Hierarchical Visualization of Network Intrusion Detection Data in the IP Address Space

Takayuki ITOH[1,2] Hiroki TAKAKURA[2]
Atsushi SAWADA[2] Koji KOYAMADA[3]
1) Department of Information Sciences, Faculty of Science, Ochanomizu University
2) Academic Center for Computing and Media Studies, Kyoto University
3) Center for the Promotion of Excellence in Higher Education, Kyoto University
itot@computer.org, {takakura, sawada, koyamada}@media.kyoto-u.ac.jp

1. Introduction

Intrusion detection is an active area of research. Many Intrusion Detection System (IDS) products are available, and these systems generally detect network intrusions and record the intrusions into log files. To understand the performance and limitations of these systems, we have conducted a study on several of the IDS products that are deployed on open and large-scale computer networks. We have identified the following issues:

- Several IDS systems send e-mails to network administrators for each incident (i.e., an intrusion record). They often send enormous numbers of e-mails if the network is large-scale. Moreover, it is very difficult to understand the relevancy and statistics of aggregate incidents by only receiving the alerts for individual incidents.
- Recent attacks often consist of complicated combinations of various incidents. Intelligent, intuitive, and real-time solutions are required for an overall understanding of the complicated behavior.
- Databases for storing IDS logs often grow huge, and therefore usability of the databases is often problematic. Solutions that assist the user in querying for data are desirable to reduce query operations.
- GUIs of current IDS products visualize information very superficially. For example, the time sequence of numbers of incidents of the whole domain may be visualized as simple bar charts or polygonal charts. The user may need to perform many operations to explore detailed information, but in many cases administrators are often too busy to take the time to operate the GUI.

Recently various studies on intrusion analysis and secure network management have been reported [1,2]. In addition to those, visualization of incidents is very effective for intuitively and quickly understanding their distribution.

This paper presents a new technique to visualize the contents of huge IDS log files. The goals of the visualization technique are to make the available statistics from IDS systems understandable and to offer an interactive way of exploring detailed information. Another feature of the visualization technique is representing the distribution of incidents in IP-address spaces, revealing the relevancy of the distribution to the organizational structure of real society.

The technique first forms a four-level hierarchy of computers, by grouping the computers according to their IP addresses byte-by-byte. It then visualizes the hierarchical data as bars and nested rectangles [3,4], where bars denote computers and rectangles denote groups of computers. It finally represents the statistics of incidents by mapping the number of incidents of each computer as heights of the bars. The technique can represent the distribution of incidents in large-scale computer networks consisting of several thousand computers.

The technique helps the user intuitively understand the distribution and trend of enormous numbers of incidents in IP-address spaces of computer networks. It also helps the discovery of relevant relationships between distributions of incidents and the organization of real society, because IP addresses are usually assigned according to the organization of real society.

Moreover, the technique can provide the capability to explore detailed information about incidents for each computer, by representing computers as clickable icons. This capability assists users in exploring the detailed information of incidents for each computer.

This paper presents experimental results on visualizing enormous numbers of real incidents, and describes what kind of trends are observed from the experimental results.

2. Related works

Many IDS products provide detection, warning, and analysis capabilities for incidents, but they have not completely solved the issues described in Section 1. Several recent works improve the issues.

On the other hand, it is important to minimize damage by discovering high-security incidents intuitively and immediately, and information visualization is a valuable technology for this task. As described in the sidebar “Visualization for computer network and intrusion detection”, recent works for visualization of network intrusion include the following features:

- Visualization and detail-on-demand user interfaces showing time sequences of network traffic,
- Filtering of error detection or unimportant malicious accesses from visualization results,
- Visual data mining for discovery of suspicious traffic patterns from general log files, and
- Visualization of distribution of traffic on IP-address-oriented display spaces.

The technique presented in this paper can be categorized as “visualization of IP-address-oriented spaces” here the difference of this technique over existing techniques is that this technique attempts to maximize the density of the information on display. This feature represents computers as small clickable icons, enabling a user interface that presents its detail on-demand for each computer.

Also, the technique is useful for discovering the behavior of incidents relevant to the distribution of computers and the organizational structure of real society. For example, it visualizes the following behaviors:

- One computer attacks many others simultaneously.
- Many computers attack one computer simultaneously.
- When a computer is attacked and virus is placed on the computer, it then turns to attack other computers.
- One (or more) computers attack other computers in the same group or department.

With these features, the presented visualization technique complements existing techniques well.

3. Hierarchical data visualization

3.1 Rectangle packing for hierarchical data

The proposed technique applies a hierarchical data visualization technique presented in [3,4]. Figure 1 is an example of the visualization by this technique, which represents leaf-nodes as black square icons, and branch-nodes as rectangular borders enclosing the icons.

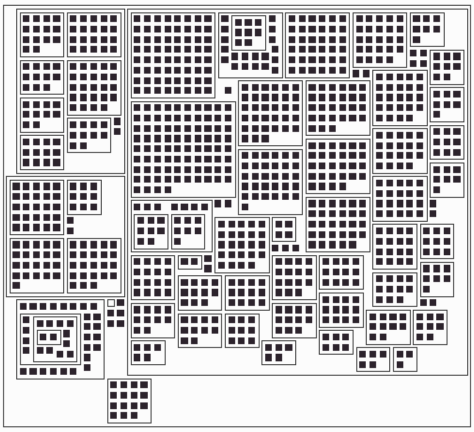


Figure 1. Example of hierarchical data visualization using a rectangle packing algorithm.

The visualization technique places thousands of leaf-nodes into one display space while satisfying the following conditions:

- It never overlaps the leaf-nodes and branch-nodes in a single hierarchy of other nodes,
- It attempts to minimize the display area requirement, and
- It draws all leaf-nodes by equally shaped and sized icons.

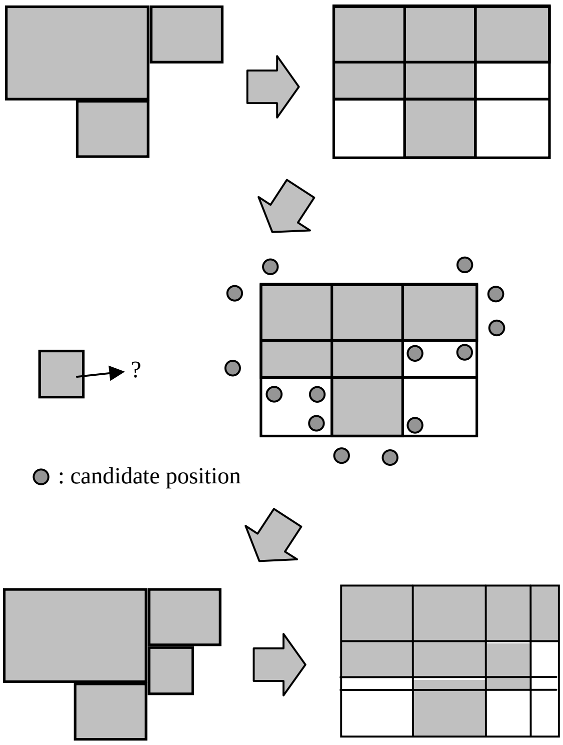


Figure 2. Improved rectangle packing algorithm. (Upper) Previously-placed rectangles, and grid-like subdivision of a display space. (Center) Candidate positions for placing the current rectangle. (Lower) Placement of the current rectangle, and the update of grid-like subdivision.

This representation style is suitable to equally visualize thousands of leaf-nodes of hierarchical data in one display space. We applied the technique to visualization of bioactive chemicals [4], distribution of jobs in parallel computing environments [5], and so on.

The technique first packs icons, and then encloses them in rectangular borders. Similarly, it packs a set of rectangles that belong to higher levels, and generates the larger rectangles that enclose them. Repeating the process from the lowest level toward the highest level, the technique places all of the data onto the layout area. The packing algorithm for icons and rectangles is the key technology for the visualization technique. Itoh et al. proposed a rectangle packing algorithm [3] for hierarchical data visualization, but an improved rectangle packing algorithm has been later presented in [4]. Both algorithms place icons and rectangles one-by-one onto display spaces, while the algorithms choose their positions from multiple candidate positions.

As shown in Figure 2, the improved rectangle packing algorithm [4] applies grid-like subdivision of a display area using extension lines of edges of previously placed rectangles. The algorithm quickly generates multiple candidate positions for the rectangle currently being placed by referring to the grid-like subdivision. It generates at most four candidates at the corner of empty subspaces of the grid-like space, where the current rectangle can be placed without yielding any unnecessary gaps with previously placed rectangles. The algorithm then decides the position of the rectangle while it avoids overlapping the rectangle with previously placed ones, and attempts to minimize the area and aspect ratio of the whole grid-like space. If there is no adequate candidate position to place the rectangle, the algorithm additionally generates several candidate positions outside the grid-like space, and selects one of the candidates to place the rectangle.

3.2 Visualization in the IP address space

The presented technique groups the computers according to their IP addresses to form hierarchical data. It first groups them according to the first byte of the IP addresses. It again groups them according to the second byte of the IP addresses, and finally groups according to the third byte of the IP addresses. Consequently the technique forms four-level hierarchical data as shown in Figure 3(Left). The technique visualizes the structure of computer network by representing the hierarchical data as shown in Figure 3(Right). Here, black icons in Figure 3(Right) represent computers, and the rectangular borders represent groups of computers.

We think that the technique is useful for the visualization of computer network spaces because:

- The technique visualizes large-scale hierarchical data containing thousands of leaf-nodes without overlapping, and therefore it can represent thousands of computers as clickable icons in one display space. The technique is therefore useful as a GUI to directly explore detailed information about incidents of arbitrary computers in large-scale computer networks.

- The technique visualizes a hierarchy of computers according to their IP addresses. Therefore, it can briefly represent the correlation between incidents and groups of computers in real society, because IP addresses are often assigned according to the structure of a real organization.

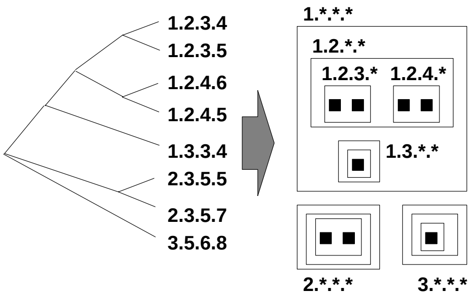


Figure 3. (Left) Hierarchy of computers according to their IP addresses. (Right) Illustration of visualization results of the hierarchical data.

4. Implementation

4.1 Network intrusion detection data

The presented technique consumes the log files of a commercial IDS system (Cisco Secure IDS 4320 [6]). The system detects incidents based on signatures that predefine the typical patterns of malicious accesses. The technique inputs the following items from the description of the log files, as shown in Figure 4(1):

- IP address of a computer sending incidents.
- IP address of a computer receiving incidents.
- Date and time.
- Positive integer ID (signature ID) that denotes the specific signature.
- Security level (1, 2, 3, 4, and 5).

4.2 Visualization procedure

Consuming the log files, the presented technique visualizes the incidents in the following processing order:

RDB-like data structure:

Consuming the log file, the presented technique forms a data structure like a relational database (RDB), as shown in Figure 4(2). It constructs tables for time, signature IDs, security levels, senders' IP addresses, and receivers' IP addresses. The data structure accelerates the aggregation of incidents.

Construction of hierarchical data:

Simultaneously the technique lists the IP addresses of senders and receivers, and forms hierarchical data by referring to IP addresses byte-by-byte, as shown in Figure 4(3). Here all the computers described in the log file are registered in the hierarchical data.

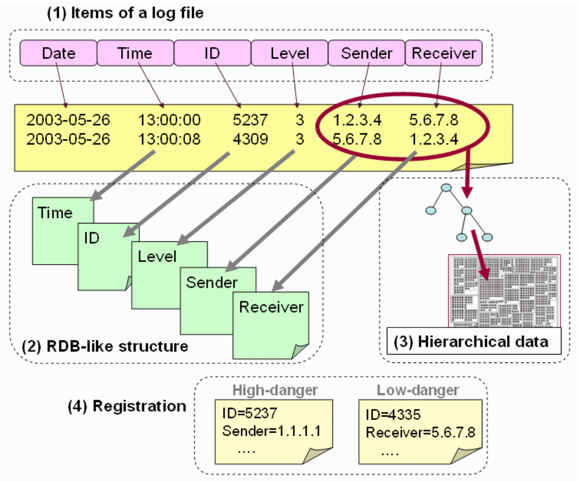


Figure 4. Processing order of the proposed visualization technique.

Aggregation of incidents for each computer:

The technique then counts the total number of sending and receiving incidents for each computer. Here it can specify the conditions, such as signature IDs, security levels, and range of times, to filter non-important incidents. If a signature ID is specified, the technique counts them, referring to the signature ID table. Similarly, it refers to the time or security level tables if the range of time or the security level is specified.

Representation:

The technique then visualizes the hierarchical data. Here it represents the numbers of sending and receiving incidents for each computer, by mapping the numbers as heights of leaf-nodes. As shown in Figure 5, the technique represents the numbers of sending and receiving incidents by assigning different colors. Examples shown in Figures 8 to 10 represent the number of sent incidents as blue, and the number of received incidents as red.

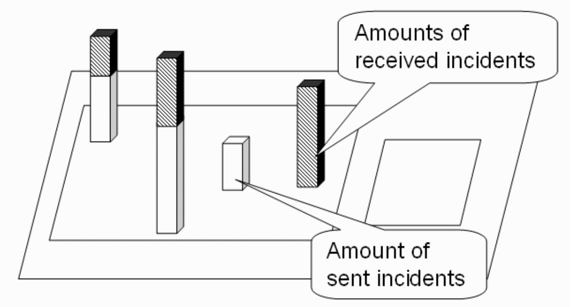


Figure 5. Illustration of visualizing the numbers of incidents as heights of leaf-nodes.

Configuration of high-security (or low-security) incidents:

Generally an IDS does not always provide adequate warning of the security level of incidents because impact of incidents strongly depends on each computer network’s situation. The presented technique consumes the description of signature IDs and IP addresses of experienced high-security (or low-security) incidents, as shown in Figure 4(4). This capability allows an administrator to configure the visualization results according to his or her preferences, for example:

- "Incidents which have specific signature IDs are always erroneous or ignorable in this network",
- "Incidents which have specific signature IDs have damaged this network in the past", and
- "Incidents which have specific IP addresses of senders have damaged this network in the past".

Also, the capability allows configuring the following computers as high-security:

- "Computers that sent or received more than a constant number of incidents in a constant time", and
- "Computers whose number of sending or receiving incidents drastically increases."

The technique can control the level of detail of visualization by eliminating or assigning dark colors to leaf-nodes corresponding to low-security computers.

Also, it can assign bright colors to leaf-nodes corresponding to computers sending or receiving pre-defined high-security incidents, to alert administrators of the return of known attacks. The example shown in Figure 10 represents computers sending or receiving high-security incidents in yellow.

4.3 GUI capability

We developed the GUI of the presented technique as a Java Applet. The features of the GUI are as follows.

Dialog window for configuring conditions for counting incidents:

The GUI pops up a dialog window for configuring conditions for counting incidents, including signature IDs, security levels, ranges of times, and IP addresses. Figure 6(Upper) shows an example of the dialog window. Given the conditions, the technique only counts incidents satisfying the conditions. The GUI enables more focused visualization, for example:

- “The network was damaged during 13:05 to 13:10, so I would like to visualize the distribution of incidents during that time,”
- “The network was damaged by the specific signatures, so I would like to visualize the distribution of the signatures,” or
- “This specific computer is often problematic, so I would like to visualize the distribution of incidents related to the computer by specifying its IP address.”

Dialog window for listing incidents for specific computer:

The GUI pops up a dialog window that displays the list of incidents for a specific computer that is the sender or the receiver. The dialog window pops up when a leaf-node is clicked, and then shows the list of incidents for the specific computer corresponding to the clicked leaf-node. Figure 6(Lower-left) shows an example of the dialog window.

Dialog for listing records of typical attacks:

Reflection of previous strong attacks may be good references for secure management of computer network. The GUI pops up a dialog window for displaying the list of pre-defined previous strong attacks, for example:

- “A specific computer sent an enormous number of malicious attacks during 12:35 to 12:40 on Feb 21.”

Figure 6(Lower-right) shows an example of the dialog window, listing date and time, and IP addresses of senders. Selecting one of the attacks from the list, the technique visualizes the distribution of corresponding incidents. The capability should be useful for administrators who want to share and analyze previous damages. If the display space allows displaying a larger dialog window, additional information, such as IP addresses of receivers, and signature IDs, is presented so that users can easily specify past attacks.

Segments representing pairs of senders and receivers:

The GUI can display segments connecting pairs of leaf-nodes corresponding to senders and receivers. It displays the segments for a specific computer when a user clicks a leaf-node corresponding to the computer. This capability helps users to explore the propagation of incidents. For example, many incidents from the same sender concentrate their attacks on a small number of receivers, or distribute their attacks to large numbers of receivers. Figure 9 shows the example of the segments.

4.4 HTML-based reporting

We developed a component to generate JPEG-format image files of visualization results. We also developed a component to generate HTML-based reports using the image generation component. The implementation of the reporting component frequently repeats generating image files while counting the incidents, finally generating HTML files as indices of the image files. Figure 7 shows an example of the Web page generated by this function. Sharing the HTML and image files, multiple administrators can easily exchange knowledge of incidents to remotely manage the network.

The report itself does not support GUI capabilities described in the Section 4.3. Administrators should use the Java-based GUI to explore the details of incidents if they find malicious traffic in the HTML report. Having the Java-based GUI pop up with the specified time span from the HTML report window may prove useful.

5. Experimental results

This section introduces the results of the presented technique. We implemented the technique with Java 1.4 and Microsoft Windows XP on an IBM ThinkPad T42 (CPU 1.8GHz, RAM 756MB). They developed the GUI using Java Swing library, and the drawing component using Java AWT library.

Figures 8 to 10 show the visualization results of an IDS log file used in a real network environment. Here, the numbers of sent incidents are represented in blue, and the numbers of received incidents are represented in red. In the all figures viewpoints are right sides of the bars and rectangles.

Figure 8 shows the time sequence of visualization results using the log file recorded in 6 hours, containing 61822 lines and 3984 computers. In our measurement, the implementation took 120 seconds for reading the log file, 0.6 seconds for forming and visualizing the hierarchy of computers, and 7.1 seconds for recounting incidents while GUI operations. Figure 8(a) shows the result of amounts of incidents in 5 minutes, Figure 8(b) shows the result in 5 minutes just after the time of Figure 8(a), Figure 8(c) shows the result in 5 minutes 2 hours after the time of Figure 8(b), and Figure 8(d) shows the result in 5 minutes 2 hours after the time of Figure 8(c).

Figure 8(a) shows several computers that sent incidents to other several computers. It might mean that the senders randomly searched for the targets of attacks. Figure 8(b) shows that a sender found a specific computer as the target of the attack, and the sender concentrated to send the incidents to it. Figure 8(c) shows that the sender shown in Figure 8(b) had been disconnected, but several new senders attempted to attack several computers. One of the new receivers was in the different department from the continuously attacked receivers. Figure 8(d) shows that some of the senders and receivers shown in Figure 8(c) had been disconnected, but many computers in the same department received incidents in a short time. The incidents might be a scan attack for the specific department.

Figure 9 shows the pairs of senders and receivers by yellow segments. When a user clicks a leaf-node on the display, the technique extracts incidents that the computer corresponding to the clicked leaf-node is sender or receiver, and represents the incidents as the yellow segments. The segments connect to the same tall red bar in the upper side of the figure, from multiple blue bars in small rectangles in the center of the figure. This example shows that many computers, in the same department denoted as a small rectangle, concentrated to send incidents to the same computer.

Figure 10 shows an example that the technique highlights leaf-nodes corresponding to the senders or receivers of high-security incidents on the real network. Here, administrators of the computer network used for these figures disconnected the senders or receivers of incidents 16 times in two months, because of pernicious attacks. We found in this enormous number of incidents that the signature IDs and IP addresses of the sender were identical in 5 of 16 attacks. They also found in another large group of incidents that the signature IDs and IP addresses of senders were identical in 3 of 16 attacks. These experiences mean that same kinds of incidents are often repeated for specific attack purposes. Therefore, the presented technique can contribute to alert the administrators by highlighting leaf-nodes corresponding to the senders or receivers of high-security incidents. Also, the technique can register previous damages, so that users can select the damages via the dialog shown in Figure 6(Lower-right).

Figure 11 shows closer-up views of above visualization results, including receivers in Figure 8(a), and senders and a receiver in Figure 9.

6. Discussion

As described in the Section 1, a feature of the presented technique is the representation of:

- statistics of incidents for thousands of computers,
- distribution of incidents on IP-address spaces, and
- relevancy of the distribution to the organizational structure of real society.

Figures 8 and 9 demonstrate that the technique visualizes interesting behavior about multiple computers in the same department. Figure 10 is useful for finding dangerous incidents from thousands of computers.

In addition to the above experiments, we think that the technique is useful for:

- observing the drastic change of incident patterns in IP-address-oriented spaces,
- observing if malicious computers attack an entire domain or only specific IP address blocks, or
- discovering that computers receive attacks from multiple computers, where most of the attacks are ignorable but a few others are serious.

On the other hand, the technique still has the following issues, which will be the focus of future work for the improvement of the technique.

Scalability:

Figures 8 to 10 shows that the technique is feasible for visualization of incidents of 4000 computers, but it might be difficult for a user to comprehend the distribution of intrusions and explore detailed information if there are more computers. There are several ideas for this problem, but we have not implemented any of them.

1) A zooming interface can be applied to the problem. Here we can switch the representation into two modes: overview mode and clickable mode. The former mode just represents the nest of IP addresses and highlights interesting areas, and the latter mode zooms into the interesting areas so that the display space can represent computers as clickable bars.
2) Another idea is removal of computers whose numbers of sending/receiving incidents are zero, and packing the representations of remaining computers into a smaller display space. The idea has a problem of stability of display layout since the content of computers changes over time, but the problem can be solved by applying layout template presented in the Section 5 of [3].

Occlusion:

The presented technique applies 3D representation for the statistics of incidents, but this style yields occlusions among metaphor of computers. One idea is applying multiple displays with independent viewpoints. Currently we are discussing to split the visualization technique into three displays per console. Another idea for minimizing the occlusions is applying the viewing optimization problem so that entropy of the visualization result is maximized. However, this approach may cause instability of viewing parameters. This is not only security-specific but also a general problem for 3D information visualization, and we think this area is ripe for future work for the enhancement of the presented hierarchical data visualization technique.

Visualization results may be crowded when the technique deals with long-term periods because numbers of incidents for each computer increase. In this case the number of occluded icons may also increase. We think that this problem will be improved by applying near-real-time observation, *e.g.*, by counting numbers of incidents and refreshing the display per minute.

Hierarchy representation:

While currently we represent hierarchy as a single color of nested rectangular borders, it might be more effective if different colors are assigned to the borders according to the hierarchy's depth, if understanding the relationship between the distribution of intrusions and the hierarchy of an organization is very critical.

Many-to-many traffic:

The presented technique only represents the link of traffic only by specifying a computer, as shown in Figure 9. This capability is not enough for the understanding of many-to-many traffic, for example the result shown in Figure 8(a). It would be useful if the implementation automatically detects important links and represents them. Parameter Focusing (introduced as [4] in Visualization for Computer Network and Intrusion Detection sidebar) may be a good reference to improve the technique for this issue.

7. Conclusion

This paper presents a technique for representing the statistics and trends of incidents in large-scale computer network.

We plan to prove the effectiveness of the technique by observing with real network management and users. Also, the following issues, as well as issues discussed in Section 6, will be the focus of future work based on this technique:

- Combination with intelligent techniques, such as data mining and knowledge management, to effectively discover and alert high-security incidents.
- Visualization of statistics of incidents in larger time span, such as a week or a month. We think that the number of incidents becomes less important as the time span grows, so representation of incidents should be enhanced for long-term visualization.
- Visualization focusing on time-varying distribution of incidents. Combining our technique with time-oriented visualization techniques, such of Mie-log (introduced as [7] in Visualization for Computer Network and Intrusion Detection sidebar) may effectively visualize the time-varying distribution of intrusions. Some kinds of trends or attack patterns can be also discovered by developing visualizations of the time-sequence of intrusions.

Acknowledgement

We appreciate the anonymous reviewers' many fruitful comments on this paper.

References

[1] Y. D. Cai, D. Clutter, G. Pape, J. Han, M. Welge, L. Auvil, MAIDS: Mining Alarming Incidents from Data Streams, SIGMOD Conference 2004, pp. 919-920, 2004.

[2] S. J. Stolfo, W. Lee, P. K. Chan, W. Fan, E. Eskin, Data Mining-based Intrusion Detectors: An Overview of the Columbia IDS, Project. SIGMOD Record, Vol. 30, No. 4, pp. 5-14, 2001.

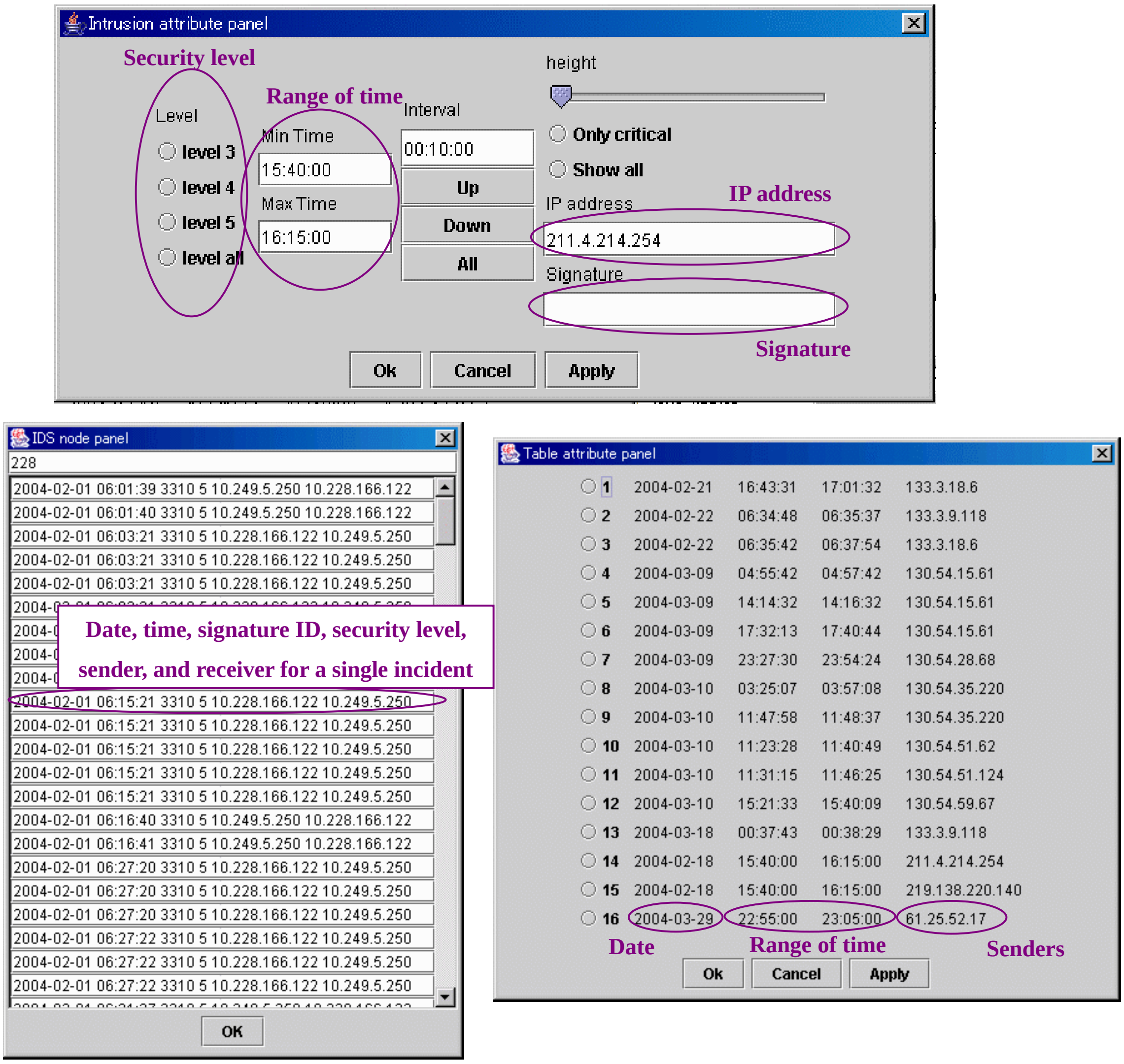
[3] T. Itoh, Y. Yamaguchi, Y. Ikehata, Y. Kajinaga, Hierarchical Data Visualization Using a Fast Rectangle-Packing Algorithm, IEEE Transactions on Visualization and Computer Graphics, Vol. 10, No. 3, pp. 302-313, 2004.

[4] T. Itoh, F. Yamashita, Visualization of multi-dimensional data of bioactive chemicals using a hierarchical data

visualization technique “HeiankyoView”, Asia Pacific Symposium on Information Visualization (APVIS) 2006, to be presented.

[5] Y. Yamaguchi, T. Itoh, Visualization of Distributed Processes Using "Data Jewelry Box" Algorithm, CG International 2003, pp. 162-169, 2003.

[6] Cisco Secure IDS.
http://www.cisco.com/japanese/warp/public/3/jp/product/security/ids/index.html



Min Time: 15:40:00

Max Time: 16:15:00

Interval: 00:10:00

IP address: 211.4.214.254

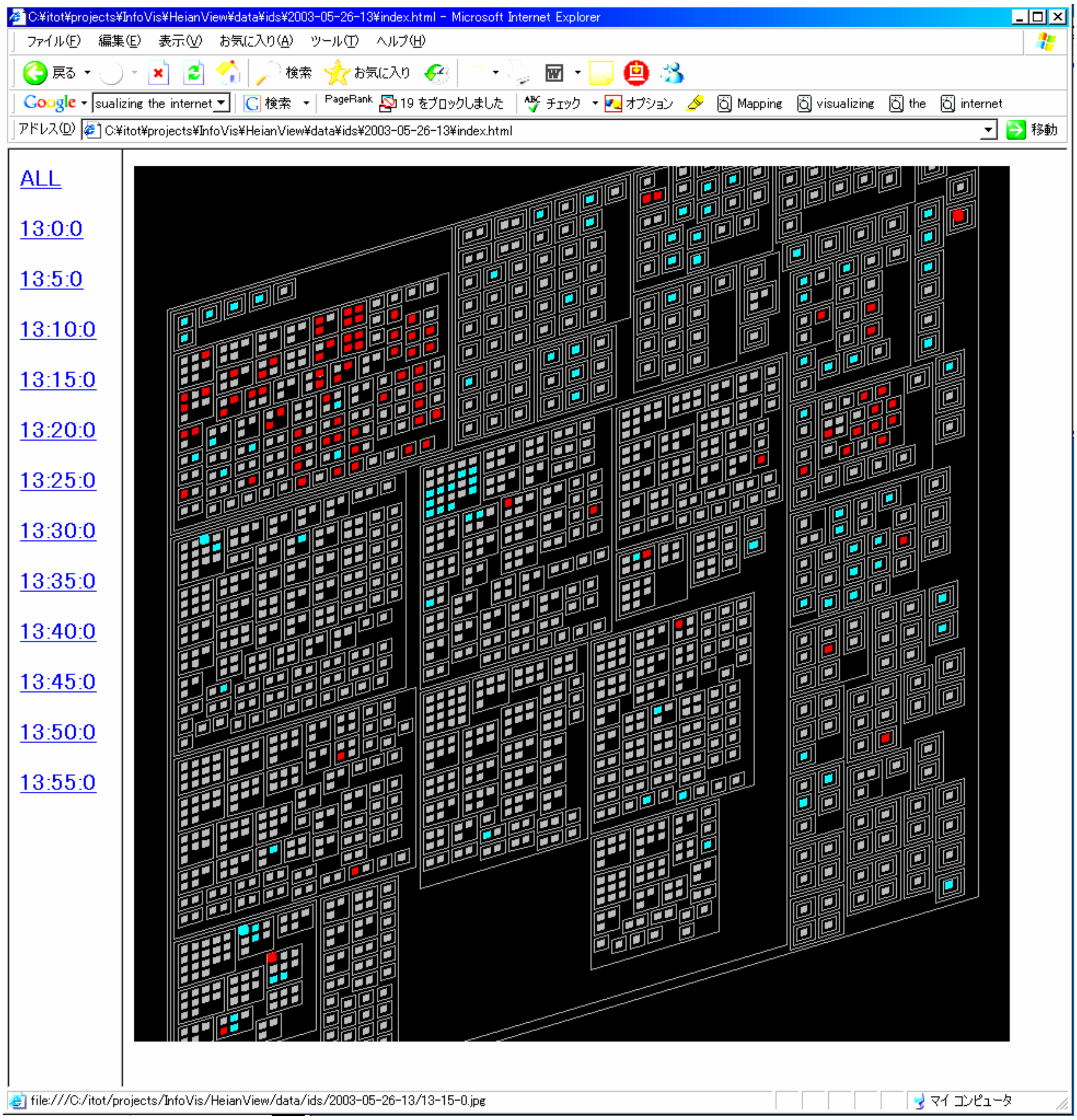
Signature:

Date, time, signature ID, security level, sender, and receiver for a single incident

```
228
2004-02-01 06:01:39 3310 5 10.249.5.250 10.228.166.122
2004-02-01 06:01:40 3310 5 10.249.5.250 10.228.166.122
2004-02-01 06:03:21 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:03:21 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:03:21 3310 5 10.228.166.122 10.249.5.250
2004-0
2004-0
2004-0
2004-0
2004-02-01 06:15:21 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:15:21 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:15:21 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:15:21 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:15:21 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:16:40 3310 5 10.249.5.250 10.228.166.122
2004-02-01 06:16:41 3310 5 10.249.5.250 10.228.166.122
2004-02-01 06:27:20 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:27:20 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:27:20 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:27:22 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:27:22 3310 5 10.228.166.122 10.249.5.250
2004-02-01 06:27:22 3310 5 10.228.166.122 10.249.5.250
```

○	1	2004-02-21	16:43:31	17:01:32	133.3.18.6
○	2	2004-02-22	06:34:48	06:35:37	133.3.9.118
○	3	2004-02-22	06:35:42	06:37:54	133.3.18.6
○	4	2004-03-09	04:55:42	04:57:42	130.54.15.61
○	5	2004-03-09	14:14:32	14:16:32	130.54.15.61
○	6	2004-03-09	17:32:13	17:40:44	130.54.15.61
○	7	2004-03-09	23:27:30	23:54:24	130.54.28.68
○	8	2004-03-10	03:25:07	03:57:08	130.54.35.220
○	9	2004-03-10	11:47:58	11:48:37	130.54.35.220
○	10	2004-03-10	11:23:28	11:40:49	130.54.51.62
○	11	2004-03-10	11:31:15	11:46:25	130.54.51.124
○	12	2004-03-10	15:21:33	15:40:09	130.54.59.67
○	13	2004-03-18	00:37:43	00:38:29	133.3.9.118
○	14	2004-02-18	15:40:00	16:15:00	211.4.214.254
○	15	2004-02-18	15:40:00	16:15:00	219.138.220.140
○	16	2004-03-29	22:55:00	23:05:00	61.25.52.17

Figure 6. GUIs for visualization of IDS data. (Upper) Dialog window for configuring conditions for counting incidents. (Lower-right) Dialog window for listing incidents for specific computer. (Lower-right) Dialog for listing records of typical attacks.



C:¥itot¥projects¥InfoVis¥HeianView¥data¥ids¥2003-05-26-13¥index.html

ALL

13:0:0

13:5:0

13:10:0

13:15:0

13:20:0

13:25:0

13:30:0

13:35:0

13:40:0

13:45:0

13:50:0

13:55:0

file:///C:/itot/projects/InfoVis/HeianView/data/ids/2003-05-26-13/13-15-0.jpg

Figure 7. Report generated as HTML and image files.

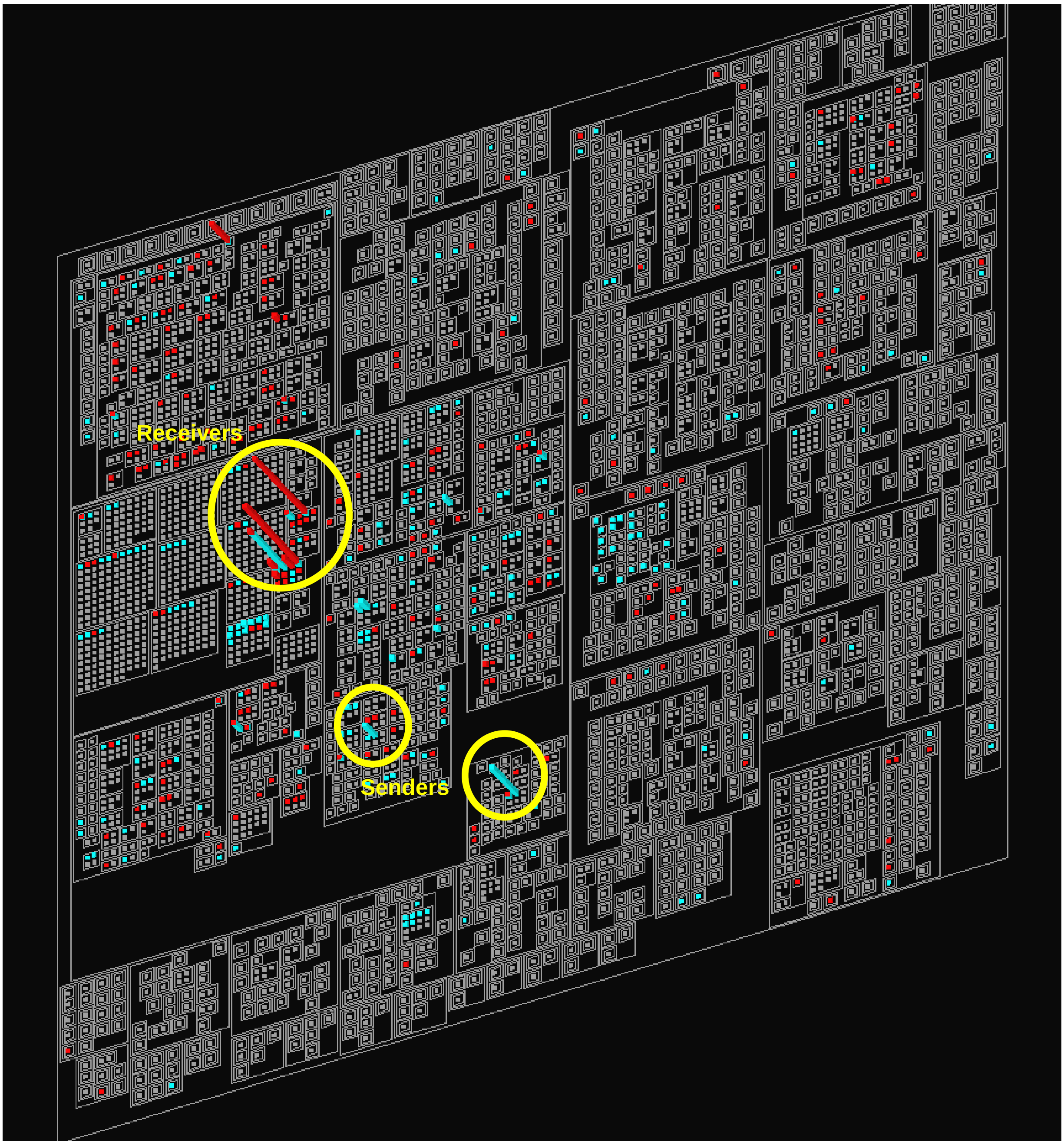


Figure 8. (a) Multiple senders and receivers are observed.

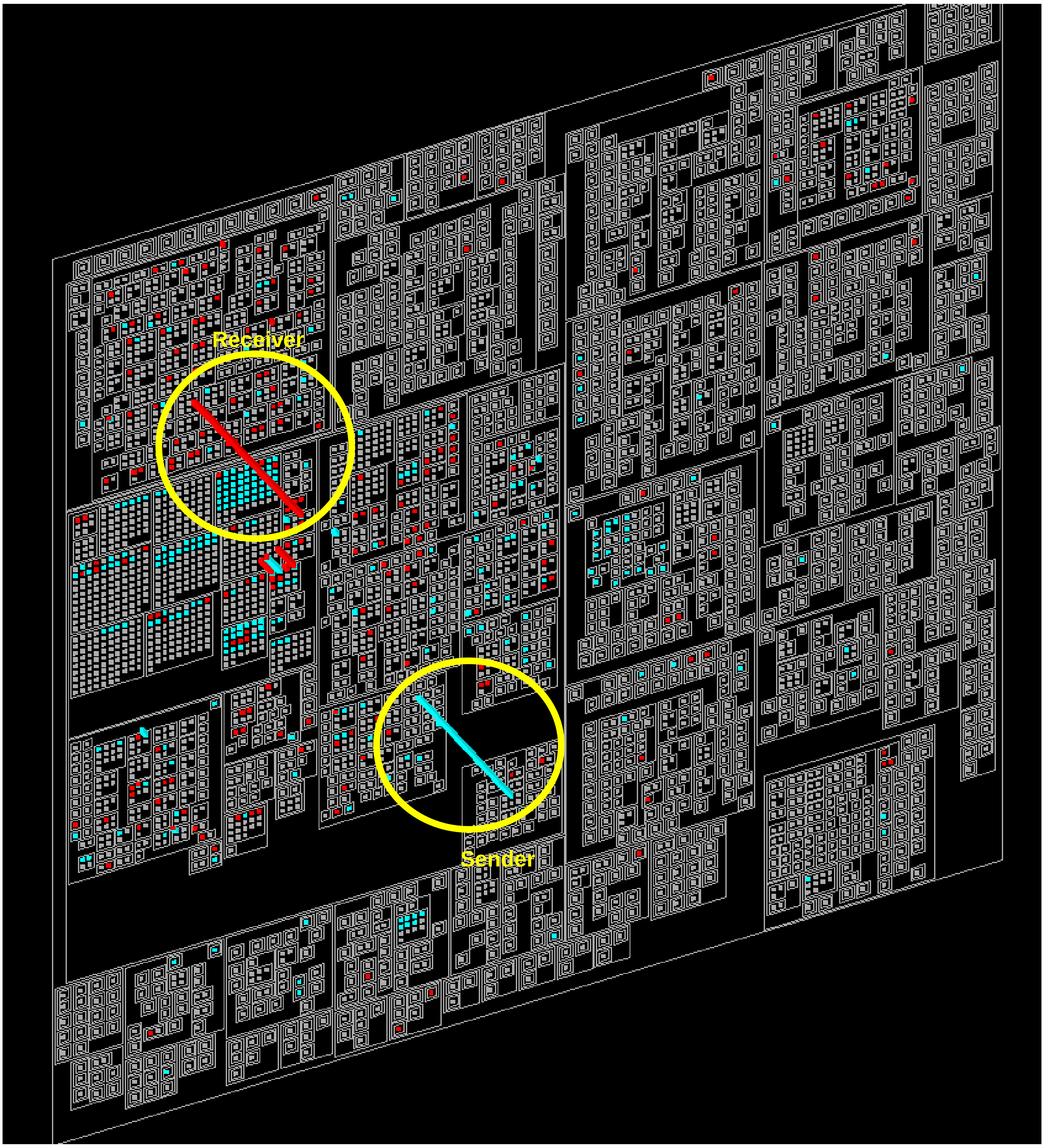


Figure 8. (b) A sender started concentrating its attack on the single receiver.

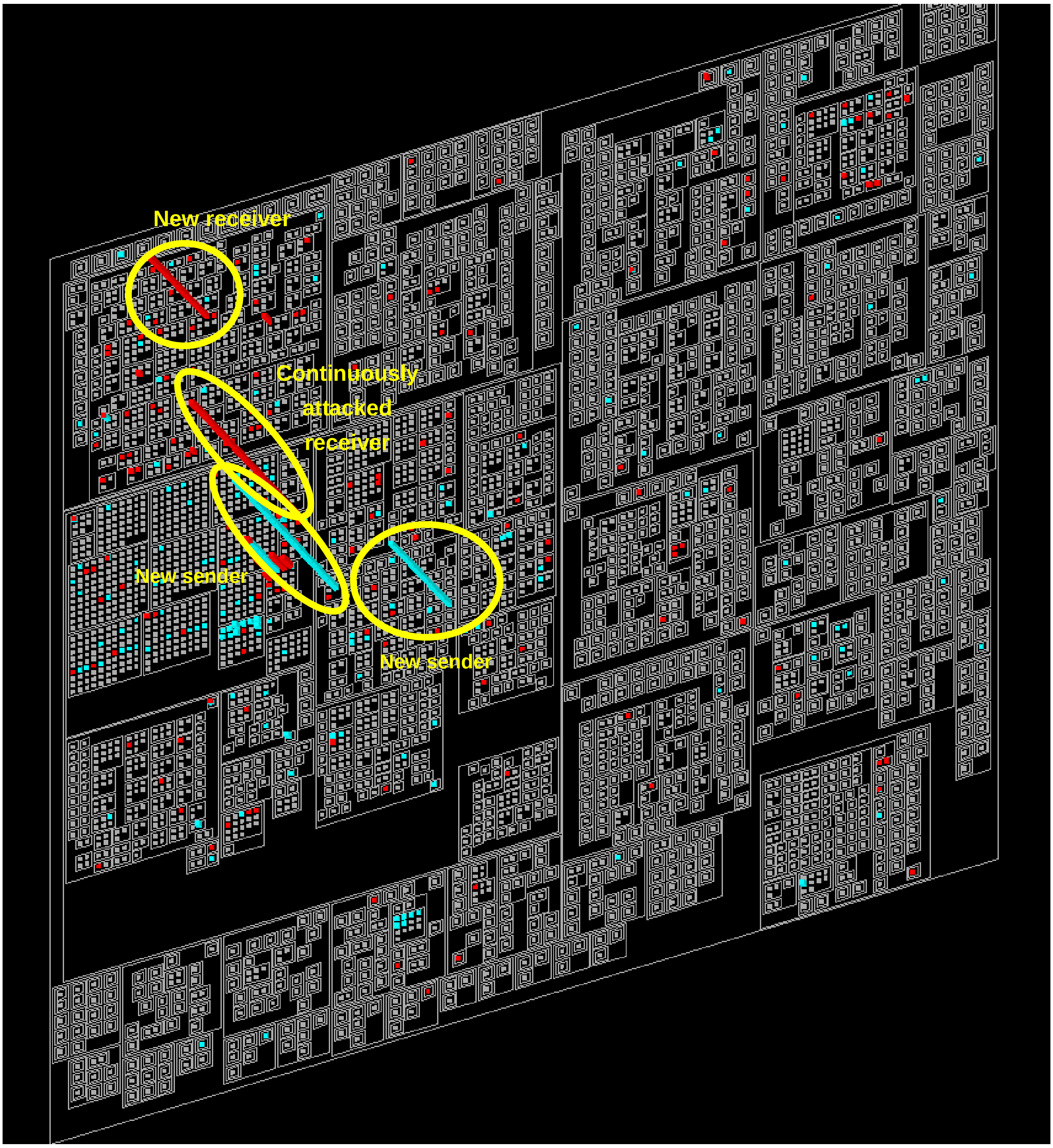


Figure 8. (c) The sender has been disconnected: however, several other computers turned as new senders, and several other computers received the attacks. The new senders were in the same department with the previous sender, but the new receiver was in a different department from the continuously attacked receivers.

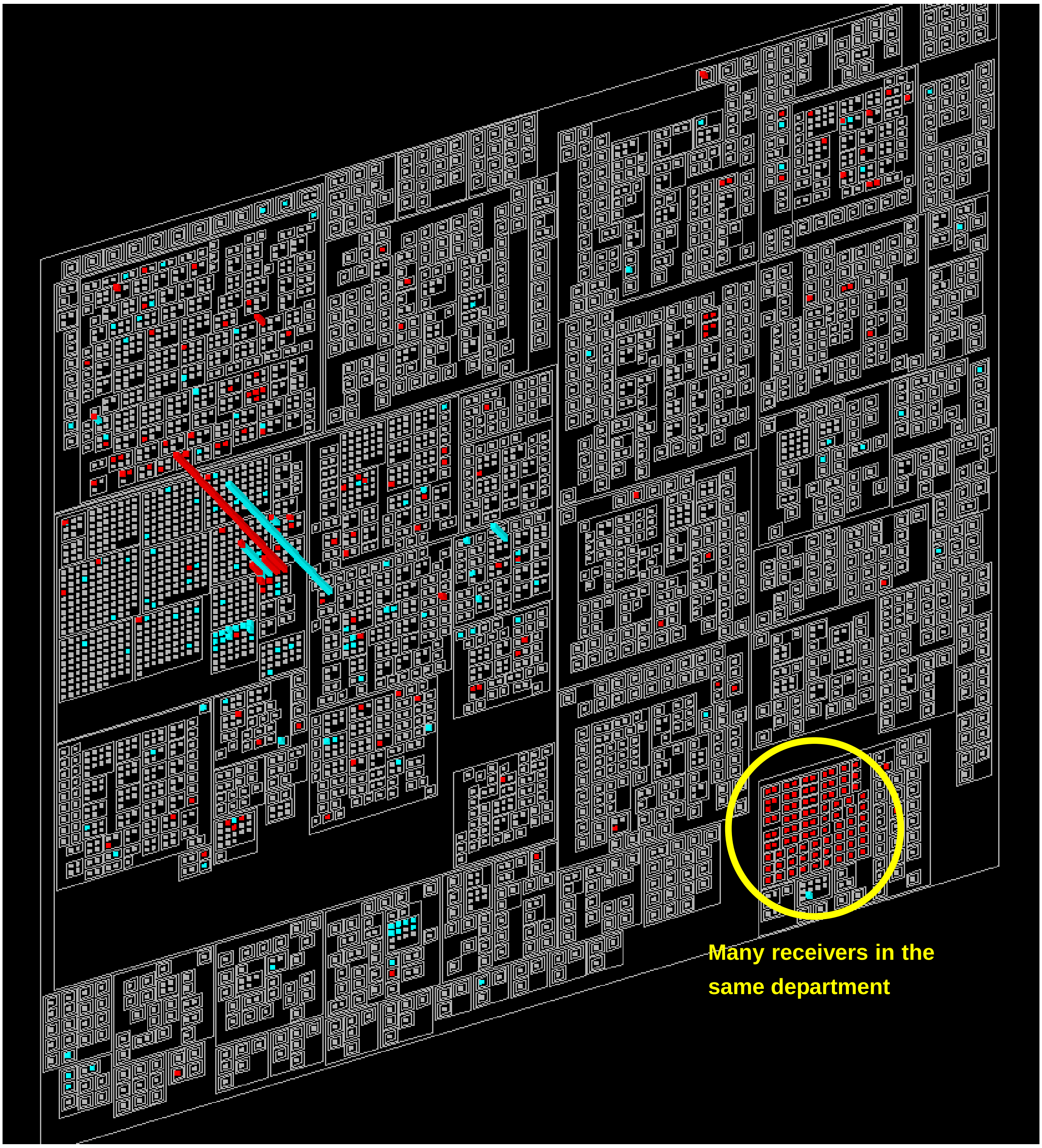


Figure 8. (d) Many receivers are observed in the same department. This pattern might be a scan attack for specific department.

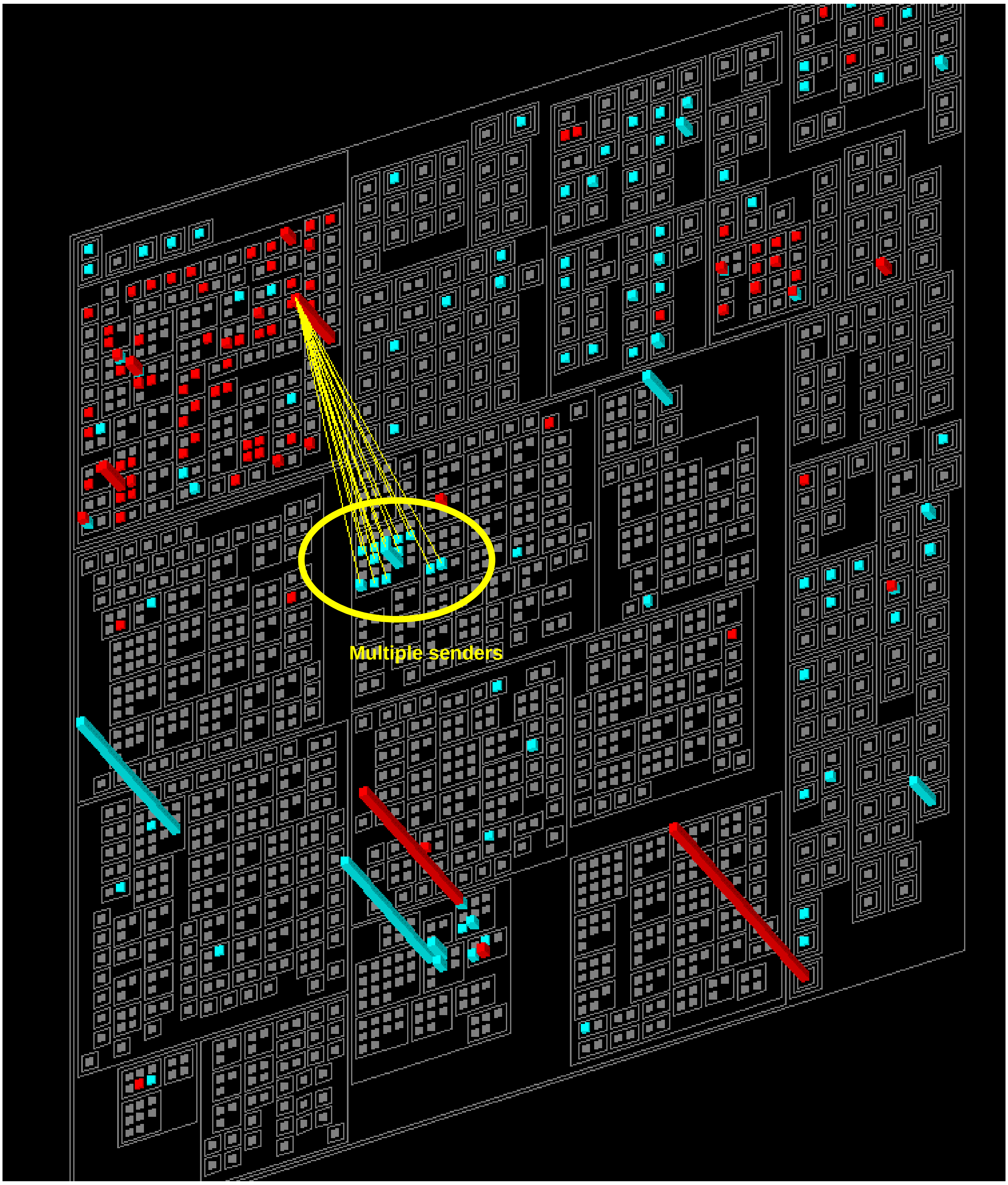


Figure 9. Multiple computers in the same organization directed to send incidents to the same computer.

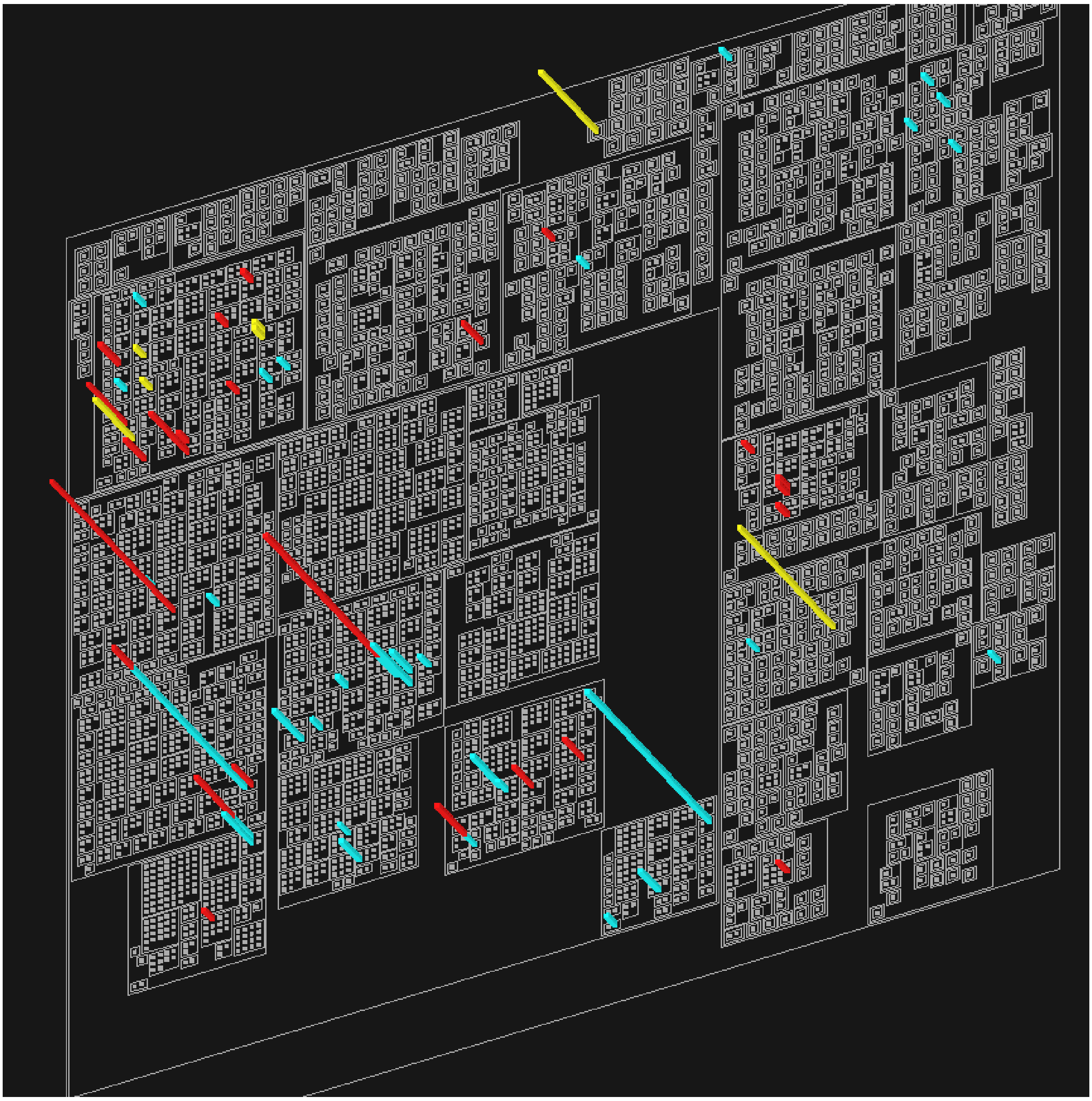


Figure 10. Yellow leaf-nodes represent the computers which sent or received incidents whose IDs and senders were same as past danger incidents. This representation helps to reduce damages by notifying the past incidents.

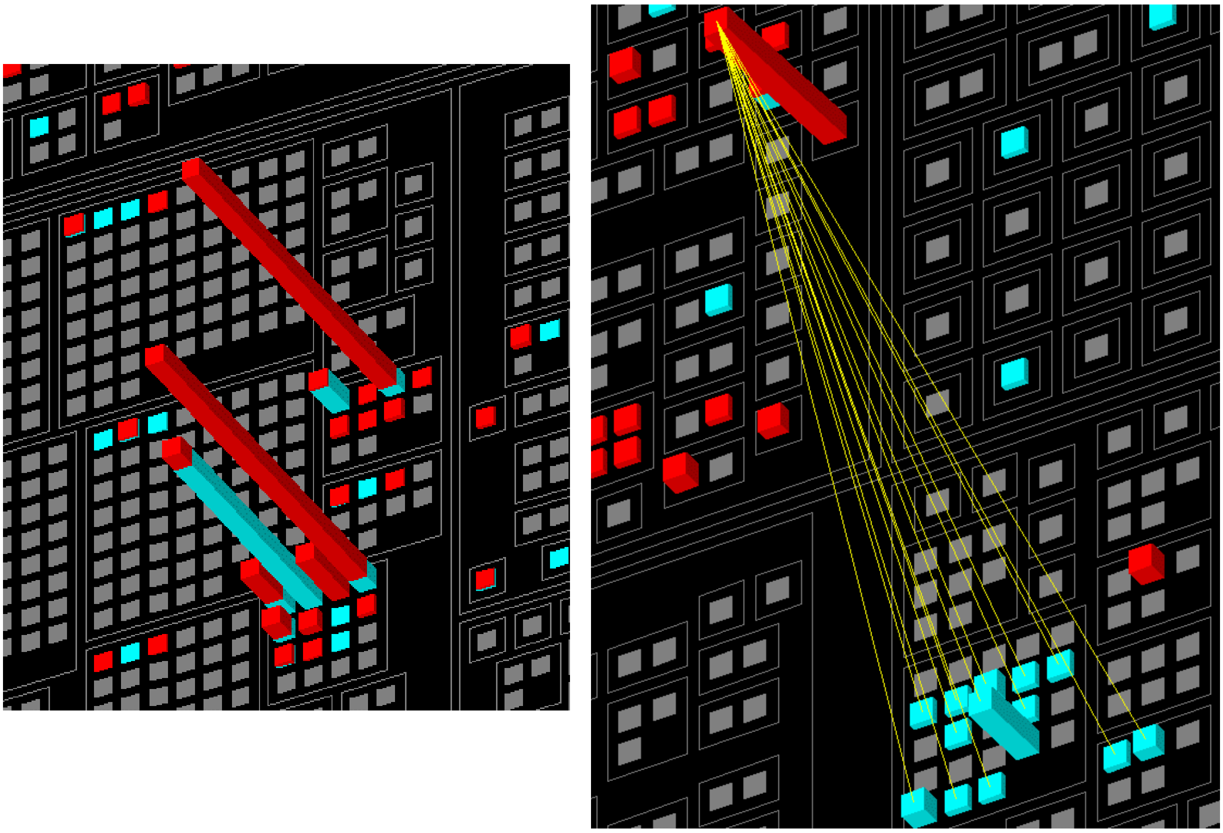


Figure 11. Closer-up views of the visualization. (left) Receivers in Figure 8(a). (right) Senders and a receiver in Figure 9.

Visualization for Computer Network and Intrusion Detection

Visualization of Internet and computer network is a very vital research topic, and therefore various works have been presented. Some works focus on representation of topology of network connection [1,2,3], and some others focus on representation of network traffic [4,5,6]. Visualization for network intrusion detection has been a vital research topic in these several years.

Takada et al. presented a technique "Mie-log"[7]. This technique horizontally divides a display space. The left-side of the display space provides bar charts of statistics of incidents per a constant time, and the right-side provides detailed information of incidents of a user-selected time span. The technique simultaneously provides global and local views, and detail-on-demand operations, for time sequence of incidents.

Logs of incidents often contain enormous numbers of error detections or of ignorable incidents. Koike et al. presented an effective visualization technique that indicates only high-danger incidents [8]. Axelesson presented a similar visualization technique combined with a possibility-based filtering system to eliminate errors or ignorable incidents [9].

Some works focused on the visual data mining of non-IDS logs to discover suspicious traffic patterns. Axelsson applied the Parallel Coordinate technique to visualize the multi-dimensional data of access log files of Web servers [10]. This technique represents various attributes of accesses in one display space, and contributes to the analysis of unknown attacks. Yin et al. also applied Parallel Coordinates to discover suspicious traffic patterns [11]. Tee Teoh et al. applied robust visual analysis technology to discover suspicious traffic from network log files, by the combination of various visualization techniques [12]. Erbacher et al. applied glyph metaphors to represent network traffic and discover suspicious accesses [13].

Some works visualize network traffic in IP-address spaces [14,15]. Their representation style simply maps values of each byte of IP-address onto horizontal and vertical axes of display spaces.

References

[1] T. Munzner, E. Hoffman, K. Claffy, B. Fenner. Visualizing the global topology of the mbone. *Proceedings of the 1996 IEEE Symposium on Information*
Visualization, pages 85–92, 1996.

[2] T. Munzner. Exploring large graphs in 3d hyperbolic space. *IEEE Computer Graphics and Applications,* 18(4):18–23, July/August 1998.

[3] B. Cheswick, H. Burch, S. Branigan, Mapping and visualizing the internet. *Proccedings of the 2000 USENIX Annual Techincal Conference,* 2000.

[4] R. A. Becker, S. G. Eick, A. R. Wilks, Visualizing Network Data, *IEEE Transactions on Visualization and Computer Graphics,* Vol. 1, No. 1, pp. 16-28, 1995.

[5] J. Pongsiri, M. Parikh, M. Raspopovic, K. Chandra, Visualization of Internet Traffic Features. *12th International Conference of Scientific Computing and Mathematical Modeling.* 1999.

[6] N. Patwari, A. O. Hero, A. Pacholski, Manifold Learning Visualization of Network Traffic Data, *2005 Workshop on Mining Network Data*, 2005.

[7] T. Takada, H. Koike, MieLog: A Highly Interactive Visual Log Browser Using Information Visualization and Statistical Analysis, Proceedings of LISA XVI Sixteenth Systems Administration Conference, The USENIX Association, pp.133-144, 2002.

[8] H. Koike, K. Ohno, SnortView: Visualization System of Snort Logs, CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC 04), 2004.

[9] S. Axelsson, Combining A Bayesian Classifer with Visualization: Understanding the IDS, CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC 04), 2004.

[10] S. Axelsson, Visualization for Intrusion Detection Hooking the Worm, European Symposium on Research in Computer Security 2003, pp. 309-325, 2003.

[11] X. Yin, W. Yuric, M. Treaster, Y. Li, K. Lakkaraju, VisFlowConnect: Netflow Visualization of Link Relationships for Security Situational Awareness, *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security,* pp. 26-23, 2004.

[12] S. T. Teoh, T. J. Jankun-Kelly, K.-L. Ma, F. Wu, Visual Data Analysis for Detecting Flaws and Intruders in Computer Network Systems, IEEE Computer Graphics and Applications, Vol. 24, No. 5, pp. 27-35, 2004.

[13] R. F. Erbacher, K. L. Walker, D. A. Fincke, Intrusion and misuse detection in large-scale systems. *IEEE Computer Graphics and Applications*, 22(1):38–48, January/February 2002.

[14] W. Yurcik, K. Lakkaraju, J. Barlow, J. Rosendale, A prototype tool for visual data mining of network traffic for intrusion detection. In *Proceedings of the ICDM Workshop on Data Mining for Computer Security (DMSEC'03)*, 2003.

[15] R. Ball, G. A. Fink, C. North, Home-centric visualization of network traffic for security administration, *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pp. 55-64, 2004.