Security visualisation

This thesis provides a guideline of how to generate a visual representation of a given dataset and use visualisation in the evaluation of known security vulnerabilities

by Marco Krebs and William Rothwell

ISTOCKPHTO/THINKSTOCK

Royal Holloway series | Information Security | Vulnerability Management | Information Visualisation

Visualisation in vulnerability management

Information visualisation is a great tool to support the vulnerability management process. If done properly, data can be visualised in a way that the critical issues stand out immediately and can be prioritised accordingly

by Marco Krebs and William Rothwell

"A picture is worth a thousand log records." Here is a guideline of how to generate a visual representation of a given dataset and use visualisation in the evaluation of known security vulnerabilities. Although this example is based on the output of an automated vulnerability scanner (Nessus), the suggested information visualisation process can be applied to generate any kind of visualisation.

Have you ever been stuck looking at a list of security vulnerabilities that seems endless? – You are not alone... Even worse: All issues appear to be of high priority and of equal importance at first sight. However, given the limited amount of time and resources in practice, it is key that each of them is carefully evaluated and prioritised to take adequate steps towards mitigation. If not done properly, the lack of prioritisation many times leads to the fact that known vulnerabilities do not get fixed within an appropriate time frame – if they get fixed at all.

This is a perfect example where information visualisation can help with the process of prioritisation. What if you had a visual representation of the findings that not only shows the underlying network architecture but makes the most critical issues sticking out immediately? What if you could see from this visual the attack path and identify potential security enforcement points along that path to cut-off the attacker?

If done properly, information visualisation takes advantages of human perception. As human beings we are literally wired to see: The human visual system is often described as a flexible pattern finder that can quickly detect changes in size, color, shape, movement or texture.

Basic graph design principles

Therefore, by following certain (simple) rules, a given visualisation can be more expressive and effective than another. Some of these principles to graphical excellence are as follows:

- Reduce non-data ink (show nothing else but the data)
- Show no more than five distinct attributes (rule of thumb)
- Follow Gestalt principles (use the human pattern detection engine)
- Emphasise exceptions (making them jump out immediately)
- Show comparisons (identification of differences)
- Annotate data (use labels and legends to guide the reader)
- Show causality (always explain the root cause of outliers)

So, how does this work? Let's look at a practical example where we take the output from the Nessus vulnerability scanner and turn that (raw) data into something meaningful (also known as information). Nessus is a widely used vulnerability-scanning tool. Its goal is to detect potential vulnerabilities on the tested system(s). The software is free of charge for personal use.

> "The goal of information visualisation is to transform abstract data into computer graphics that are easy to understand and to use to support decision making and reasoning"
> Greg Conti

This leads us to the information visualisation process.

> "Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space"
> Edward Tufte

The information visualisation process

The generation of any visual representation is an iterative process. Starting from a given dataset and feeding it to a visualisation engine the outcome is a graphical representation, which communicates a message to the reader.

1) Problem definition and message: First, the problem or objective must be clear to start with. What is it that you want to visualise? Implementation details do not matter yet. Is the purpose of the visual representation to communicate something that is already evident but hidden in the underlying data, or is the goal to identify trends and relationships? This is defined best in a goal statement, for example:

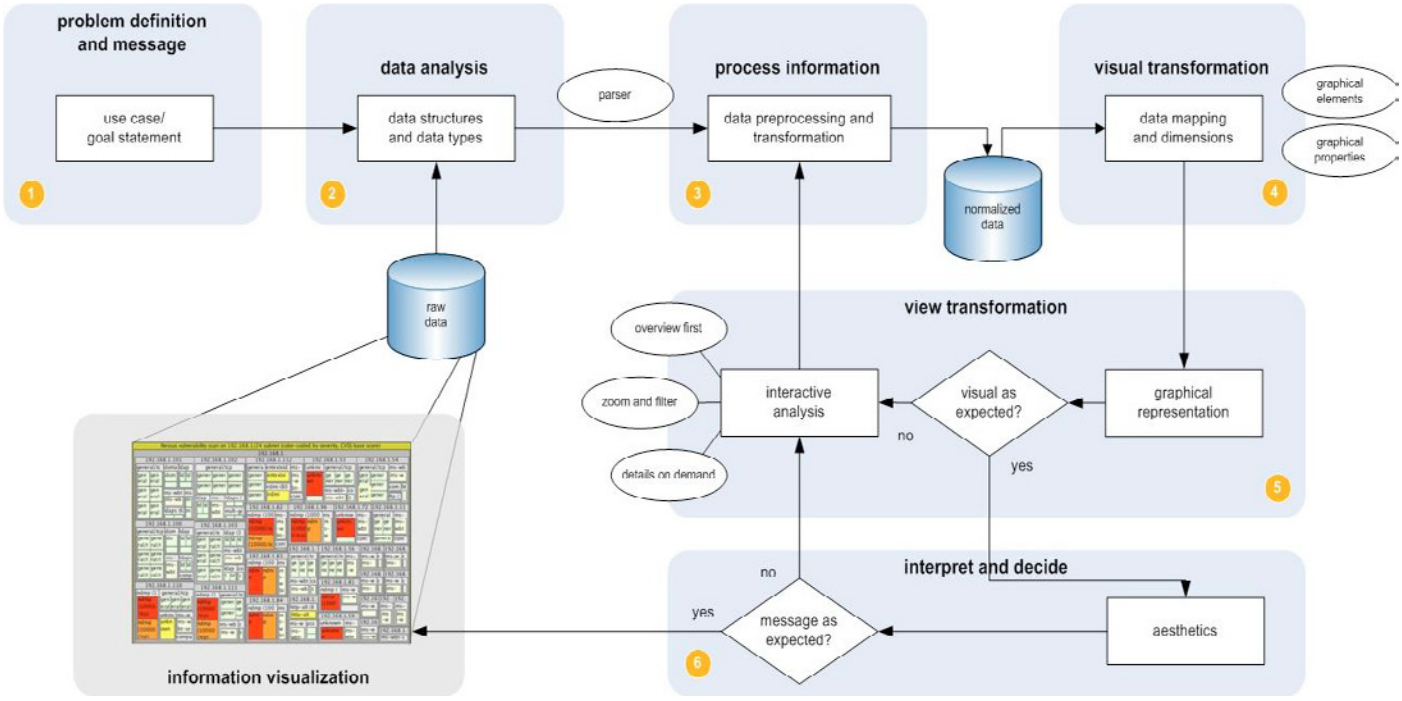
"Starting from the results of a Nessus vulnerability scan, my goal is to generate a visual representation that goes beyond the form of the usual list/table of weaknesses. The graph should reflect the underlying network infrastructure and allow the quick identification of critical issues. It should support the reader in the (risk-based) prioritisation of further actions."

2) Data analysis: Second, we need to find out what type of data is required and what the data structures look like. Based on the goal statement above we need data on network topology, the vulnerabilities identified per system but also some rating/scoring system to determine the severity level.

Reports from the Nessus scanner can be saved in various formats. One example is the NBE format, which is a pipe-delimited text file that can be easily imported into external programs. Besides general information about the scan it contains the scan results as well as routing information from the scan host to the destination machine(s). This means that the network architecture can be drawn based on these routes recorded.

The level of detail of the findings listed in the NBE file varies from an "open port found" message to a more detailed explanation containing a risk indicator

Figure 1: The information visualisation process



such as the CVSS base score. CVSS stands for Common Vulnerability Scoring System. Over the years it has emerged into a de-facto standard on how to evaluate security vulnerabilities. The base score is build from the evaluation of different metrics and ranges from 0 (lowest, least severe) to 10 (highest, most severe). This score is a good indicator of prioritisation: Obviously, the issue(s) with the highest score should be addressed first.

3) Process information: The data provided in the NBE file need to be pre-processed (parsed) into logical structures that can be understood by the visualisation tool used to generate the graphical representation. One tool requires a simple comma-separated text file whereas another one requires input in a more proprietary format. For this example, I used GraphViz, a widely used open-source graph visualisation software. GraphViz properties are defined in the so-called DOT language. Hence a parser is required to extract the relevant data and to perform a translation and/or normalisation between the Nessus data and the attributes of the DOT language. Figure 2 below shows a graphical representation of the network architecture based on the routing information in the Nessus file. It is apparent that all systems can be reached from the scanning host (in red) via the device with IP address 10.1.1.3 (in rectangular shape). What is still missing is the information about the identified vulnerabilities. This will be added next.

4) Visual transformation: The fourth and next step is about mapping the scan information to graphical elements and assigning visual properties. In order to make critical issues visible ("pop out") immediately to the human eye, some scoring system needs to be applied that allows building a relationship between key visual attributes and the number of vulnerabilities and their severity.

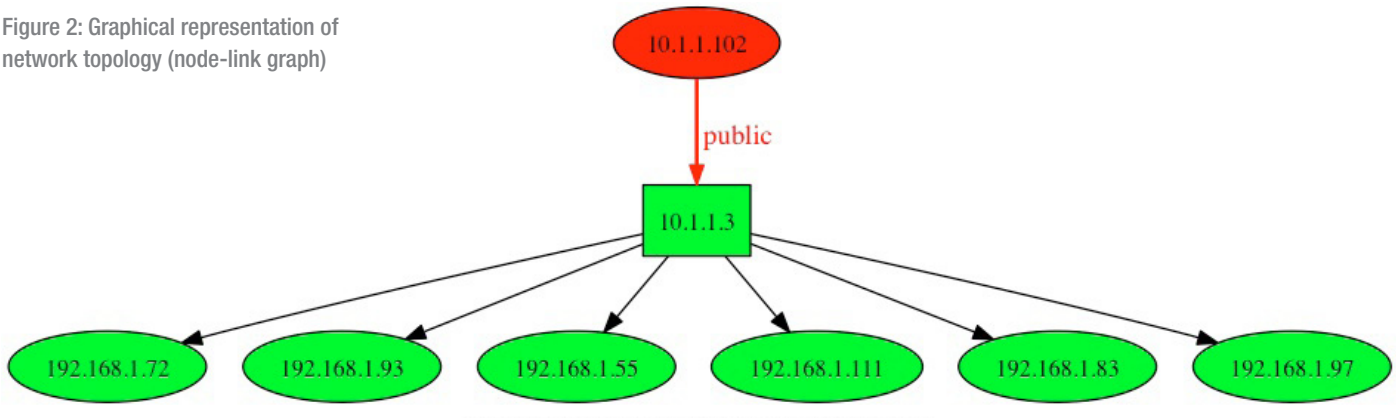
Color can be used to mark a node according to the rating of the most severe vulnerability found on that host. For this example, I used a scale that can be determined from the CVSS base score as follows:

Another attribute can be the size of a node, which varies depending on the total number of vulnerabilities found on that host. The bigger a node, the greater the number of vulnerabilities assigned to that host.

5) View transformation: Now, at the fifth stage of the information visualisation process, we are ready to generate the graphical representation. This is often an iterative process by itself because this is a question of design, too. Does the graph look as expected? Are all objects visible? Is a modification of graph elements/attributes required? If yes, we need to go back to the previous stage, select a different set of attributes and run the data visualisation engine again… I consider this the hardest part of the process.

> "The trick lies in creating the visualisation system that best fits your needs"
> Greg Conti

Figure 2: Graphical representation of network topology (node-link graph)



Node-link graph example using GraphViz

TABLE 1: COLOUR-CODING SCHEME

Colour	Rating	CVSS base score
Red	Critical	Greater than 8
Orange	Important	Between 6 and 8
Yellow	Moderate	between 4 and 6
Light blue	Low	between 2 and 4
Grey	Info	less than 2

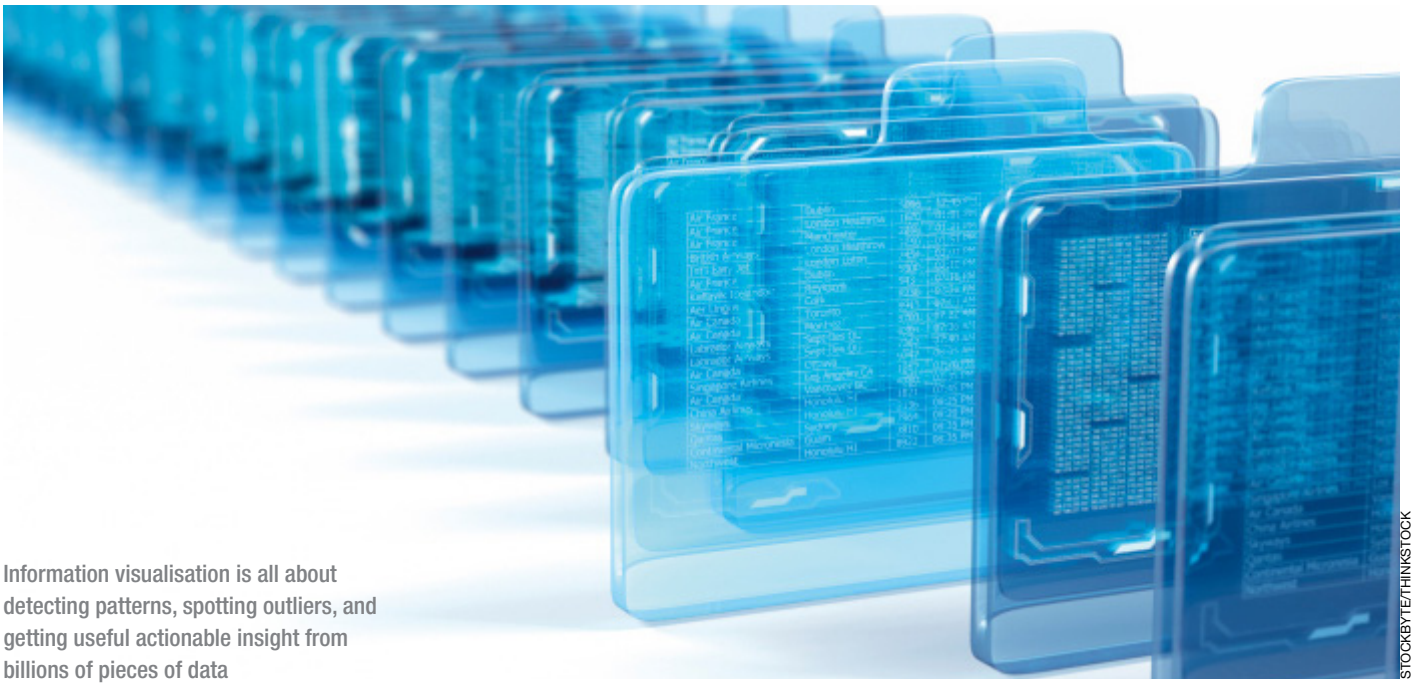
Another option to think of is data aggregation so that less important information can be shown in condensed form. One of the key principles of how to bring data into visualisations is known as the information seeking mantra by Ben Shneiderman.

Overview first: Gain an overview of the entire data collection first. The first visualisation often serves as a general indicator and can be used to further tune the graph attributes in the right direction.

Zoom and filter: Once an area of interest has been identified, another graph can be created that zooms in on them. Items that are not of interest can be filtered out (aggregation). This requires changing the graph attributes once more.

Details on demand: Depending on the necessary level of detail required, this process is repeated to show a few selected items only.

6) Interpret and decide: When satisfied with the outcome, the aesthetics of the visual design and the question on the result can be considered. Does the graph communicate the message as expected (refer to the goal statement set in step one)? And, of equal importance, is the reader able to understand it? If you can answer both questions with a yes, then congratulations! The journey has come to an end – at least for now. As data tell a story, you may likely discover new elements that lead to further analysis.



Information visualisation is all about detecting patterns, spotting outliers, and getting useful actionable insight from billions of pieces of data

STOCKBYTE/THINKSTOCK

From this... (Nessus output file)

...to this

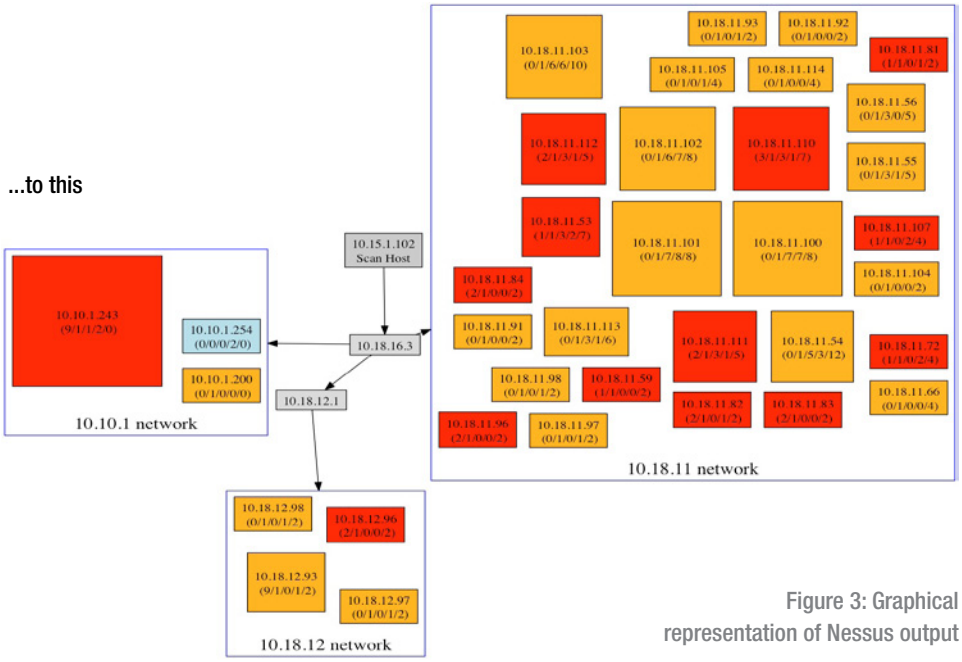


Figure 3: Graphical representation of Nessus output

The outcome of this process is as follows:

Figure 3 was generated following the six-step visualisation process described above. Nodes marked in red contain the most critical vulnerabilities (refer to table 1 above). The size of each node further gives an indication of the total number of issues found on that host. The numbers in brackets represent the amount of vulnerabilities per severity level (critical/important/moderate/low/info). Network topology information is provided as well. You can easily see that most promising system to attack is 10.10.1.243 (on the left) for its high number of (critical) vulnerabilities.

At its core, information visualisation is all about detecting patterns, spotting outliers, and getting useful actionable insight from billions of pieces of data. By emerging into the field of visual analytics, people will be able to interactively interact, discover, and manipulate data to reveal and communicate the story that is important to them. I hope that you will become part of that community, too. Seeing is believing – let the data tell the story. ■

About the authors

Marco Krebs' interest in information security was caught when given the opportunity to engineer a Linux-based Internet gateway in 2001. Since then, Marco has been working in different areas of information security, including security/network/systems engineering, consulting, and security testing. His current position includes building-up and running the computer emergency response team for a Swiss financial institute. Subjects of interest are information visualisation, R&D, security monitoring, and forensics.

William Rothwell is an experienced security practitioner with over 20 years in the information security field for government, military, finance and commercial organisations. He specializes in security architecture and design, crypto application and system intrusion defense research. In 2004 he founded Abatis Limited, which provides a unique, patent protected, non-signature based solution to protect computers from malware infection and hacking intrusions.

"Overview first, zoom and filter, then details on demand"

The information-seeking mantra by Ben Shneiderman