Chapter 11: Building Interactive Security Visualizations

Many in the design community understand that design must convey the essence of a device’s operation; the way it works; the possible actions that can be taken; and, through feedback, just what it is doing at any particular moment. Design is really an act of communication, which means having a deep understanding of the person with whom the designer is communicating.

Donald A. Norman, The Design Of Everyday Things

The main purpose behind any of your data visualization efforts should be to help consumers understand and learn from data. In other words, *effective communication* is the primary goal of your visual creations.

As we’ve seen in previous chapters, developing simple and successful *fixed* tables and charts requires knowledge, skill and practice but can provide substantive illumination of a topic, issue or problem if executed correctly. In most cases—probably 95% of the time—these fixed views are all that is needed to achieve the goal of communication. There are situations, however, when static views of data are either insufficient or just not practical, requiring the move to a more dynamic medium to help consumers explore the messages the data has to offer. This chapter will help you understand when the move to interactive visualizations makes sense and introduce you to some of the resources and techniques that will help you craft effective messages, dashboards and exploration tools.

type="note"

The skills, art and science surrounding interactive visualizations span a multitude of disciplines across many decades. As a result, this single chapter serves more as a survey and reference for further study for the topic as a whole while providing practical guidance for where to apply interactivity within the scope of information security.

Moving From Static To Interactive

Assuming our “95%” premise holds true, your first instinct when planning out visualizations should be to “go static”. It will generally take much less time to construct fixed visualizations even with the tweaking and polishing necessary to produce a consumer-worthy graphic. You should also consider sticking with stationary images if the project you’re working on is fairly discrete with a data set having minimal number of dimensions (i.e. rows, variables/columns/fields). As Scott Murray put it in his book, Interactive Data Visualization for the Web, “*A fixed image is ideal when alternate views are neither needed nor desired, and* ***required*** *when publishing to a static medium, such as print”.*

If you’re still feeling the “interactive itch”, there are three primary goals to consider when contemplating a new visualization:

* **Augmentation**: If adding interactive capabilities to visualization will help speed up or automate tasks consumers would normally perform manually, going interactive is definitely the right thing to do.
* **Exploration**: If the number of dimensions and size/diversity of the data set grow sufficiently large, it may be better to give the consumer the ability to explore the relationships and outcomes on their own versus trying to guess which set of static graphics will be most useful.
* **Illumination**:If a topic is complex enough it may help to provide a well-executed, interactive visualization that provides a user-friendly interface for directed/constrained navigation around the data you’ve chosen to present.

Let’s delve a bit further into each of these areas with a focus on information security examples.

Interaction For Augmentation

There are many repetitive, time-consuming, data-driven tasks in information security. Logs must be collected and correlated, alerts must be received and attended to and anomalies must be investigated. These actions often involve running a variety of utilities over individual pieces of data or sets of data elements to determine if there truly is badness occurring on your network. Any tool that helps alleviate this tedium and speeds up reliable detection of malicious activity is a welcome addition to any security engineer’s toolbox.

Recognizing this, a research team led by Robert Erbacher worked to understand both the problem domain—situational awareness of malicious network activity—and how incident responders think and process information, which resulted in the creation of VisAlert ( http://digital.cs.usu.edu/~erbacher/publications/VisAlertCGA2006.pdf1), a visual correlation tool that facilitates situational awareness in complex network environments.

Figure 11.1 The VisAlert Visual Correlation Tool [793725c11f01.png]

It’s fairly straightforward, with tools such as Circos (http://circos.ca/), to build a radial diagram similar to the VisAlert model in Figure 11.1 and add some interactive features. However, it takes more than eye-candy appeal for any visualization—fixed or interactive—to be truly useful, and it was the detailed focus on the following areas that makes VisAlert notable.

Define The Problem

This is merely an extension of the “start with a question” manta you’ve seen in many of the preceding chapters. While there is merit in building visualizations in a vacuum to learn how to work with a new language or framework, it is imperative that you understand what problem you’re trying to solve with a consumer-oriented interactive visualization and who the users will be before you attempt to deliver a finished product. Even if you’re an established practitioner, your personal experiences may only give you insight into one aspect of a problem domain, and collaboration with others—especially those who you believe to be the natural consumers of your interactive visualization work—can make or break a project.

For the VisAlert team, this ultimately meant their goal was to aid analysts’ decision-making processes by providing a robust visual correlation mechanism. Rather than try to build a new intrusion detection system or deliver a “toy model” solution that works only with perfect and limited data sets, they chose to design a system that works at-scale with real world data volumes and types that security analysts already use in their daily workflows. So, while the problem scope is fairly narrowly defined, it has sufficient breadth and scope to be useful as well as visually appealing.

Seek Domain Expertise

The VisAlert team started with real-world information security analysts to understand their **mental models** of how they go about identifying badness. Mental models are our conceptual models of the way things work or our understanding of how to interact with the world or systems around us. Security analysts develop domain specific mental models through their training and practical work experiences. These models evolve with each successful (or failed) identification and eradication of malicious activity. With each investigation, analysts learn which processes provided the most value and these are automatically added to their existing mental framework. By working with these individuals throughout the design process, the team was able to identify what parts of the analysts’ workflows would benefit from enhanced visualizations (e.g. inclusion of salient parts of network diagrams and automatically highlighting specific protocols and paths) and automation (e.g. DNS lookups, targeted correlations).

Take An Interdisciplinary Approach

The team drew on the talents and works of experts in the fields of information architecture, cognitive psychology, application development and computer science—along with the domain experts—to build and refine the tool in a process they describe as a “modified hermeneutic circle”—the movement back and forth between the parts and the whole—which is shown in Figure 11.2.

Figure 11.2 The VisAlert Visual Correlation Tool Design Methodology [793725c11f02.png]

Their methodology has a strong resemblance to the agile development process (http://agilemanifesto.org/principles.html) where all those involved are equal partners, each working together to yield a successful finished product. If your organization has an application development team and you’re not familiar with agile, you would do well to invite one to lunch to understand how agile works in the real world. (Plus, you’ll have made a friend in the development community and can hopefully help them understand application security a bit better as well.)

Fundamentally, both concepts employ highly effective and efficient feedback loops to help ensure your project stays on the rails and arrives at the desired destination as quickly as possible. You may be the one building the finished product and you may be a savvy practitioner, but you should also regularly seek input and feedback from others in and outside your domain to ensure you’re constructing the right elements.

The VizAlert tool has been a feature in papers and security-oriented conferences since 2006 but has not manifest itself as a commercial or open source product as of this book’s publication.

Interaction For Exploration

Most networks contain their fair share of vulnerabilities and the Nessus (http://www.tenable.com/products/nessus) vulnerability scanner is one of the de-facto tools that can help you find them. If you’ve ever seen the output from a detailed Nessus report (Figure 11.3) you know that each host will have a listing of vulnerable components and each component will have many attributes, including basic and detailed descriptions, overall rating and CVSS score. A full report can be hundreds of pages long and makes for excellent nighttime reading if you’re having trouble sleeping.

Figure 11.3 Sample Nessus Detailed Vulnerability Report [793725c11f03.png]

Even a small network, such as the one created for the VAST 2011 visualization challenge (http://hcil.cs.umd.edu/localphp/hcil/vast11/) can have thousands of vulnerability findings (the VAST network data—included on the book’s web site—has over 2,000). While it’s possible to spin the data multiple ways and produce reams of static visualizations, this is definitely a perfect example of where an interactive tool could really help security analysts explore and prioritize how they will attack the problem of which vulnerabilities to remediate first.

Tenable does provide interactive reporting tools, but we’ll be focusing on an innovative open source tool released in 2013 by John Goodall called the Nessus Vulnerability Explorer (NV) [[http://ornl-sava.github.io/nv/#](http://ornl-sava.github.io/nv/)]. NV allows you to take an export from your Nessus scans, drag the file right into your browser and begin exploring the vulnerabilities contained within.

Figure 11.4 Nessus Vulnerability Explorer Interactive Treemap Interface [793725c11f04.png]

The interface is based on a treemap, which is a visualization that enables presentation of hierarchical data in a very compact way through nested rectangles, with the size and color of each rectangle being mapped to categorical or quantitative variables within the data set. Treemaps take a bit of getting used to, but once you learn how to decode them they can become valuable allies in targeted visualizations.

Goodall’s interactive treemap lets the consumer rearrange the structure of the hierarchy through a simple drag-and-drop action, so you can present a traditional IP address-centric view of the vulnerabilities or switch to a view based on Nessus vulnerability (plugin) ID or even by port. Through a single click, nodes can be sized by volume or potential impact and vulnerability details are revealed through single clicks on individual rectangles.

The view in Figure 11.4 has over 240 nodes, yet it’s very straightforward (and quick) to see all nodes with similar vulnerability profiles. All necessary information is kept on-screen and the bar charts at the bottom of the display provide a useful high-level overview to help guide exploration. A traditional summarized report view would no doubt require much scrolling and panning to provide the same type of information and it would be almost impossible to discern patterns in the environment.

Figure 11.5 Excel Pivot Table With Linked Charts [793725c11f05.png]

However, all exploratory interfaces do not need to be this elaborate. Figure 11.5 shows a simple Excel workbook of a firewall log extract that includes filtering controls at the top of the log entry data table and two pivot tables showing views by firewall and port (respectively) with matching bar charts that dynamically change as you manipulate the pivot table values. More modern versions of Excel do not have the workbook size limitations of previous offerings and can comfortably fit over a million rows and 16,000 columns provided you have a robust enough system to support such a large workbook. You might be surprised just how useful it can be to simply provide intelligently summarized tabular views of data sets—paired with basic visualizations—that can be easily sorted on demand by the consumer. It may sound simple, but remember: you still need to do the hard work of finding, cataloging, acquiring, cleaning, augmenting and processing the data (ah, the glamorous life of a security data scientist).

Interaction For Illumination

While everyone may seem to be carrying an i-device of some sort and constantly plugged-in to everything, the truth is that most individuals still have only a surface-level understanding of the digital world they live in. For instance, they know that their Instagram app requires an account with a username and password before they can post pictures for their friends to see, but the details of the binary world below that process—where hue, saturation and brightness are digitized, network packets are exchanged and information is transported and stored potentially thousands of miles away—remains as much a mystery as does most of the inner workings of a modern automobile engine.

Even in our workplaces, business processes may be well-understood but the complexity of the entirety of information technology components that make those processes possible can be somewhat overwhelming to IT specialists, let alone business professionals. Consider that a modest application has code that might be touched by over thirty developers, supported by over fifteen operations administrators, span three firewall zones and have components that reside on sixteen disparate systems. It’s incredible we have as much security as we do in such diverse and complex environments and a bit more understandable why all of those individuals involved in the process don’t fully grasp all the nuances of how to ensure that security is a primary emergent property of the system as a whole.

Understanding how complexity is masked, hidden or ignored should make it easier to see why topics we security-folk are passionate about, such as encryption, system/data integrity and data privacy are faint blips on the radars of most individuals. However, our cause and profession have merit and we *can* help raise awareness of these important topics. One good way to do this is through the use of interactive visualizations.

A great example of *how* to do this is the “World’s Biggest Data Breaches” visualization (http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/) created by David McCandless and Tom Evans of Information is Beautiful (http:// informationisbeautiful.net/).

Figure 11.6 World’s Biggest Data Breaches Interactive Visualization [793725c11f06.png]

Data breaches, as seen in Chapter 7, are a reality yet are not well understood outside of the security domain (perhaps not even fully *within* the security domain). When the technical and general news media reporting almost a breach-a-week it can be difficult for people to keep up, let alone digest the diversity of the attacks. David and Tom—who are visualization and development experts, not information security professionals—set out to build an easy-to-use tool that would help consumers gain a better understanding of the quantity, variety and magnitude of breaches that have made headlines over the past few years.

By following a paradigm of “*overview first, zoom and filter, then details-on-demand”* put forth by Ben Shneiderman back in 1996 in his “Visual Information Seeking Mantra”, they created an interactive bubble chart (Figure 11.6) which is organized vertically by year. Consumers can filter the display to show breaches by organization type or method of leak and can also change the factors that make up bubble size and color.

Publications such as the Verizon Data Breach Investigations Report (http://www.verizonenterprise.com/DBIR/2013/) and Trustwave’s Global Security Report (http://www2.trustwave.com/rs/trustwave/images/2013-Global-Security-Report.pdf) plus online databases such as DataLossDB (<http://datalossdb.org/>) and the Privacy Rights Clearinghouse (http://www.privacyrights.org/data-breach) have covered breaches for many years yet tend to be read and mined mostly by information security professionals. What has made this interactive tool more appealing and useful to a much broader audience than these established resources?

Making Interfaces Accessible

There’s nothing quite like a never-ending, scrolling table filled with security jargon and wrapped in cold, official language to make the average person head for the nearest cat picture. Even a well-crafted, comprehensive report can be daunting to pick up and look through when the topic is so far removed from the daily experience of even the most tech-savvy business executive.

The World’s Biggest Data Breaches visualization succeeds because it presents the data within a familiar and friendly setting—a web page—and makes excellent use of color, style and design to present a tool that has an intuitive look-and-feel with no fear of “breaking” anything. The “buttons” look and behave as expected. The filtering interface has plenty of whitespace and steers clear from too much jargon or too little context. Mouse movements and actions provide instant, game-like feedback; and, even without instruction, the interface is almost instantly usable.

Imagine if this had been released as a Microsoft Excel file (yes, you can make clickable bubble charts with Excel) with macro warnings popping up on open and the ribbon and column headers consuming prime real-estate space and with your operating system switching between Excel and your default browser whenever you clicked to see the news story behind the detail. The basic functionality would have been the same, but the user experience would have been radically different.

Your consumers live in the browser and that’s where most (if not all) your creations should be targeted for deployment. Latter sections in this chapter introduce some of the technologies that make these visualizations possible, but they will not include phrases like “Java applet” or “Adobe Flash”. Relying on the native capabilities of modern browsers and web frameworks will help you reach the largest possible audience in the most compatible and accessible ways possible. It will have the added benefit of making you sympathize a bit more with the complexities faced by user-interface developers (whom you should also take to lunch on occasion to trade security knowledge for useful front-end coding tips and techniques).

type="general"

The (Slow) Demise of Flash and Java

There was a time when Java and Flash applets were the only way to add “decent” visual interactivity on a web site. Java was (and is) a formal language taught in many schools, which had made it an especially easy choice for academic visualizations. Flash was (and is) easy to learn with friendly development tools that have made it highly popular among the general web development community.

Figure 11.7 The Decline Of Flash [793725c11f07.png]

While Flash still commands a presence on around 17% of web sites (Figure 11.7) the use of it as a visualization medium is in a slow, steady decline. In contrast, Java applets hold on to a razor thin 0.1% share of the web.

The fading of each technology can be attributed to many factors, including

* the never-ending vulnerability, breach and security update cycle along;
* the rise in popularity of platforms such as the iPad, iPhone and other touch environments that do not provide support for web site elements built with these tools, and
* the increased native platform capabilities due to widespread adoption of HTML5, CSS and Javascript across the most used browsers

To reach the broadest audience, it’s best to avoid proprietary technologies or visualization toolkits that require browser extensions.

Facilitating Directed Exploration

Donald Norman coined the phrase “the tyranny of the blank screen” in his book, The Design of Everyday Things. The perfect, illuminating, interactive visualization lies somewhere between this fully open, on-screen world and a fixed graphic. What design choices made the “Breaches” visualization easy to explore?

**Critical exploration elements and operations were prominent and visible**. Through consistent colors, shapes and prominent placement, the controls for the visualization are immediately discernable. By having the filter controls come up right after the visualization loads there is the immediate reaction of “*Oh, I can click this!*” on the part of the consumer. Color also draws attention to what the creators feel are especially compelling stories.

**All components and actions were consistent and deliberate.** Mouse movements highlight elements and mouse clicks select options and provide detail. There is no jumping between mouse and keyboard or switching between dragging and clicking. The interface becomes immediately predictable with no surprises, apart from interesting and engaging stories.

**Feedback is instant and all operations are safe**. While the site loads fairly quickly given all the data and resources it uses behind the scenes, there is a slight delay and this is where the helpful feedback starts. A familiar “loading” message appears but quickly fades directly to the core visualization. Every click produces instant feedback that is 100% undoable, either via the controls on the visualization or with a quick hit of the browser reload button. This feeling of safety puts consumers at ease and encourages them to explore.

**Actions are limited**. The interface provides options to change color and size of bubbles and highlight certain organization and breach types. You cannot group elements together and generate a bar chart or select individual organizations out from a list of thousands. These constraints make the interface much less daunting—a condition referred to as the *paradox of choice*2 since our intuition would argue that we want more freedom and more tools and ways to explore. Limiting actions also enables you to shape or guide the exploration in a particular direction. Considering how fixed graphics represent the extreme in limiting actions, you should be able to think back to what made the data interesting to you as you explored it and come up with a set of constrained, exploratory actions that lie somewhere between the freedom of an RStudio window and the constraints of a static graphic.

Including Appropriate Detail

Breaches are complex entities as illustrated by the breadth and depth of the VERIS taxonomy explained in Chapter 7. This level of technical detail would have been completely inappropriate for the mass-consumer audience of the “Breaches” visualization. Rather than bombard the consumer with multi-level taxonomy details, McCandless and Evans opt for simple summaries and succinct descriptions available on-click while making detailed news stories also available on demand.

The level of detail you choose to provide in this type of visualization is highly dependent on the target consumer. Including VERIS-level taxonomy details within a similar tool released at, say, Metricon (http://securitymetrics.org/) is both appropriate and expected by the audience. You must have a solid grasp of who will be using your creations and what their level of expertise and expectations are in order to build a truly successful interactive visualization.

Developing Interactive Visualizations

Yes, you read the section title correctly. Most interactive visualizations will require you to roll up your sleeves and develop code. When we say “development” we are definitely implying that this process is more involved than the fairly light Python and R scripting we’ve introduced in the book so far.

World’s Biggest Data Breaches Processing VIZSweet <http://processingjs.org/>; jquery isotope

Nessus D3 http://d3js.org/

In Summary

For Further Reading

Norman, Donald A. *The design of everyday things*. Basic books, 2002.

Murray, Scott. *Interactive Data Visualization for the Web*. O'Reilly Media, 2013.

Cairo, Alberto. *The Functional Art*. Peachpit Press, 2012.

Foresti, Stefano, and James Agutter. "VisAlert: From Idea to Product." In*VizSEC 2007*, pp. 159-174. Springer Berlin Heidelberg, 2008.

References

1Stefano Foresti, James Agutter, Yarden Livnat, Shaun Moon, Robert Erbacher, "*Visual Correlation of Network Alerts,*" IEEE Computer Graphics and Applications, vol. 26, no. 2, pp. 48-59, March/April, 2006

|  |
| --- |
| 2Schwartz, Barry. "The paradox of choice: Why less is more." *New York: Ecco* (2004). |