Interactive analysis of computer scenarios through parallel coordinates graphics

Gabriel D. Cavalcante [#1], Cleber P. Souza [#2], Sebastien Tricaud [*3], Paulo L. de Geus [#4]

[#] *Institute of Computing, University of Campinas*
Albert Einstein, 1251, 13083-852 – Campinas – Brazil
[1] gabriel@las.ic.unicamp.br [2] cleberl@las.ic.unicamp.br [4] paulo@las.ic.unicamp.br

[*] *Picviz Labs*
40 Avenue Guy de Collongue, 69130 – Ecully – France
[3] stricaud@picviz.com

***Abstract*—Parallel coordinates is a well-recognized method to visualize multivariate data. It uses an n-dimensional representation to reveal correlations in multiple dimensions. A security analyst plays a key role in tackling incidents, albeit being a hard task to achieve properly: a single service can generate a massive amount of log data in a single day. Among several techniques available, parallel coordinates have been widely used for visualization of high-dimensional datasets and are also highly suited to plot graphs with a huge number of data points. Unusual conditions and rare events may also be revealed in parallel coordinates graphs when they are interactively visualized, which would be a nice feature for the analyst to count on. To address that, we developed the Picviz-GUI tool, adding interactivity to the visualization of parallel coordinates graphs. With Picviz-GUI one can shape a graph to reduce visual clutter and to help finding patterns. With a set of simple actions, such as filtering, changing line thickness and color, and selections, the user can highlight the desired information, search through the variables for that subtle data correlation, and thus uncover security issues. This article shows how we implemented these features on top of parallel coordinates graphs, using a practical example to illustrate and demonstrate the effectiveness of this approach.**

I. Introduction

Nowadays, a security administrator is responsible for the analysis of large amounts of data that represent activities on the network as a whole. System logs and network traffic can provide useful information to describe what is going on with the infra-structure. However, a single service can produce a myriad of lines of log a day. Combined with the complicated topologies of private networks, this really might constitute a overwhelming volume of data.

Normally, computer evidence is transparently created by the computer's operating system, as instructed by its operator through service configuration files. Even so, some useful information might be hidden from inspection, requiring special tools and techniques for the analyst to become aware of them [1].

A large range of software tools aid investigators in finding illegal activities on affected systems. These tools reduce tedious efforts on the part of the examiner, especially when searching for some weird event in a very large amount of data. In addition, the investigators need correlate and interpret such data, which by itself is an error-prone process that consumes an abundance of time and patience. This is especially true in the absence of a hint or particular reason to suspect anything.

Visualization techniques can help forensic specialists direct their efforts to suspicious events, processes, or hosts, through assisting the data interpretation process. Furthermore, one can draw from the visualization world how to approach the problem. In Computer Security, it is important to be able to extract useful information from sparsely clustered events without losing sight of the big picture, i.e. doing it with all data available.

Since parallel coordinates graphs can represent large datasets in multiple dimensions, handling their multivariate abilities help factor out different categories of computer security data, whenever this is needed. Picviz provides mechanisms to automate the graph creation in a simple way, and Picviz-GUI provides a powerful interactive interface to parallel coordinates. This interaction is useful in many forensic scenarios.

II. Brief explanation about parallel coordinates

Parallel-coordinates—hereafter called ||-*coordinates*— is a famous visualization technique which plots individual data across many dimensions. This is feasible because any individual data element can be described as a tuple.

Let us imagine how much different information one can grab about a type of data, for example, a soccer league table (position, team name, number of matches, wins, draws, losses, goals conceded, goals scored, number of points, etc.). To complete a table like that we must have many tuples containing a value for each of these data elements.

In a formal way, we could define each line of the soccer table as a vector $\vec{v}$ described as a tuple $(x_1, x_2, x_3, \ldots, x_n)$ which could be represented on an N-dimensional plane $\mathbb{R}^N$. Unfortunately, when n becomes bigger than 3 it is difficult to plot n-dimensional vectors using a 3-dimensional physical space.

WIFS'2011, November, 16-19, 2011, Foz do Iguaçu, Brazil.
978-1-4244-9080-6/10/$26.00

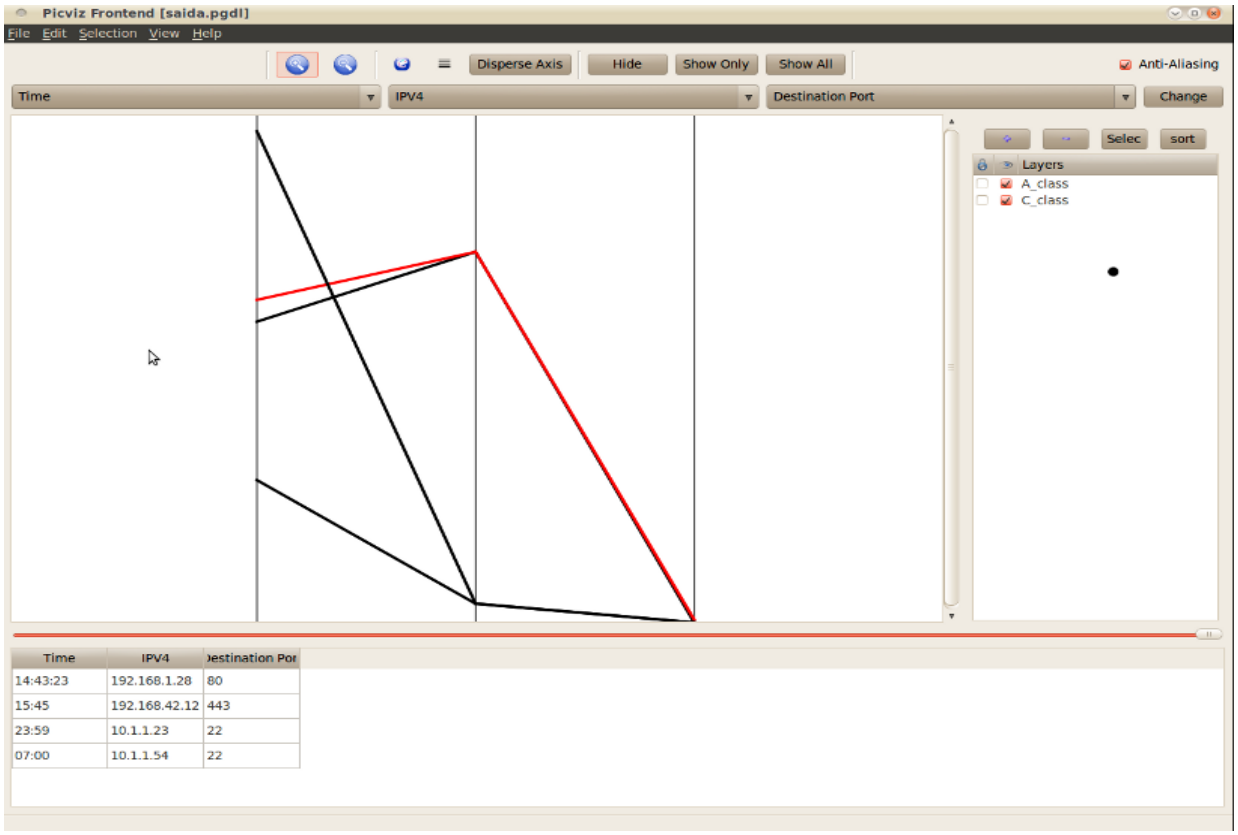


Fig. 1: Picviz-GUI showing the example described in Section III-A with layers.

||-*coordinates* overcome the 3-dimensional space limitation by adding a parallel axis to each dimension of $\vec{v}$. As such, the vector $(x_1, x_2, x_3, \ldots, x_n)$ is visualized by plotting x_1 on axis 1, x_2 on axis 2, and so on through x_n on axis n. In this way, each $\vec{v} \in \mathbb{R}^N$ is represented by a polygonal line $\bar{V}$.

As goes the old say, a picture is worth a thousand words, so one had better see an image than read information. Figure 1 shows a trivial representation of ||-*coordinates* to a single extract of network activity. It uses a set of parallel lines (not necessarily vertical), where each one corresponds to an axis of data from the table. Each row of the table is then represented by a polygonal line across these axes, such that the point at which it crosses each axis corresponds to the value of the tuple's data point on that axis. The data plotted on this example is borrowed from Section III-A. For additional understanding and information we highly recommend literature such as [2] and [**?**].

III. Creation of Parallel Coordinates

||-*coordinates* being commonly used for visualizing multivariate data, recent work has evaluated how efficient this kind of visualization can be for different tasks, in comparison with other multidimensional techniques[**?**][**?**].

Picviz[3] is a suite of tools that automate the creation of ||-*coordinates* images, either from logs or any other data representation, though we are mainly concerned with data from log files here. To transform data into a ||-*coordinates*plot image, the user first needs access to the data that will be plotted.

From the security analysis viewpoint, one needs to analyze the largest possible amount of data that can describe something about the whole scenario. By following the lead provided by a user's complaint, for instance, the analyst can focus on specific data that might be related to suspicious events. For example, if a web browser has an abnormal behavior, related events might be found on the web browser log.

Typically, important data result in large amounts of information, and to deal with it we can use a process called **parsing**. Parsing, the process of identifying individual tokens of information, takes here an extra meaning, since data coming from different sources may be combined to return a single output. The parsing process in the Picviz workflow should transform captured logs into the Picviz Graph Description Language (PGDL).

A. Picviz Description Language – PGDL

From logs to the graph, the PGDL language aids in structuring the information, allowing users to define how data will be treated on graph creation through three main sections: header, data and axes. The header section defines the title of the graph, whereas the axes define how data will be represented on the graph, describing each axis of the ||-*coordinates* graph and their type. The data section contains all events that will be represented in the graph; each line must comply with the

format specified in the axes section. In PGDL, each dimension is represented by an axis, which is defined in the axes section:

```
header {
        title = "See the different groups of IPs";
}
axes {
    timeline t [label = "Time"];
    ipv4     i;
    integer  dport [label = "Destination Port"];
}
```

The keyword before each axis name defines the way data shall be placed on the axis. It is similar to variables in programming languages: integer, short, string, timeline, enum etc. The types described in the axes section for each variable are tips used by the Picviz engine for the graph construction process; for example, the engine will predict 2^{32} possible positions for the integer axis in the example above. Once the axes section is defined, data must be inserted into the data section as shown below:

```
data {
 layer C_class{
  t="14:43:23", i="192.168.1.28", dport="80";
  t="15:45", i="192.168.42.12", dport="443" [color
      ="#FF0000"];
 }
 layer A_class{
  t="23:59", i="10.1.1.23", dport="22";
  t="07:00", i="10.1.1.54", dport="22";
 }
}
```

"Data" is the original data from the log file, with dimensions being comma-separated. If needed, the subsection "layer" could be added to the data section to describe layers. It is a good way to ease the search for relevant information.

B. Picviz-GUI

As mentioned in the introduction, the ||-*coordinates* graph really shows its potential when it is used interactively. An interactive frontend was written for this purpose (see Fig. 1). Picviz-GUI is responsible for the creation and representation of ||-*coordinates*. It was written in Python and Trolltech's QT4 library to provide a skillful interaction toward finding relations among variables, allowing one to apply filters, drag the mouse over the lines to see information displayed, change axis order on the fly, zoom, brush lines and change line thickness.

One of ||-*coordinates* disadvantages is the choice of a suitable order for the axes intended to display relationships among variables. There has been a variety of proposed automatic solutions to select the best sequence of axes in ||-*coordinates*. However, greater satisfaction comes with user manipulation of the axes order[4]. The top bar allows the user to reorder axes on the fly, which provides for an effortless search for the best representation for the data under analysis; this is done without any modifications to the parser. All of Picviz suite tools are open-source under GPLv3 and can be obtained from [5][1].

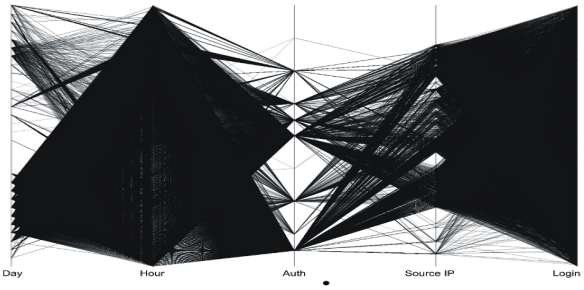


Fig. 2: Plot of raw data without any analysis.

IV. SSH AUTHENTICATION LOG ANALYSIS

SSH (*Secure Shell*) is widely used to provide remote access to a Unix-like host. Its log files keep track of successful and failed login attempts, time, status and source IP address. Patches and specific configuration and debugging options may extend the service to also track passwords, usernames [6] and detailed connection options, used normally for debugging or special purposes.

Each log analysis requires from the analyst the knowledge about what is provided by the logging mechanism for that service and how to tune it if necessary. A traditional log analysis scarcely allow the detection of anomalies such as distributed brute-force scan attempts, because critical data are sparsely distributed over long log files encompassing many days, which helps attack activities to pass unnoticed.

Fig. 2 plots the day, hour, auth (authentication status), source IP and login axes for each entry of an entire month of a SSH server log file. The careful selection of the right axis and the data plot order depends on the information that needs to be retrieved and how the analyst wants to view the information. Layers and coloring features help group suspicious data and provide him/her with a range of possibilities to look into the data for intelligence gathering.

The first step after plotting raw data is to try and detect concentration areas in order to isolate events in layers. On the bot of `Day` axis, the most active days could be noted.

Another good point to start is the `Auth` axis, which describe the status about each remote login attempt. These may indicate abuses in that service as noticed on the auth and login axes in Fig. 2, related primarily to authentication denied status for "Invalid user" and "User not allowed" attempts. By isolating these candidates it was possible to confirm the existence of two types of massive login attempts.

By simply picking a color for each of most active days, the first type is clearly identified: an individual source IP address tries many combinations of usernames and passwords in a short time window, as dettached in Fig. 3; an attack that was probably performed by some existing automated tool. This method is not very effective from the attacker's point of view because traditional defense tools are prepared to implement deny rules for these cases, since the high activity performed

[1]A professional version of Picviz suited to handle even larger amounts of events can be obtained from www.picviz.com

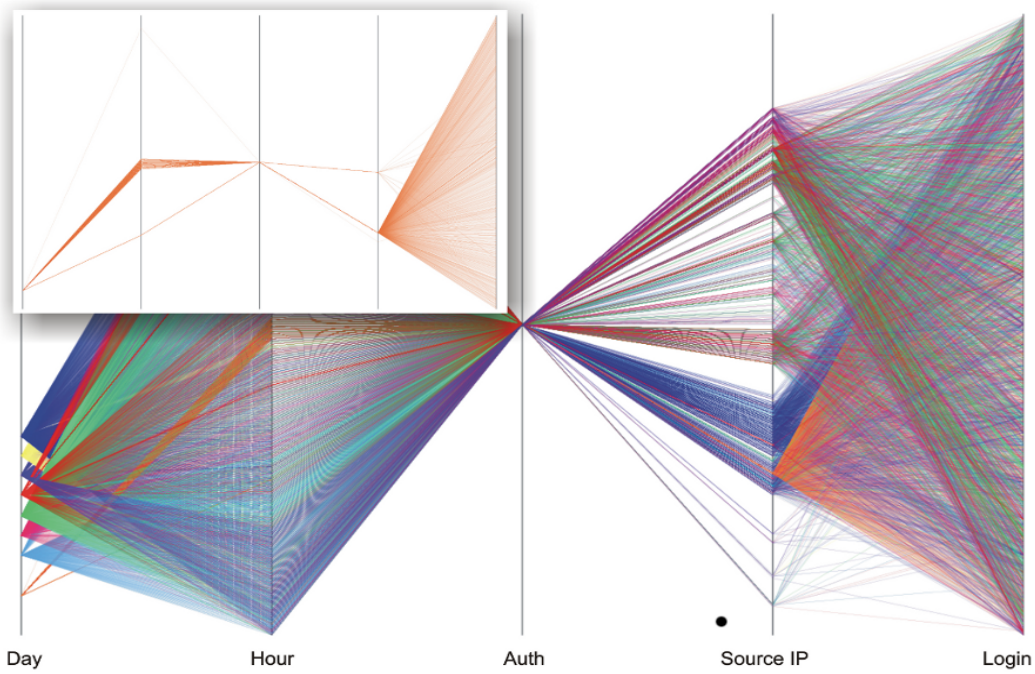


Fig. 3: Graphic showing a layered version of the data for authentication errors by unknown logins. Highlighted is a detected single source IP address SSH brute-force attack.

by a single IP source address [7] is easily identified by simple heuristics. On this point of analysis, only authentication errors were selected to be plotted in this excerpt.

As for the second type, many source IP addresses try combinations of usernames and passwords, but do so by implementing special evasion techniques.

The suspect is confirmed with details in Figure 4a, where a good concentration of red (day 7) events could be observed. The red day have a massive amount of login attempts distributed over different ips, a closer look in the top half of `Hour` axis reveals a "coordinated stop" of red login attempts.

On the same picture, the light blue events (day 4) demonstrates the same coordinated behavior. It is a bit difficult to see, but with a detailed look in the bot of day axis confirms the same suspicious stop in the light blue login events.

This property allows the conclusion that a third element on the network was coordinating the login attempts by a variety of sources, a behavior that characterizes the usage of botnets[2]. Not only that, the coordination required is much more elaborate than previous botnets have shown in the past:

- A good variation of login ids, representing a coordinated dictionary attack, unfortunately the server was not prepared to record passwords in failed logins (that could help understand dictionary distribution over the bots);
- At most two attempts are launched each day from the same address.

In the Fig. 4b that add some days when in Fig. 4a, demostrates an effective grow in quantity of ips attempting to access the SSH server (in the bot of `Source IP` axis).

Filtering the ip axis to show only ip starting with 200 (*200.**) and modifying the ip axis configuration (to spread ips along the axis) results on Fig. 5, that reveal the uniqueness of the attack. The highlight in Fig. 5 shows an absence of large fan-outs from each point on the `Hour` and `Source IP` axes, when compared to the detailed view in Fig. 3. With different colors for each day, we could easily see that many ips have one on the most two login attempts in the same day. It represent a very stealth scanning, that subvert standard configurations of defense tools like Denyhosts[3] and Fail2ban[4]. Distributed attacks and its derivations are difficult to detect and visualize with traditional defense tools [7].

PicViz-GUI was capable of showing a full view of the distributed SSH brute-force attack that was passing unnoticed by other means, revealing that present defense tools need improvements to detect and block this new approach. Shortly after this analysis was performed, news were seen [5],[6] of large botnets actively exploiting a vulnerability in older versions of the phpMyAdmin[7] tool (described in the *Common Vulnerabilities and Exposures* database under the number CVE-2009-1151).

By exploiting this vulnerability—in older phpMyAdmin

[2]A group of compromised machines that keep their normal funcionality for valid users but are simultaneously serving a remotely commanding attacker.

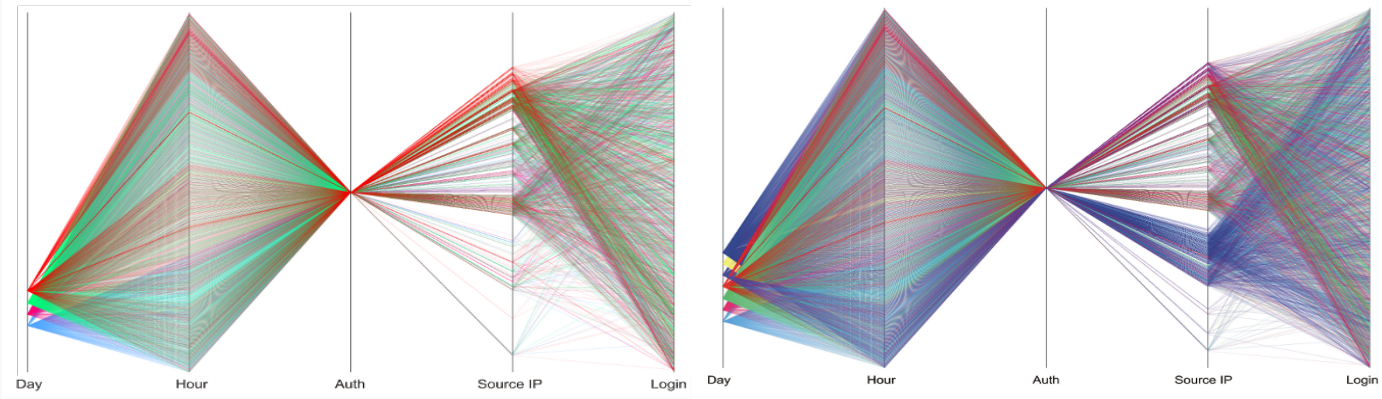
[3]http://denyhosts.sourceforge.net/

[4]http://www.fail2ban.org/

[5]Botnet Trend: phpMyAdmin & SSH Attacks http://www.malwarecity.com/community/index.php?showtopic=1177

[6]SSH - new brute force tool http://isc.sans.edu/diary.html?storyid=9370

[7]phpMyAdmin http://www.phpmyadmin.net/



(a) Selected days: 4, 5, 6 and 7.

(b) Selected days: 4, 5, 7, 8, 9 and 10

Fig. 4: SSH scanning over a six-day period.

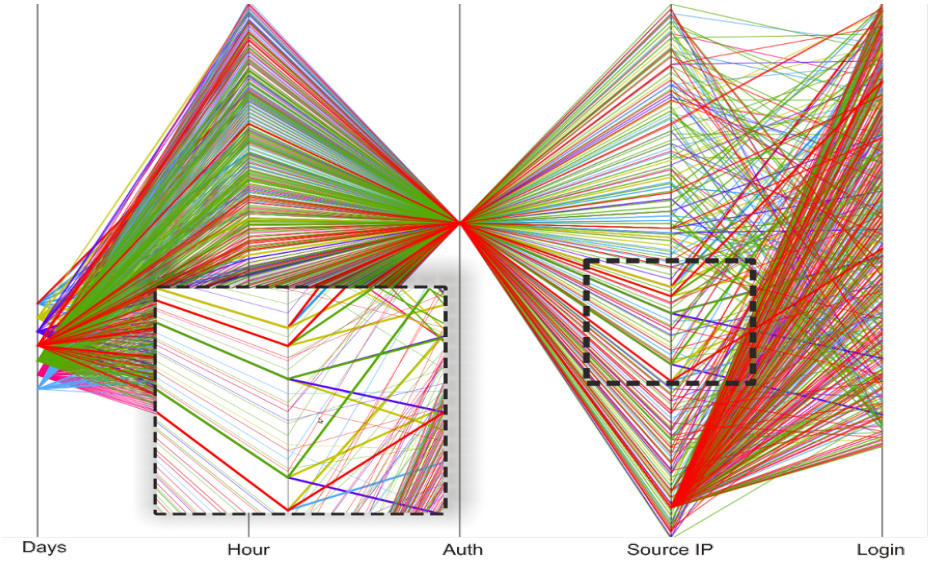


Fig. 5: Isolation of IPs from the 200 prefix range shows that individual hosts are trying remote accesses once a day (each day was colored differently).

versions—attackers were capable of executing malicious code on vulnerable hosts. The botnet herders exploit this vulnerability and upload a nasty bot named "dd_ssh"; this bot then conducts a brute-force SSH scan attack on random IP addresses under the command of the herder. Also noticed on the host were many HTTP requests looking for vulnerable phpMyAdmin instances, as described in the sanitized excerpt below:

```
202.X.X.X - - [10/Jun/2010:05:20:10 -0300] "GET ///
    scripts/setup.php HTTP/1.1" 404 215 "-" "ZmEu"

202.X.X.X - - [10/Jun/2010:05:21:56 -0300] "GET //
    phpMyAdmin//scripts/setup.php HTTP/1.1" 404 227
    "-" "ZmEu"

202.X.X.X - - [10/Jun/2010:05:22:51 -0300] "GET //
    pma//scripts/setup.php HTTP/1.1" 404 220 "-" "
    ZmEu"
```

The increasing number of source IP addresses noticed over the days in the plot is related to the number of hosts successfully infected by the CVE-2009-1151 vulnerability.

Picviz identified this new distributed SSH brute-force attack that went stealth to traditional defense tools. This tool allows quick identification of complex attack patterns that are not easily detectable and so allows for quicker mitigation and development of countermeasures.

V. Conclusion

This article described an interactive tool to analyze and document computer security incidents through ||-*coordinates*. Picviz provides simple mechanisms to interpret data from large data sets, thus allowing for an easier identification and understanding of complex events, such as those present in security log files.

We tested the tool on log files from the SSH service running on a host that offers remote access service to its users. Due to the amount of attacks directed to this service and constant new approaches employed by attackers to evade detection, these logs supplied the required data and at the same time buried the critical information that a security analyst should find. The data used were, however, good enough to explore the power of visualization tools when applied to security issues.

||-*coordinates* provide a simple way to recognize information, whereas Picviz enhance that ability. Security log file information was processed to detect attack patterns against the SSH service that were not easily recognizable. The tool allowed the detection of both simple brute force scanning attacks and more sophisticated attacks launched through botnets. The latter could not be detected without a visualization tool that brought all the log information in a unique and consolidated view. The attackers' behavior noticed during the analysis is of substantial importance for the improvement of tools aimed to detect and block such attacks.

Picviz is however considered at an early stage in its development, with some features still missing, especially in the realm of finding correlated events automatically—for example, it would be useful to highlight the same IP on different layers to find a successful attack. As it is at the moment, though, we have shown its strength while trying to find rare and unusual events in computer security. This capability is of paramount interest in computer forensics-related activities such as responding to a computer security incident, by accelerating identification and possibly, through a deeper understanding, mitigation of future attacks of the same kind.

REFERENCES

[1] W. Kruse and J. Heiser, *Computer forensics: incident response essentials*. Addison-Wesley, 2008.

[2] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *Proceedings of the 1st conference on Visualization'90*. IEEE Computer Society Press, 1990, p. 378.

[3] S. Tricaud and P. Saadé, "Applied parallel coordinates for logs and network traffic attack analysis," *Journal in computer virology*, vol. 6, no. 1, pp. 1–29, 2010.

[4] J. Yang, W. Peng, M. Ward, and E. Rundensteiner, "Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets," 2003.

[5] (2010) Picviz homepage. [Online]. Available: http://www.wallinfire.net/picviz

[6] D. Ramsbrock, R. Berthier, and M. Cukier, "Profiling attacker behavior following ssh compromises," jun. 2007, pp. 119 –124.

[7] J. Thames, R. Abler, and D. Keeling, "A distributed active response architecture for preventing ssh dictionary attacks," apr. 2008, pp. 84 – 89.