuously produced and it is a time-consuming task to read and understand such logs. Therefore, it is rather difficult to monitor attacks in real-time. Our system applies the information visualization technique and will help to reduce the user's cognitive load.

The second advantage of our system is that it enables us to judge false detection visually. The system was designed to use the administrator's heuristics when he/she judges false detections. As we described previously, it requires deep knowledge and great skill to customize NIDS signature DB. The administrators who have such knowledge and skills are

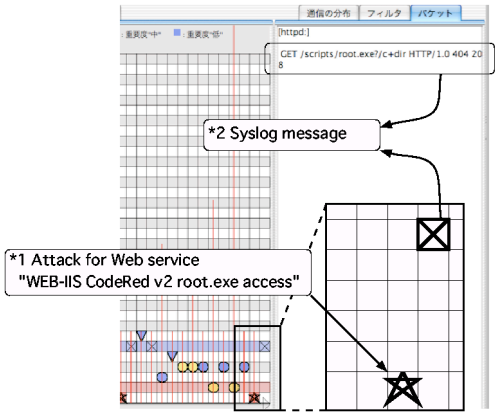

Figure 8: Detection using syslog information

not many. On the other hand, there are many administrators who have *medium* knowledge and skills. Although they might not be able to appropriately adjust their signature DB and judge false detections, they have basic knowledge and skills about NIDS. Our system will help such administrators to find false detection much more easily.

5.2 Limitations

There is information which are not used in our visualization. For example, header information of each packet and URLs related to alerts are not used. In backdoor or DoS attacks, it is often seen that unexpected code is written in packet headers. Showing such packet header information might help to recognize false detections.

Another issue is the amount of information displayed on the computer screen. SnortView overlays statistical information onto each attack. This prevents the visualization from being overwhelmed by the series of the same attack. By utilizing this visualization technique, the system currently shows 40 attacks which are not in succession. In our network environment, the system can visualize the events that occur during 4 hours in one screen. Since the usual attacks take a few minutes or tens of minutes from start to end, the system has enough capability to display such periods of time. However, the amount of events differs depending on the environment; the system might not be able to visualize such a sequence of attacks.

5.3 Related Work

SnortSnarf [4] and ACID [1] analyze Snort logs and show statistical information in HTML format. It can be referred from remote by using Web browsers. However, since SnortSnarf processes statistical analysis of Snort alerts all at once, the real time analysis is difficult. Although ACID does real time analysis, administrators have to read and understand the result which is written in text format.

RazorBack [5] reads a Snort alert and displays it on a window. It provides a GUI interface for selecting the log file, etc. Priority of each alert is represented as a colored circle. By selecting the reload button, the newest alert is shown. However, it just displays the log in text and there is no essential difference in browsing logs using “less” or “more”.

Tudumi [7] visualizes connection to a particular server in one 3-D visualization by taking several logs (e.g., syslog, sulog, wtmpx, etc.) as input. In Tudumi, the connecting hosts are categorized by its network domain and displayed as a stack of circles. It does not have real time monitoring capability.

Mielog [8] is an interactive log browser which utilizes some visualization techniques. Mielog visualizes each log event as a colored line in order to obtain an abstract view of the log. The colors of lines are assigned by calculating the frequency of the word in each event message. Mielog also performs statistical analysis of how many logs are produced in a unit of time. Mielog can be used to see NIDS logs, but it does not explicitly show the essential information in NIDS, such as source/destination IP, time, etc.

6. CONCLUSIONS

This paper described a visualization system of Snort logs, named SnortView, which was designed to help administrators in judging false detection. Although the basic visualization framework of SnortView is a traditional 2-D time diagram, we added a matrix representation which enables us to see source IP and destination IP simultaneously in the time diagram. SnortView is also used to overlay visualization of statistical information thereby enabling us to see the number of consecutive attacks in compact style. The real time monitoring capability of SnortView is also important.

We are investigating other knowledge or heuristics which can be used in judging false detection. Then we will apply such knowledge and heuristics to the future visualization framework.

7. REFERENCES

[1] Analysis Console for Intrusion Databases (ACID), http://www.andrew.cmu.edu/%7Erdanyliw/snort/snortacid.html

[2] Rebecca Gurley Base, Intrusion Detection, Macmillan Technical Publishing USA (1999).

[3] Robert F.Erbacher, Deborah Frincke, Visualization in Detection of Intrusions and Misuse in Large Scale Networks, Proceedings IEEE International Conference on Information Visualisation, pp.244-249 (2002).

[4] James A. Hoagland, Stuart Staniford, Viewing IDS alerts: Lessons from SnortSnarf, Proceedings of 2001 DARPA Information Survivability Conference and Exposition (DISCEX 2001), pp.12-14 (2001).

[5] RazorBack, http://www.intersectalliance.com/projects/RazorBack/

[6] M. Roesch, Snort - Lightweight Intrusion Detection for Networks, Proc. of the 1999 USENIX LISA conference (1999).

[7] Tetsuji Takada, Hideki Koike, Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs, Proc on Information Visualization (IV2002), IEEE/CS, pp.570–576, 2002.

[8] Tetsuji Takada, Hideki Koike, Mielog: A Highly Interactive Visual Log Browser Using Information Visualization and Statistical Analysis, Proc. of LISA XVI Sixteenth Systems Administration Conference, The USENIX Association, pp.133–144, 2002.