I Live and work in Paris (FR)
Happy Linux user since 1995
I work for INL as CRO:
  • The company (www.inl.fr), not the lab (www.inl.gov)
  • We work on Netfilter
  • We develop NuFW (GPL) and differentiate users from IP addresses
  • You define what each group is allowed to access, and NuFW enforces it at the network layer
  • We know which packets a given user sent
Lead the French Honeynet project
Developer of Linux PAM, Prelude IDS, OSSEC, Wolfotrack and Picviz

<stricaud@inl.fr>
What are logs?

Syslogs
Nov 6 13:12:04 quine avahi-daemon[2285]: Interface eth0.IPv4 no longer relevant for mDNS.
Nov 6 13:12:06 quine ifplugd(eth0)[1811]: Program executed successfully.
Nov 6 13:12:06 quine kernel: ADDRCONF(NETDEV_UP): eth0: link is not ready
Nov 6 13:12:24 quine kernel: Unhandled event received : 0x50

Database
sql> SELECT * FROM logdb WHERE user = "ptc";

Network
08:50:01.522077 arp who-has 10.0.0.254 tell 10.0.0.1 08:50:01.522115 arp reply 10.0.0.254 is-at 00:69:de:ad:be:ef
08:50:01.522210 IP 192.168.0.1.5860 > 172.16.17.235.333 73: UDP, length 25
08:50:01.522377 IP 192.168.0.1.5860 > 10.30.254.247.18946: UDP, length 25

Others
stderr, binary/text file, . . .
What (normal) people do with them?

They grep

grep -i "segmentation fault" /var/log/*

They watch

tail -f /var/log/messages

They use tools

OSSEC\textsuperscript{a}, Prelude LML\textsuperscript{b}, Sisyphus\textsuperscript{c} \ldots

\begin{itemize}
  \item \textsuperscript{a}http://www.ossec.net
  \item \textsuperscript{b}http://www.prelude-ids.org
  \item \textsuperscript{c}http://www.cs.sandia.gov/jrstear/sisyphus/
\end{itemize}

They even correlate!

http://security.ncsa.uiuc.edu/research/mithril/Mithril.html
What (normal) people do with them?

They visualize

http://www.secviz.org

They even do communities!

http://www.secviz.org
Actual issue

- A lot of information
- Syslogs are unstructured
- Human interaction needed after the problem
- When automated, needs signatures (usually pcre based)
- Overwhelming a single machine

\[^1\text{yeah, it is not fixed yet, wait for WASL2009}\]
Picviz and Honeynet

Typical low-interaction honeypot setup

Nepenthes → var/log/nepenthes/logged_submissions

var/log/nepenthes/logged_downloads

Snort → /var/log/snort/alert

SSH authentication → /var/log/auth.log (Debian Linux)

Auditd → /var/log/audit/audit.log
220574 lines of logs in total

- This is a log overdose
- Most people are happy just to extract known patterns
- The French honeynet chapter is full of busy (lazy?) people
- Keep the fun where it is, avoid log file slavery
Deal with logs a better way. Use Picviz, that:

- Creates a picture of your logs
- Does not interpret anything, just displays logs as they are
- Is not signatures based
- Can deal with an infinity of events
Picviz

Moto

"Finding a needle in a haystack... when you don’t even know how the needle looks like"
Picviz

Moto

"Finding a needle in a haystack... when you don’t even know how the needle looks like"

To generate pictures like this
1. Introduction

2. Parallel Coordinates

3. Picviz

4. Analysis
Parallel Coordinates

||-coords are

Picviz finding a needle in a haystack

Usenix, San Diego 2008
Parallel Coordinates

Inventors

Invented by Maurice d’Ocagne in 1885

Applied by Alfred Inselberg in 1959

- Senior Fellow San Diego Supercomputing Center and Computer Science and Applied Mathematics Departments Tel Aviv University, Israel
- Conflict Resolution, One-Shot Problem and Air Traffic Control, 1st Canadian Conf. on Comp. Geom., 1989, 26-9
Parallel Coordinates \textbar\textbar-coords introduction

\textbf{-coords}

\[ \vec{u} = (0.6, 1.6, -0.8, 1.2) \in \mathbb{R}^4 \]

Properties

- N-dimensions: one axis per dimension
- Axes are equidistants
- \( \infty \) of events: one line per event
- Lowest value at each axis bottom
x and y are linked by an affine relationship \( y = \alpha x + \beta \)
Today's objectives

Apply $\parallel$-coords to logs:

- Focus on security
- See if by doing this we succeed in finding things
1 Introduction

2 Parallel Coordinates

3 Picviz

4 Analysis
Picviz goals

- Help to generate |-coords images
- Scalable architecture (filters, real-time, . . . )
- Provide an interface to query lines and reorganize axes
- Mainly security oriented
Picviz world

Three main parts

- **Perl scripts**: Transforms your logs into Picviz graph description language (PGDL)
- **pcv**: CLI to transform PGDL into an image
- **picviz-gui**: Frontend

Code architecture
Global architecture

Picviz finding a needle in a haystack
Use

PGDL source

header { title = "Usenix WASL 2008"; }
axes {
    timeline t;
    integer in;
}
data {
    t="14:42", in="12" [color="red"];  
    t="14:45", in="432";
}

Generate the image

```bash
pcv -Tpngcairo file.pcv 'filter' > out.png
```
Axes

Types

- Time: timeline, years
- Numbers: integer, short, gold, char
- Addresses: ipv4, ipv6
- Strings: string
- Specials: enum, ln

Properties

- relative: to place data relatively to each other
- print: to turn off data value printing
- label: display this name
Strings

- The hardest variable to place
- Two algorithms can be chosen:
  - Basic: Ascii value addition and place the string compared to a famous quote\(^2\)
  - Prefix: strings are placed collision-safe with their first 4/8 characters (prefix size is architecture dependent)

\(^2\)The competent programmer is fully aware of the limited size of his own skull. He therefore approaches his task with full humility, and avoids clever tricks like the plague.
## Enumerations

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Enumeration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00</td>
<td>Éric</td>
</tr>
<tr>
<td>11:00</td>
<td>Charles</td>
</tr>
<tr>
<td>12:00</td>
<td>Joe the plumber</td>
</tr>
<tr>
<td>13:00</td>
<td>Utroo</td>
</tr>
<tr>
<td>14:00</td>
<td>Pablo Picasso</td>
</tr>
<tr>
<td>15:00</td>
<td></td>
</tr>
<tr>
<td>16:00</td>
<td></td>
</tr>
<tr>
<td>21:00</td>
<td>Joe le lutin</td>
</tr>
</tbody>
</table>
Lines

Properties

- color: line color
  - red
  - #ff0000
  - (1,0,0)
- penwidth: line width

Why a custom format? why not CSV?

- Flipping the axis order is as simple as moving the axis declaration order
- Line properties are already computed by generators
- Actually CSV can be used as input, it is simply converted into PGDL
Some CLI options

• `-r..r`: Increase the image height and width
• `-a`: Display lines values
• `-Ln`: Display value every n lines
• `-Tplugin`: Output plugin
• `-Rplugin`: Rendering plugin
• `-Astuff`: Plugins argument
Filter

- Plot filtering: show plot > 250 on axis 2
- Plot percentage filtering: show plot > 50% on axis 2
- String filtering: hide value = ".*[fF]oo.*" on axis 1

Eg.: Display only lines going < 10% on the axis 2 and carrying the value "denied" on the axis 4

```bash
pcv -Tpngcairo fichierlog.pcv 'show plot < 10% on axis 2 and value = "denied" on axis 4' >filtered.png
```
Frequency analysis

- The more an event appears, the higher the frequency is
- Break lines color to do a gradient
- from green (low) to red (high) via yellow (medium)
- Two modes:
  - Axes pair (standard)
  - Infection (virus)

Create an image with the **virus frequency** analysis mode

```
pcv -Tpngcairo -Rheatline -Avirus file.pcv > out.png
```
Let's see my syslog in `-coords`

We run

```bash
syslog2picviz.pl /var/log/syslog* > syslog.pcv
pcv -Tpngcairo syslog.pcv > syslog.png
```

We have

**red** = kernel logs
Start Picviz with a socket to listen at and a template to use
pcv -Tpngcairo -s local.sock -t samples/test1.pcv -o out.png

Client
echo "t='12:00’, i=’100’, s=’Hello, World!’;" > local.sock
1 Introduction

2 Parallel Coordinates

3 Picviz

4 Analysis
Nmap

Command line

```
pcv -Tpngcairo nmap-scan.pcv -Rheatline -r >nmap.png
```
Nmap: only lowest ports

Command line

```
pcv -Tpngcairo nmap-scan.pcv -Rheatmap -r 'show plot < 5% on axis 5'
>nmap2.png
```
OpenVPN Traffic

<table>
<thead>
<tr>
<th>timestamp</th>
<th>source</th>
<th>sport</th>
<th>destination</th>
<th>dport</th>
</tr>
</thead>
</table>

Picviz finding a needle in a haystack
SSH authentication
Detect a weird behavior

It is sometimes simple to automate a behavior we don’t want that
||-coord helped to see.

• Based on SSH authentication log, We alert the administrator if:
  • Many different IP log on the same account
  • If a user authenticated in different manners
  • A login IP address matches the Dshield database

• http://www.wallinfire.net/files/artcore.pl

---

3http://www.dshield.org
SSH scan

PGDL source

time="05:08", source="192.168.0.42", log="Failed keyboard-interactive/pam for invalid user lindsey";
time="05:08", source="192.168.0.42", log="Failed keyboard-interactive/pam for invalid user ashlyn";
...
Botnet
Analysis objectives

On my webserver, Apache access.log has 412429 lines:

1. How to easily understand those logs?
2. How to detect attacks?
Create the picture

Generate the PGDL

perl apache-access2picviz /var/log/apache2/access.wal linfire.net.log >access-wal linfire.net.pcv

Generate an image with frequencies, high resolution + text

pcv -Tpngcairo -Rheatline -Avirus -rrrrrrrra access-wal linfire.net.pcv >access.png
Result
Filter weird urls

Generate an image with frequencies, high resolution, text + filter

```
pcv -Tpngcairo -Rheatline -Avirus -rrrrrrrara 'show plot > 50% on axis 4'
access-wallinfire.net.pcv >urls-abnormals.png
```
Result

![Network Diagram]

Sébastien Tricaud (INL)
Every IP is suspicious

We take to easy to read IP: 213.192.60.19

$ host 213.192.60.19
19.60.192.213.in-addr.arpa domain name pointer gw9.vslesy.cz.
Who is it?

- We search on http://www.dshield.org: nothing
- We search on Google: *tada*
Roadmap

- 0.5 version going to be released very soon
- Windows port, anyone?
- Add more frequencies types
- Share the work among several machines
- More work is needed on the frontend
- Divider type, to split a string into several axes and put more than an axis per variable
Questions?

- Email: stricaud@inl.fr
- Blog: http://www.gscore.org/blog
- Get the sources: svn co http://www.wallinfire.net/svn-picviz