Data visualization for discovery, analysis and presentation of digital evidence

IMT4022 Digital Forensics II
ANDRÉ NORDBØ[*]

Abstract

In this project we are going to answer how visualization techniques can help a digital forensics analyst in finding evidence in digital data by looking at what makes visualization effective, challenges and possible solutions to dealing with huge amounts of data and examples of different tools and methods used in the domain. The methodology will be literature study.

1. INTRODUCTION

In this paper we look at how visualization techniques can be used in the forensics process of identification, analysis and presentation when dealing with digital data in order to make these processes more efficient. The amount of information available, even on a single device is enormous and increasing, hence the need for aiding the human investigator in finding and categorizing evidence in an efficient way. The methodology used in this paper will be literature study in order to highlight the current situation and to get a grasp on the challenges faced.

Before we move on to the research questions, lets look at the terminology digital forensics and visualization. *Digital forensics* is using scientific methodology to find evidence in digitally stored information with the purpose of supporting or refuting a hypothesis related to incidents governed by criminal or civil law. Figure 1 illustrates one way to think of the forensics process, and as marked in the illustration, the main focus here will be on the first and last two steps.

As explained in [Aigner et al., 2011, page 44] tree main reasons why to visualize is to explore, confirm and present. Exploring is important in the identification stage, and both exploring and confirming is important in the analyze stage. Presentation focuses on communicating a story like how to make the judge and jury understand the often deep technical information presented at court.

Visualization, as introduced in [Aigner et al., 2011, page 3] is to form a mental image of something, transforming symbolic to geometric. One main reason is to see the unseen, like when scientists started mapping non-visible light into the visible range in order to see infrared heat radiation and background radiation from distant stars. Visualization has been an independent research field for about 20 years, and it's goal is to

> "Integrate the outstanding capabilities of human visual perception and the enormous processing power of computers"[Aigner et al., 2011]

This symbiosis between man and machine is very interesting, and Shyam Sankar[Sankar, 2012] in this talk explains the importance of not only focusing on algorithms, but to create processes and interfaces that maximizes the cooperation of humans and machines, as envisioned by Joseph Licklider more than 50 years ago. Machines exceed at doing easy tasks fast, and trying billions of combinations. Humans have curiosity, creativity, intuition, understanding of context, and this is why we should not try to instruct a computer to do all the work alone. The interface is important and visualization is a way to make humans understand and contribute to solving problems.

[*] Gjøvik University College (12HMISA)

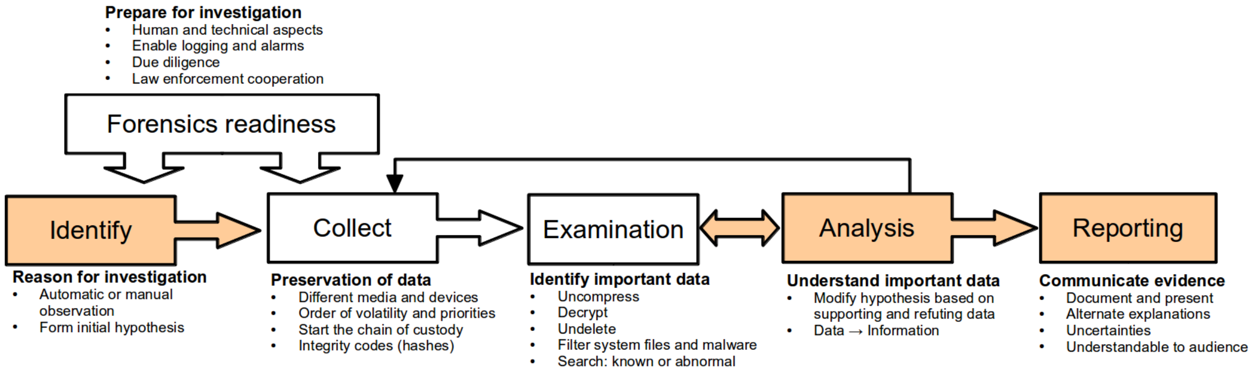

Prepare for investigation
Human and technical aspects
Enable logging and alarms
Due diligence
Law enforcement cooperation
Forensics readiness
Identify
Collect
Examination
Analysis
Reporting
Reason for investigation
Automatic or manual observation
Form initial hypothesis
Preservation of data
Different media and devices
Order of volatility and priorities
Start the chain of custody
Integrity codes (hashes)
Identify important data
Uncompress
Decrypt
Undelete
Filter system files and malware
Search: known or abnormal
Understand important data
Modify hypothesis based on supporting and refuting data
Data → Information
Communicate evidence
Document and present
Alternate explanations
Uncertainties
Understandable to audience

Figure 1: *An example of a forensics process based on NIST SP800-86 and lectures in digital forensics I/II. This paper focuses on the identification, analysis and reporting phase of this model and how visualization can help in these phases.*

These are the research questions guiding the rest of this paper:

1. What makes visualization effective?
2. Do digital forensics and visualization have any challenges in common, and what is being done about it?
   - How can machine learning and pattern recognition help in visualization?
3. In what situations are visualization being utilized in digital forensics and what tools exists?

2. Main content

In the following sections each of the questions asked in the introduction will be discussed.

2.1 Effective visualization

As briefly explained in the introduction, visualization is a way of utilizing the power of the human cognitive systems for exploration, verification and presentation, and in this section we take a deeper look into what makes visualization effective - important things that separate good from bad visualizations. The book Visualization of Time-Oriented Data[Aigner et al., 2011] and paper[Aigner et al., 2007] focuses on data sets that have a temporal (time) dimension and how it must be treated differently than ofter variables because of it's unique nature. Time is also a major factor in forensics dealing with cause and effect. The authors in the cited book introduces 3 important aspects for visualizations early on: **what**, **why** and **how**.

2.1.1 The what

Knowing your data and it's properties. A famous quote from Edward Tufte:

> "Above all else show the data."

and it summarizes the essence of letting the observations speak for themselves, and not letting design come in the way.

Some examples of properties mentioned in the book:

- Temporal data can be point or interval based (have scale). A point or event is a single observation and an interval has a start and end value. The time points can also be ordinal: before, during or after a known event.
- Temporal data can be linear, cyclic or branching. One example from the book of branching time can be eye witness reports of the same time span with different, perhaps contradicting stories. Example of cyclic can be slow port scanning of a network using a long period between probes.
- Temporal data can have different uncertainties in them, for example because of different time zones, logs using local time while others use UTC time, and the machine time itself can be out of sync because of drift or manipulation. The same goes for spatial (space / location) data: A country name, a street address and a GPS position (latitude, longitude, height) can have different levels of positional uncertainty.

2.1.2 The why

What the goal of visualization is. Examples mentioned are *exploring*, *confirming* and *presenting*.

Exploring or discovery is looking for something you still don't know. Norman's model in [Aigner et al., 2011, page 109] was used to illustrate this concept. Applied for visualization you have some data, visualize it, spot something interesting, change the parameters and redraw the visualization. A summary of interaction techniques:

- *Explore* and *filter* by changing combination of narrowing down variables. Related to this is the level of abstraction as exemplified by how more details are typically shown when zooming in on time in a calendar or in location on a map, or when removing known good or good bad files based on hash values

  > "Overview first, zoom and filter, then details-on-demand"[Shneiderman, 1996]

- *Reconfigure* as exemplified by changing between linear or cyclic representation of time. Changing *encoding* like timezone, time format
- *Mark* or select interesting points or areas, save for later or use in new queries. A good demo of this property can be found at [1].
- *Connect* or relate: showing similar or otherwise related information.
- *History* is allowing for undo and redo actions, remembering user actions.

Confirming hypothesis is related to exploring based on previous knowledge. One could argue that because the parameters are known, the need for the above mentioned interaction techniques is less relevant.

Presenting is when you have found something interesting, and the goal now is to convince others. Noah Iliinsky[Iliinsky, 2012] points out the difference between data visualization and infographics. Infographics are designed posters manually created using data[2] and might be what you want to show in court. Data visualization is often simpler and automated so that it scales with larger and changing data. Good for exploring data and could therefore align with what's needed for the identification and analysis steps in digital forensics.

Noah also talks of making data accessible and how visualization makes it possible to consume lots of data that would otherwise be difficult to comprehend in list or table form. The human visual system is evolved to spot trends and patterns violations such as outliers and gaps. He also talks of the difference between exploring data, where the story is still unknown, compared to explanation focusing on communication and divides it up into whether education or persuasion is the goal and presents a model of the data, designer (creator) and the reader (consumer) in figure 2 and how important it is to understand the data, what your intentions are and the readers' situation as your success at visualizing is determined by the readers ability to consume it.

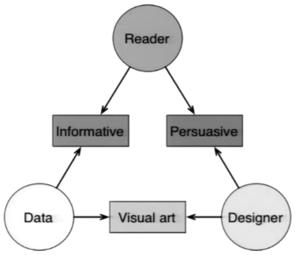


Figure 2: *Model of data, designer and reader, from speaker slides[Iliinsky, 2012].*

2.1.3 The how

How to transform the data into something visual. It's important to understand the benefits and disadvantages of how this is performed. Time can be mapped statically to space like in a time line or it can be mapped dynamically to time itself like an animation freeing up a dimension for other variables, but at the same time making comparisons more difficult. It's also important to know how to leverage properties such as position, length and size that the human mind is very good at understanding, and choices related to 2D versus 3D, especially related to interaction.

Noah goes into detail on different ways to encode data visually like position, size, angle, color and shape[3] in terms of how useful they are for different purposes like ordered, ordinal, categorical and relational data. He brings up a

[1] JavaScript demo of marking data `http://mbostock.github.io/d3/talk/20111116/iris-splom.html`

[2] Examples by Fabian Fischer `http://pinterest.com/erikaiv/timeline-infographic/`

[3] His table of visual encodings: `http://complexdiagrams.com/wp-content/2012/01/VisualPropertiesTable2.gif`

quote by Moritz Stefaner:

> "position is everything, color is difficult"[4],

and illustrates it by how color has so many cultural meanings. Blue is cold, red is warm, pink is girl, blue is boy, and imagine a dashboard where emergency is depicted with green while normal status is depicted by red. Anther thing to watch our for is color blind readers.

Both Noah and also David McCandless[McCandless, 2012] talk of the importance of being able to show relative data. One example as shown in figure 3 is plotting states in the U.S. colored by votes. Plotting by space will lead to a skewed visual impression because of differences in population and land mass. A better way would be to alter the map so that the area of the state is proportional to the number of inhabits. McCandless also highlights examples like comparing military budget to the population instead of per country.

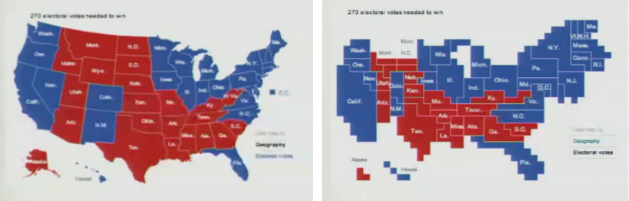

270 electoral votes needed to win
Geography
Electoral votes
270 electoral votes needed to win
Geography
Electoral votes

Figure 3: *Example of how a geography map is bad for illustrating political votes. Left: geographical correct map. Right: Map adjusted by electoral votes, a function of population[Iliinsky, 2012].*

For a survey of visualization techniques, see the article "A Tour through the Visualization Zoo"[5] with description and demonstration of many inspiring techniques.

2.2 Visualization challenges

Based on the books[Aigner et al., 2011, Chapter 8], [Keim et al., 2010] and the paper [Childs et al., 2013] several research challenges are mentioned and an abstract consists of:

- Automatic suggestion of appropriate visual representation.[Aigner et al., 2011, Chapter 8]
- Representing the quality of data.[Aigner et al., 2011, Chapter 8] and dealing with incomplete, inconsistent, or erroneous data[Childs et al., 2013]
- Scalability: wast amounts of data, being able to move between long term and very short term.[Aigner et al., 2011, Chapter 8] and effective data visualizations with billions of items and or hundreds of dimensions[Childs et al., 2013]
- Interaction methods: ranging from small scale devices up to multiuser cooperative display environments.[Aigner et al., 2011, Chapter 8]. Distributed and collaborative visual analytics[Childs et al., 2013]
- Advanced analytical methods: deal with patterns at different scales and how to parameterize the analytical methods.[Aigner et al., 2011, Chapter 8]
- Visualize time series data in order to spot patterns, including repetitions.[Keim et al., 2010]
- Meaningful visualization of connections and graphs.[Keim et al., 2010]
- On-line visualization: New updates all the time. Also applies to analytical methods.[Keim et al., 2010]
- Generic tools instead of writing a new tool for every new task.[Keim et al., 2010]

Many interesting challenges, but to narrow down the scope only the one related to *scalability* and *analytical methods* will be looked at in more detail. The reason is that they are also very relevant for digital forensics where investigators are drowning in "big data". It's a bit fuzzy term, but after reading the Wikipedia article on the topic, one can summarize important properties of it by saying it's

> "when encountering limitations in terms of storage, transportation and processing of data ... it kicks in at varying thresholds depending on the capabilities of the organization" [Wikipedia and [Smith, 2013]]

The principle solution to big data has been known for a long time, including distribution and parallelism. The reason for the hype might be that a lot of organization are hitting this wall at the moment. A well known implementation is Apache Hadoop, inspired by ideas from Google. Jacop Homan in his talk "Petabytes and terawatts" [Homan, 2011] explains what Hadoop is. It basically consists of distributed *storage* and *computation*.

[4] Twitter page of moritz_stefaner `https://twitter.com/moritz_stefaner`

[5] A Tour through the Visualization Zoo: `http://queue.acm.org/detail.cfm?id=1805128` (ACM queue)

Storage, using Hadoop Distributed File System (HDFS) takes care of splitting huge data into smaller blocks, and replicate them on connected Hadoop nodes. There is a name node keeping track of location and integrity of the blocks for dealing with failures. The file system is written in Java and has currently a limitation on 4000 nodes, but is being expanded. Nodes are co-located in the same data center for performance.

Map-reduce is the *computational* module performing work, and the idea is that the calculation should occur where the data is stored. The duplicate copies can be utilized for speed up. Map reduce is a programing language and technique using key-value pairs, and data is written back to HDFS after calculation. A job tracker has a similar job as the name node in keeping track of the distributed calculations. Sorting and aggregation is the primary functions and the difficult part of writing map-reduce is thinking in key-value pairs. As a consequence languages like Pig, Hive and Crunch are developed on top of it.

Many other libraries are in development including

- Hbase: Real time access to Hadoop data
- Accumulo: Hbase + security / permissions
- Avro: Data serialization including schema with the stored data
- OOzie: Coordination of conditional tasks
- Flume and Sqoop: For collecting existing data (unstructured and structured)
- Whirr: Automate deployment on rented clusters like Amazon and Rack space
- Mahout: Machine learning including SVM and clustering
- Giraph: Calculation on graphs, like social graphs. His example of usage is the LinkedIn "InMaps" [6]

A related technique for storing huge amounts of data is to store it as key-value pairs, create a hash of the key (or value) used as a key and distribute it among storage units. The hashes are likely to be spread out evenly, and some redundancy in terms of overlapping hash spans can be implemented as used in distributed hash databases[7].

Another example of calculation on huge amount of data is a project by Google presented at Google I/O 2013: "All the ships in the world"[an Kurt Schwehr and Marks, 2013]. Google uses their proprietary technologies "Buckets", "App engine" and "BigQuery" for storing ship positions from the AIS[8] navigation system and displaying it in Google maps.

Another important trend discussed in the paper "Research Challenges for Visualization Software" [Childs et al., 2013] is the fact that CPU's of computers tend not to get any faster in terms of speed per core, but instead increase the amounts of cores. In addition specialized Graphic processing units (GPU) take this even further by implementing huge amounts of very simple cores capable of massive simple computations in parallel. This shift requires visualization software to be written with parallel computation in mind, and although distributed tasks like in Hadoop is one way to go, pushing workload to a GPU can also increase performance an order or more of magnitude. Examples of this includes digital currency mining like bitcoin and to some degree litecoin. Anther example is how hackers utilize GPU's for password cracking, which is actually the same kind of problem of performing hashes.

One of the questions introduced in the introduction was "How can machine learning and pattern recognition help in visualization?", and a small chapter of [Aigner et al., 2011, Chapter 6] mentions some methods of analytical support:

- Classification: Determine identity of an item
- Clustering: Group similar items
- Search and retrial: From a known pattern, find more of the pattern (exact or fuzzy / similar)
- Pattern discovery: Find frequent patterns
- Prediction: What will happen in the future?

Machine learning is a collection of mostly statistical algorithms for having computers learn from examples rather than being specifically programed for the task. Challenges includes what features or properties to measure and learn from, what algorithms to use and combine, how to configure them and how to interpret the final output. The algorithm can be designed to solve a particular

[6] For InMaps see `http://inmaps.linkedinlabs.com/`

[7] Security Now #398 on Distributed hash databases at `https://www.grc.com/sn/sn-398.htm`

[8] Automatic Identification System: Self reporting of identity, location, course and speed by ships using radio communication

problem very efficiently, but it does not understand the context outside it's function. These are all very important reasons why the human is a very important and why it's important to have effective interfaces between human and machine.

Machine learning like classification and pattern recognition can be used to highlight interesting combinations of attributes, and help in reducing the amount of points to be visualized. When searching for interesting papers, some swarm based intelligence methods showed up, using flocking algorithms (stay in center, avoid collision and predators) to visually show trends in stock markets[Moere, 2004] and using web usage logs[Saka and Nasraoui, 2010]. Although exotic and algorithm focused, they still have the problem of interpretation and parameter adjustments.

Danny Holten[9] has published[Holten, 2006] an interesting way to visually cluster interaction in graph trees. After showing some traditional ways to graph trees such as rooted, radial, balloon and treemaps, he shows the difficulty of understanding what's going on when using straight lines. His technique is to curve the connection lines depending on the hierarchy of the graph, controlled by a bundling strength factor[10]. Transparency is then used to be able to see several lines at the same time, and color is used for showing source and destination. Figure 4 shows the curving principle, figure 5 shows the effect of the bundling factor and figure 6 shows the "details on demand" view.

2.3 Existing tools and examples

In this section we will look at visualization techniques in different digital forensics domains.

2.3.1 File systems

File systems contain information needed to locate files on a storage medium. A well known way to visualize this information is to follow the path tree, showing folders and files as icons as most graphical operating systems allow you to do. Collected filesystem evidence can be mounted in read only mode and browsed this way, but one must be aware of hidden files, file permissions, alternate streams, and deleted files. Commercial solutions such as Encase and FTK eases the access to this information by not hiding such information from the examiner. Open source solution such as Autopsy for SleuthKit now supports graphical timeline in version 3[11], and similar functionality is listed for FTK [12].

Another interesting and quite old tool is the "Perspective wall"[Mackinlay et al., 1991] utilizing 3D and perspective to project files in time. Files are grouped by extension. Figure 7 shows a screen-shot of the implementation of this technique.

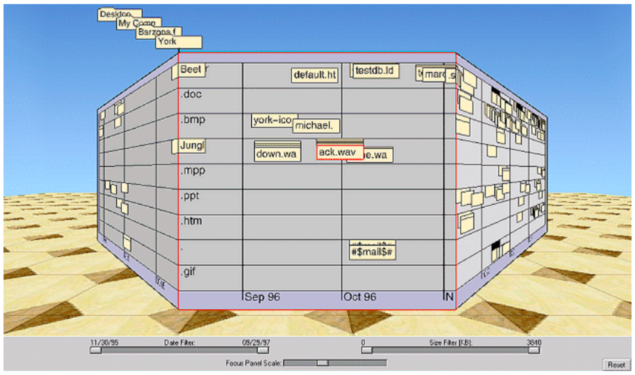


Figure 7: *Perspective wall [Mackinlay et al., 1991]*

Going even deeper one can extract content of files including log files, local e-mail storage, and browser history. One such open source tool is log2timeline[13] that can generate a flat file from the sources it supports. The main problem when relying on such automated tools is the limited protocols it supports, and the continuous change of formats. On the web page of log2timeline a widget called SIMILE is being used to graph[14] this data.

For logs there is a paper on a tool LogView "Visualizing Event Log Clusters"[Makanju et al., 2008]. The idea is to take the output of a clustering algorithm and visualize it using treemaps. Similar approaches exists for file systems that map size of files in order to figure out what occupies space on a disk such as Disk Inventory X[15] (figure 8 and WinDirStat[16].

[9] http://www.win.tue.nl/~dholten/#research

[10] See http://mbostock.github.io/d3/talk/20111116/bundle.html for simple d3 JavaScript implementation

[11] Autopsy at http://www.sleuthkit.org/autopsy/timeline.php

[12] Under "Features" and "Data visualization.." : http://www.accessdata.com/products/digital-forensics/ftk

[13] log2timeline by Kristinn Gudjonsson: http://log2timeline.net/

[14] SIMILE visualization http://log2timeline.net/browser.html

[15] Disk Inventory X for MAC http://www.derlien.com/

[16] WinDirStat for Windows http://windirstat.info/

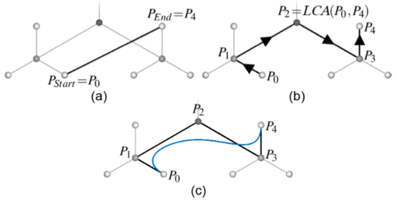


Figure 4: *How to bundle based on structure [Holten, 2006]*

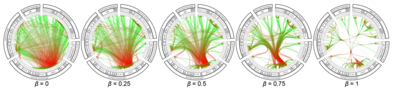


Figure 5: *The effect of bundle strength [Holten, 2006]*

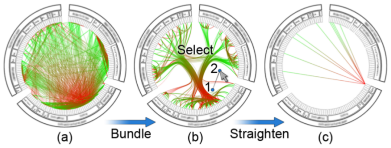
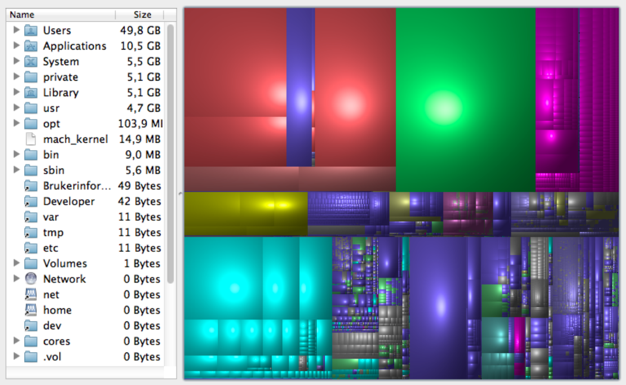


Figure 6: *Cluttered to bundled and "zoom in" function [Holten, 2006]*



Name	Size
Users	49,8 GB
Applications	10,5 GB
System	5,5 GB
private	5,1 GB
Library	5,1 GB
usr	4,7 GB
opt	103,9 MI
mach_kernel	14,9 MB
bin	9,0 MB
sbin	5,6 MB
Brukerinfor...	49 Bytes
Developer	42 Bytes
var	11 Bytes
tmp	11 Bytes
etc	11 Bytes
Volumes	1 Bytes
Network	0 Bytes
net	0 Bytes
home	0 Bytes
dev	0 Bytes
cores	0 Bytes
.vol	0 Bytes

Figure 8: *Treemap of disk content from Disk Inventory X*

2.3.2 Malware

Some files contain code with malicious intentions, and being able to detect them and what they do is an important part of forensics. Reverse malware engineering can be divided in static and dynamic analysis and dynamic analysis can contain debugging such as stepping through and modifying instruction on the fly. When doing dynamic analysis of code one can record requests to the operating system for logging file modifications, registry, network access and access to other shared libraries.

The site malwr.com hosts a database and front-end to the cuckoo sandbox where one can upload malware samples and have them analyzed both statically using anti-virus and monitoring API calls during execution. It has some graphs for showing API calls over time shown in figure 9. The main issues are the amount of calls and the time scale, and also that training a human to understand the patterns is hard. What does a normal program look like and the same for a malicious one? This is where machine learning using clustering and classification can contribute.

Using radial trees and bundling as shown in figure 6 is another example of visualizing software behavior.

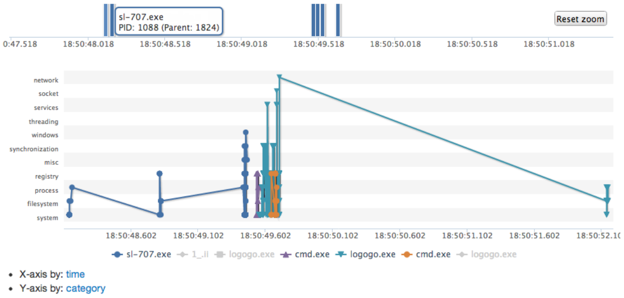


Figure 9: *Sample from Malwr.com:* `https://malwr.com/analysis/NGViNDEwYjRhMGY0NDlkMmFmNGJhMmUxZDAwMDIwMDM/`

2.3.3 Computer networks and intrusion detection

When we take one step back and look at communication between nodes, even meta data such as addresses, ports, packet size, and timing can be used to detect abnormal behaviors. Sudden increase in activity on an unknown port, nodes that usually don't communicate, timing patterns between packages and so fourth.

In this paper[Wright et al., 2006], the authors present a method for visualizing how different network protocols such as HTTP, SMTP, SSH and also chat and peer to peer programs use can be separated based on packet size, direction and timing (figure 10. They present several methods of plotting using arrival time alone and also combining the inter packet timing and size. Even though communication is encrypted, patterns will be visible unless packets are buffered and sent with a constant interval and size. Application could be to visualize to spot anomalies based on what normal behavior looks like, look deeper into the content for manual classification and then use

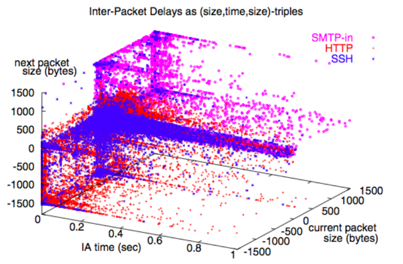


Figure 10: *3D plot of packet timing and size for pairs of packages [Wright et al., 2006]*

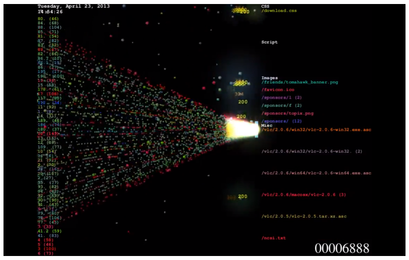

Tuesday, April 23, 2013
14:54:26
CSS
/download.css
Script
Images
/friends/tomahawk_banner.png
/favicon.ico
/sponsors/l (2)
/sponsors/f (2)
/sponsors/topix.png
/sponsors/ (12)
Misc
/vlc/2.0.6/win32/vlc-2.0.6-win32.exe.asc
/vlc/2.0.6/win32/vlc-2.0.6-win32. (2)
/vlc/2.0.6/win64/vlc-2.0.6-win64.exe.asc
/vlc/2.0.6/macosx/vlc-2.0.6 (3)
/vlc/2.0.5/vlc-2.0.5.tar.xz.asc
/ncsi.txt
200
00006888

Figure 11: *Web server access logs of distributed denial of service attack* `https://www.youtube.com/watch?v=hNjdBSoIa8k`

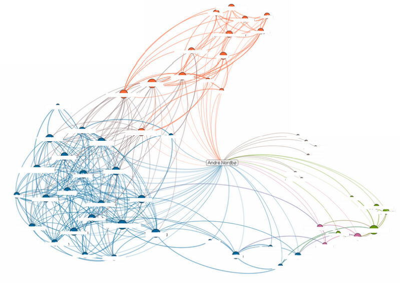


Figure 12: *My network: Bachelor, previous work and current master connections are easily to spot based on the clustering* `http://inmaps.linkedinlabs.com/network`

machine learning to create rules for classifying the behavior for future.

A program called Logstalgia[17] visualize web server logs by playing ping-pong with the incoming requests in real time. A demo video of a DDoS attack[18] as seen in figure 11 shows what it looks like.

Fabian Fisher[19] with others have created tools such as NFlowVis for visualizing NetFlow[20] data and Banksafe focusing on visualizing both security data, explained as intrusion detection and firewall logs, and also monitoring data such as health and status (cpu, disk, uptime). The later uses cloud based solutions (Google BigQuery) for handling huge amounts of data, and the front end is web based.

RSA Silver Tail[21] is a commercial product, focusing on detecting anomalies in web applications. From their presentation videos they mention several use cases such as detecting denial of service attacks where the criminals try to find the most resource intensive parts of your web applications and target it to slow it down, business logic abuse such as creative usage of coupons and other ways to get items for too low prices, and lastly detecting man-in-the-middle attacks such as multiple logins from different locations and session hijacking.

Splunk[22] is also a well known commercial solution for indexing, searching and also visualizing log data in many formats.

2.3.4 Social media

Finding patterns in social media is a hot topic. The paper "Visual Analysis of Social Media Data" [Schreck and Keim, 2013] defines the term the following way:

> "all media formats by which groups of users interact to produce, share, and augment information in a distributed, networked, and parallel process"[Schreck and Keim, 2013]

Tons of services fall under this definition, although the most famous ones are sites like Facebook, LinkedIn, Twitter and Youtube. Other sources like blogs, message boards, e-mail lists, dating sites, IRC channels and photo sharing all contain information and behavior that when aggregated can tell a lot about an individual or group. In a way, using social media in investigation is simply an extension of wiretapping performed for decades, but with the advantage of huge amounts of data being collected centrally. We also see this tendency of wanting to store session data for phone and Internet usage in the the European data retention directives. One interesting example mentioned in Security Now episode 408 is how

[17] Logstalgia at `https://code.google.com/p/logstalgia/`

[18] Logstalgia DDoS attack: `https://www.youtube.com/watch?v=hNjdBSoIa8k`

[19] Fabian Fisher at `http://ff.cx/`

[20] Cisco protocol for network metadata

[21] RSA Silver Tail at `http://www.emc.com/about/news/press/2013/20130605-01.htm`

[22] Splunk `http://www.splunk.com/`

a group of terrorist could buy these "throw away" cell phones and only use them for communication between each other in the hope to stay anonymous, and how suspicious that pattern would look like compared to normal phone usage.

As with file systems there is meta data and content in social media. At the meta data level there are examples of aggregating who communicates with whom, for how long and the frequency in order to find groups and roles within groups. Who are the leaders, connectors, askers of questions, who answer etc as explored in[Hansen et al., 2010] using message board data,

The LinkedIn's InMaps as mentioned earlier, visualizes you own connections and makes it easy to spot the networks you have been a part of based on the connections between your connections, like in figure 12. Similar visualization of facebook connections can be generated by Wolframalpha[23]

2.3.5 Presenting evidence

When presenting evidence it's all about making it understandable and believable to non-experts, and although visualization results of previous stages can be used, they might require too much explanation and training in order to make sense. Keeping it simple focusing on what happen using a simple time line and then go into details on demand. Requirements for such a tool would be to deal with uncertainties in time, different sources of information and linking of events like illustrated in figure 13. The forensics wiki[24] has a list of developer tools and user ready tools commonly used in the field. Other interesting ideas on visualizing evidence can be found on the LegalInformatics[25] blog.

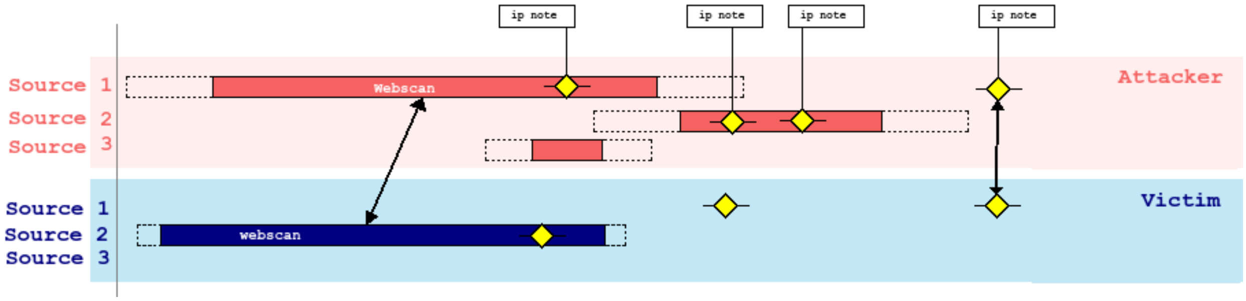


Figure 13: *Ideas for a timeline tool*

3. Conclusion

The research question for this project is usage of visualization techniques in digital forensics for identification, analysis and presentation, and answering it by finding answers to what makes visualization effective, common challenges and examples of tools and methods. It is important to know the data, what the goal of visualization is and how the human mind works to be most efficient in visualization. Common challenges discussed are how to deal with huge amounts of data by utilizing parallel computing and some ideas around using analytical methods to find the needles in the haystacks of data with algorithms like clustering and classification. There are many tools for visualizing data such as file system, malware, network and social network. Although the ideas behind the tools can be very interesting, many of them are highly dependent on the format of the input data and the platform they run on. The major issue seems to be how to be able to choose a fitting method in the jungle of methods and use them across a variety of input data sources, at at the same time not locking it to a specific device or platform. Building based on web technology seems to be where we are heading.

4. Acknowledgments

This work was supervised by Katrin Franke and Jayson Mackie, both pointing to interesting books and articles helping to focus this project.

[23] Wolframalpha: http://flowingdata.com/2012/09/03/analyze-your-facebook-profile-with-wolframalpha/

[24] http://www.forensicswiki.org/wiki/Tools:Visualization

[25] http://legalinformatics.wordpress.com/2013/06/09/12-june-starger-presents-legal-visualization-software-penn-state/

REFERENCES

[Aigner et al., 2007] Aigner, W., Miksch, S., Müller, W., Schumann, H., and Tominski, C. (2007). Visualizing time-oriented data—a systematic view. *Computers & Graphics*, 31(3):401 – 409.

[Aigner et al., 2011] Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011). *Visualization of Time-Oriented Data*. Springer.

[an Kurt Schwehr and Marks, 2013] an Kurt Schwehr, F. C. F. and Marks, M. (2013). Apache hadoop - petabytes and terawatts (google i/o, pub 17th may 2013). `https://www.youtube.com/watch?v=MT7cd4M9vzs`. Visited May 2013.

[Childs et al., 2013] Childs, H., Geveci, B., Schroeder, W., Meredith, J., Moreland, K., Sewell, C., Kuhlen, T., and Bethel, E. (2013). Research challenges for visualization software. *Computer*, 46(5):34–42.

[Hansen et al., 2010] Hansen, D. L., Shneiderman, B., and Smith, M. (2010). Visualizing threaded conversation networks: mining message boards and email lists for actionable insights. In *Proceedings of the 6th international conference on Active media technology*, AMT'10, pages 47–62, Berlin, Heidelberg. Springer-Verlag.

[Holten, 2006] Holten, D. (2006). Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748.

[Homan, 2011] Homan, J. (2011). Apache hadoop - petabytes and terawatts (linkedintechtalks, pub 30th now 2011). `https://www.youtube.com/watch?v=SS27F-hYWfU`. Visited May 2013.

[Iliinsky, 2012] Iliinsky, N. (2012). Designing data visualizations (linkedintechtalks, pub 5th apr 2012). `https://www.youtube.com/watch?v=R-oiKt7bUU8`. Visited May 2013.

[Keim et al., 2010] Keim, D. A., Kohlhammer, J., Ellis, G., and Mansmann, F. (2010). *Mastering The Information Age-Solving Problems with Visual Analytics*. Florian Mansmann.

[Mackinlay et al., 1991] Mackinlay, J. D., Robertson, G. G., and Card, S. K. (1991). The perspective wall: detail and context smoothly integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 173–176, New York, NY, USA. ACM.

[Makanju et al., 2008] Makanju, A., Brooks, S., Zincir-Heywood, A. N., and Milios, E. E. (2008). Logview: Visualizing event log clusters. In *Proceedings of the 2008 Sixth Annual Conference on Privacy, Security and Trust*, PST '08, pages 99–108, Washington, DC, USA. IEEE Computer Society.

[McCandless, 2012] McCandless, D. (2012). The beauty of data visualization (tededucation, pub 23rd nov 2012). `https://www.youtube.com/watch?v=5Zg-C8AAIGg`. Visited May 2013.

[Moere, 2004] Moere, A. V. (2004). Time-varying data visualization using information flocking boids. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, pages 97–104, Washington, DC, USA. IEEE Computer Society.

[Saka and Nasraoui, 2010] Saka, E. and Nasraoui, O. (2010). On dynamic data clustering and visualization using swarm intelligence. In *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*, pages 337–340.

[Sankar, 2012] Sankar, S. (2012). The rise of human-computer cooperation (ted, pub sept 2012). `http://www.ted.com/talks/shyam_sankar_the_rise_of_human_computer_cooperation.html`. Visited May 2013.

[Schreck and Keim, 2013] Schreck, T. and Keim, D. (2013). Visual analysis of social media data. *Computer*, 46(5):68–75.

[Shneiderman, 1996] Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343.

[Smith, 2013] Smith, T. (2013). Exploration on the big data frontier (tededucation). `https://www.youtube.com/watch?v=j-0cUmUyb-Y`. Visited June 2013.

[Wright et al., 2006] Wright, C. V., Monrose, F., and Masson, G. M. (2006). Using visual motifs to classify encrypted traffic. In *Proceedings of the 3rd international workshop on Visualization for computer security*, VizSEC '06, pages 41–50, New York, NY, USA. ACM.