# 6 Visualization

*One picture is worth ten thousand words.*

**—Fred R. Barnard (1927)**

*Graphic design which evokes the symmetria of Vitruvius,*
*the dynamic symmetry of Hambidge,*
*the asymmetry of Mondrian;*
*which is a good gestalt,*
*generated by intuition*
*or by computer,*
*by invention*
*or by a system of coordinates,*
*is not good design*
*if it does not communicate.*

**—Paul Rand (1985)**

Security is such a complex topic that it defies easy description. In addition to the natural camouflage afforded the subject by its often-esoteric terminology and concepts, the sheer scope and volume of available security data overwhelms practitioners and laypeople alike. Because of these two facts, most information security professionals have no real idea how to "show security," either literally or figuratively.

It is fair to say that one of the reasons security professionals have so much difficulty dealing with their bosses is because they lack simple and clean metaphors for communicating priorities. Astute analysts recognize—correctly—that visual representations can dramatically enhance managers' abilities to understand security issues. Unfortunately, few analysts have received any formal training on the subject of presentation graphics or (more generally) graphic design. In addition, product vendors generally provide poor or inflexible graphical reporting tools.

This chapter is all about how to get your point across—that is, how to present your hard-won data in a clean and elegant manner that informs, illustrates, and illuminates.

In my view, information security urgently needs fresh thinking about data visualization. Most of what passes for information graphics in the security field generally takes one of two tired forms:

- **Simple bar and pie charts** showing samples of a single coarse-grained metric, such as the number of vulnerabilities found on BugTraq or the number of undesirable e-mails caught by an organization's spam engine
- **Traffic lights** that show the "health" of a range of analysis topics, typically built by hand-coloring reds, yellows, and greens into a grid or thermometer bulb

I find both of these approaches problematic. Bar and pie charts tend to be graphically inefficient; pie charts in particular take up a great deal of space relative to the number of distinct data points they show. In addition, they tend to include only a single metric or data range, rather than (for instance) juxtaposing several ranges.

Traffic lights are worse, because they oversimplify issues too much. In the same way that arithmetic means dilute important points by steamrolling over the outliers (see Chapter 5, "Analysis Techniques"), traffic lights obscure the variation, exception, and detail that lead to insight and smart decision-making.

"But wait," wail fans of traffic lights. "Senior management likes nice graphics. They want something simple. They don't understand the rarefied world of information security. If we give them anything more complicated, they won't understand it!" A former colleague of mine once made a statement like this to me, apparently seriously. I have met many information security professionals who agree with him. But the statement is pure rubbish, and arguably condescending. Want proof that the boss is not a simpleton? Consider a typical stock index chart in the *Wall Street Journal* or *New York Times*. Most charts of the Dow Jones Industrial Index contain these features:

- A time-based horizontal axis, often the last 30 trading days
- High, low, and closing positions for the dates in the range

- Trading volumes for each day in the range
- Often, a 30-day moving average

Doing the math: 4 data points per day, times 30 days, equals 120 pieces of data. These data appear in a compact, two-or-three-square-inch graphic. The boss understands this quite well, thank you very much. Compare this to a traffic light graph that shows exactly *one* data point that is neither accurate nor precise, or with the low-resolution "DefCon"-style bar charts espoused by the likes of Symantec[1] and ISS[2].

As an industry, we can do better than simple pie charts and traffic lights. We need to treat viewers of security metrics data—managers, regulators, and the general public—with more respect. In this case, "respect" means recognizing that intelligent people can, with a minimal amount of training, learn powers of discernment that go beyond nodding and smiling at low-resolution, brain-damaged exhibits.

We need to think of graphically representing metrics as an information visualization challenge, not simply as a "reporting issue." The term "information visualization" is relatively new to the business landscape. Broadly defined, it refers to the practice of using high-resolution graphics and related exhibits to display sets of data, particularly when the sets are large. If the analytical techniques reviewed in the preceding chapter describe ways to uncover patterns in data, information visualization provides methods of showing them off to maximum effect. Visualization concerns include composition, color, typography, arrangement, and use of space (both positive and negative).

Many readers might perk up their ears here and say, "Ah, so you mean making charts!" Yes and no: while information visualization does indeed often mean creating charts, these are means but not ends. Charts are often one part of the larger process of carefully evaluating the best way to present the information at hand.

As mentioned previously, this chapter discusses ways to graphically show off data to their best advantage, without losing the richness and texture that best facilitate deep understanding. Unfortunately, some of the most compelling examples described in this chapter cannot be easily reproduced with standard off-the-shelf office productivity packages. In these cases, I'll describe ways to create the exhibits yourself using custom tools.

A warning to the reader: as if you could not tell already, this chapter is heavily flavored with the strong tastes of my own opinions. If the taste seems excessively bitter, that is because I find more affinity with the aesthetic tastes of graphic designers and high-end management consultants than those of information security vendors and professionals.

---

[1] Symantec DeepSight™ Threat Management System, http://tms.symantec.com.

[2] Internet Security Systems AlertCon™, https://gtoc.iss.net/issEn/delivery/gtoc/index.jsp.

A relative latecomer to security, in my early years I was part of a business team that contracted Boston Consulting Group (BCG) for a seven-figure management consulting engagement. Believe me, several million dollars buys you a hell of a lot of management-grade graphical excellence. Since then, I have been a fan of the management consulting "house style" in general, and of McKinsey & Company in particular. Certain business magazines strongly influence my worldview, notably *The Economist*. Needless to say, the sophistication of visualization used by the organizations I have just listed could not be more different from the sorts of things we have been seeing in information security lately.

This chapter contains three major sections:

- Design principles—six basic rules to live by
- Guidelines for various exhibit formats—theory and practice for sixteen ways to visualize security data
- Thinking like a cannibal—three real-life examples showing how to rework existing exhibits

## DESIGN PRINCIPLES

Before diving into the fun bits (the graphics!), I'd like to lay down some fundamental design principles that will help you create high-impact exhibits. These principles apply equally to all charting and data analysis packages: Microsoft Excel, Keynote, SAS, SPSS, JFreeChart, and others. However, the most common tool used for prototyping business graphics is the spreadsheet. What I am about to say will make the most sense to readers in that context. You can also apply these principles to automated exhibit generation, too, although I leave that as an exercise for the reader.

Generally speaking, mainstream software packages do not serve the cause of information visualization well. The default chart exhibits produced by spreadsheets are far too loud, colorful, and needlessly decorative. Excess chart bloat buries data in an avalanche of shininess, tick marks, unnecessary grids, irrelevant backgrounds, and other foolish bits of graphical frippery. But wisdom, as P.J. O'Rourke one put it, is "knowing the difference between *can't* and *shouldn't*." Just because an analyst can use a program to pollute charts with distracting visual noise does not mean it is a good idea to do so.

This chapter does not attempt furnish a treatise in graphic design. Others, notably the great Edward Tufte, have written beautifully and extensively on the subject already. You should, instead, see this chapter as a summary of effective presentation principles—part *Envisioning Information*, part *How to Lie With Statistics*.

Effective visualization of metrics data boils down to six principles:

- It is about the data, not the design
- Just say no to three-dimensional graphics and cutesy chart junk
- Don't go off to meet the wizard
- Erase, erase, erase
- Reconsider Technicolor
- Label honestly and without contortions

Following these six principles will result in exhibits that are clean, clear, and visually attractive. Let us start with the first one.

## IT IS ABOUT THE DATA, NOT THE DESIGN

Good information visualization is like good graphic design. If the reader does not notice anything amiss, it succeeds: the audience pays attention to the data, not the decoration. But if the reader sees something that prompts a gawk or a head-scratch, the exhibit design may be overwhelming the data.

Data should stand on their own, without extra supporting props or bangles. Forcefully and reflexively check any urges to "dress up the data."

## JUST SAY NO TO THREE-DIMENSIONAL GRAPHICS AND CUTESY CHART JUNK

I have never understood the fascination with three-dimensional pie and bar charts. I am continually astounded at how otherwise respectable security software companies insist on shipping reporting modules that sport ridiculous, gratuitous 3-D graphics. Unless your professional duties include preparing exhibits for the Department of Energy's nuclear weapons simulation program, few conceivable data sets genuinely merit a 3-D exhibit.

Simple, clean, "flat" charts make the same points a faux 3-D chart does, but with less ink. Certainly, ordinary bar charts and pie charts do not require them; the artificial depth only distracts the viewer from the data.

Recent versions of Microsoft's ubiquitous Excel spreadsheet software allow users to add photographs and flashy wallpapers to the backgrounds of charts or to the colored portions of area charts. Avoid these unless the exhibit serves some theatrical purpose. For example, a flashy photo background might feel right at home as part of a sales-oriented slide deck containing scads of music and the obligatory slide transitions. Nobody will take the exhibit seriously anyway, so the extra flash will not matter. But for situations

in which the presenter intends to inform, persuade, or present results of analyses, charts should use white or translucent backgrounds and should omit 3-D.

## DON'T GO OFF TO MEET THE WIZARD

Thanks to the profusion of "wizards," "assistants," talking paper clips, and other assorted digital menservants, modern desktop applications have made it easier than ever to create incredibly busy and tasteless graphics. It *is* helpful that Excel's wizards speed users through the process of selecting data series, titling charts, and labeling axes. However, the results disgorged at the end are, at best, overeager. Even the humblest line chart is festooned with a Technicolor palette, distracting axis tick marks, unnecessary grid lines, and a drab gray background. All these aspects distract the reader from the data.

An additional downside is that Excel's default layout wizards produce a particular, immediately recognizable style, one that screams "amateur"! (For me, spotting Excel punters is an admittedly snobbish, and slightly guilty, pleasure.) Use digital menservants carefully, and only as a starting point for exhibits. Generally speaking, graphics created for all but the most casual personal uses require cleanup.

## ERASE, ERASE, ERASE

Most charts produced by desktop software default settings contain a profusion of superfluous ticks, grid lines, plot frames, and chart frames. There is a good reason why most mainstream business publications use them sparingly: they look clumsy, and they distract attention from the data. You can eliminate all these ornaments without losing any meaning. In fact, your chart will look cleaner as a result.

The general rule: *if you do not need it, erase it.* Start getting into the habit of eliminating the tick marks immediately after creating a chart. Generally this involves formatting the axes with "No major tick marks" and "No minor tick marks." Likewise, eliminate the plot frame and chart frame by formatting each with "No border." These are not needed; the axis lines provide all the framing the chart needs. For bar charts, eliminate the enclosing borders for the bars; the bars themselves provide all the information needed.

Grid lines are trickier. Although I usually erase them, they do have appropriate uses. For sparse exhibits in which subtle comparisons are neither possible nor desirable, omitting the grid eliminates visual noise without sacrificing readability. For dense exhibits containing large data series, however, muted grid lines help readers compare individual data points. When using grid lines, always draw them in a light color (20 to 25% gray) or in black as sparse dots. They should not intrude on the data and should sit in the background.

In fact, other than those required to plot the data, good charts contain no lines other than the x- and y-axes, and (perhaps) some muted grid lines. Even the axis lines can be muted further: try choosing a thin line (1-point) and softer color (50% gray).

The cumulative effect of these erasures results in a crisp chart with few distracting lines. Although my recommendations may seem Spartan—severe, even—the results are worth it.

## RECONSIDER TECHNICOLOR

Make no mistake—when used judiciously and appropriately, color can add tremendous depth and richness to charts and graphs. The eye's ability to make sense of, and discern between, wide ranges of colors is one of the great wonders of the human physiognomy. It is what enables us to discern objects in our peripheral vision or spot a blazer-wearing deer hunter from a long distance.

Tufte has previously noted that small, saturated spots of color are often the best way to draw attention to key points or to outliers in data sets. By that rationale, it stands to reason that many large swatches of saturated color are almost certainly overwhelming to the human eye.

In that light, the default Technicolor palette for Excel charts is less than ideal; the colors are far too saturated for most uses. The default palette includes Lemon Pledge Yellow, Kermit the Frog Green, Ticket-Me-First Red, and Cobalt Blue. For charts with multiple data series, that is quite an eyesore.

To prevent your exhibits from looking like an irradiated piece of luggage as it goes through an airport metal detector, consider these two suggestions:

- **Mute the color palette.** Reds, blues, greens—beautiful colors, all. But they need not saturate the screen. Consider replacing red with burgundy, blue with navy, and "Kermit" green with hunter or forest green. Readers will thank you for it; their eyes will relax rather than twitch. That said, if you need to emphasize a particular data point or series, use a small, focused swatch of saturated color.
- **Use a monochromatic palette.** An alternative to a less saturated palette is one that uses only black, white, and shades of gray. Monochromatic palettes work well when the target output device cannot be guaranteed, *and* when the number of data series is about five or less. A reasonable monochromatic palette includes white (with a black border), 20/25% gray, 50% gray, 75% gray, and black. Use pure colors; avoid fill patterns because they tend to "vibrate." On a related note, because photocopies of good exhibits (like the ones you will produce after reading this book!) tend to proliferate mysteriously into unforeseen hands, get into the habit of printing all exhibits in black and white *first*, before finalizing designs. By "proofing" exhibits this way, you can catch potential reproduction problems before they become an issue.

While I'm on the subject of color, be careful with yellow. There is nothing intrinsically wrong with yellow, but it tends to wash out in printed work and presentations. Use it as a "highlighter pen" accent color, but *not* as a data series color unless the background is very dark.

## LABEL HONESTLY AND WITHOUT CONTORTIONS

Labels matter. Labels convey an exhibit's intent; lack of proper labels leads to loss of clarity and meaning. Label honestly so that readers understand the units of measure, time intervals, and data series—and do it in a professional manner that does not cause torticollis.

A few guidelines are in order. First, *pick a meaningful title* that summarizes the exhibit's main point. A plain title like "Application Security Defects" is fine. More-forceful titles can help too; for example, "Decreased Risk from Applications" succinctly provides the main takeaway message. For charts that display data over a range of time, subtitles help establish the data source and context. For example, a good subtitle might be "Defects reported per application, 2001–2004."

Second, *label units of measure clearly*. Although this sounds simple enough, you might be surprised to see how many people forget to label either the independent or dependent axes, as if the thing being measured were somehow self-evident. Nothing is worse than a beautifully formatted line chart that insightfully points out that over time, a company observed a clear and definitive increase in the number of . . . uh, something.

Axis labels should succinctly describe the unit of measure and scope of each data point and should typically include one of these magic words: "of," "per," "by," or "from." For example:

- Number of defects per application
- Percentage of passwords
- External attacks, by source
- Median number of days per patch

Exception: axes containing units expressed in years do not require labels, since the unit of measure is self-evident.

Third, *do not tilt text toward the vertical* if you're running out of axis room, or, in fact, for any other reason. With apologies to my East Asian and Middle Eastern readers, Western-language text was meant to be read left to right. Slanting x-axis labels or turning them 90 degrees forces viewers to crane their necks. You don't want to be responsible for unwanted chiropractor bills, do you? Of course not. In all seriousness, though, tilted text tends to indicate deeper problems with the exhibit format itself, generally in the orientation. In such cases, try switching the x- and y-axes.

Spreadsheet software (Excel is a notorious offender) often rotates text by default because it believes it is being helpful. Do not let it. Instead, always position chart axis labels with 0° rotation—that is, exactly horizontal.

Fourth, for multiseries charts, *consider eliminating series legends* if you can get away with it. Place the series labels directly on or near the data series themselves—that is, at the point of use. This practice works especially well with line charts.

Fifth, *do not abbreviate.* Although it may seem more efficient to label axes with "nmbr.," "app.," and "bus," doing so forces readers to unconsciously pause while reading the chart, an unnecessary distraction from the data. Also, abbreviations look sloppy. Of course, any rule has exceptions. For example, most people understand that *%* stands for "percentage" and that *IT* denotes "information technology." In most cases, though, try expanding all abbreviations. If narrow space on the y-axis forces an abbreviation, try giving the axis more breathing room by widening the left margin.

Sixth, *use simple and consistent fonts.* Charts are not the place to trot out that new typeface downloaded from the Internet. Use classic sans-serif typefaces like Helvetica, Franklin Gothic, or plain old Arial. In addition, keeping text the same size throughout the chart helps readers focus on the data, rather than the labels. Therefore, as a general rule, all labels other than the title (axes, data, subtitles) should be the same size and font. For printed documents, I recommend 9-point Helvetica plain or 9-point Arial plain. For space-constrained exhibits, the "narrow" versions of these fonts work pretty well, too. Opinions differ on correct formatting of titles; I prefer to make them the same size and font as the other labels, but in boldface.

Finally, *cite any data sources used to make exhibits.* To make a citation, place a small, short caption at the bottom of the exhibit. A simple "Source: Security Metrics Study (1999–2004), Andrew Jaquith Institute" in 6-point type (or something similar) works nicely. In addition to making the exhibit look more official, the caption provides valuable information to readers about sources and methods.

## EXAMPLE

Although my suggested design guidelines may seem onerous, when followed they can dramatically improve the look and feel of metrics exhibits. For example, consider the very basic password-quality data set in Table 6-1.[3] The analyst has decided to create a graphical exhibit for management showing the results of the latest password audit. He fires up Excel and selects a standard bar chart (formatted in 3-D because it "looks cool"). Figure 6-1 shows what Excel disgorges when using default settings.

[3] This example is deliberately simplistic; normally, very small data sets should be presented in a table format, *not* a chart.

**Table 6-1** Sample Password Data Series

| Department | Value |
|---|---|
| IT | 230 |
| Account. | 22 |
| Ops. | 129 |
| Sales | 40 |

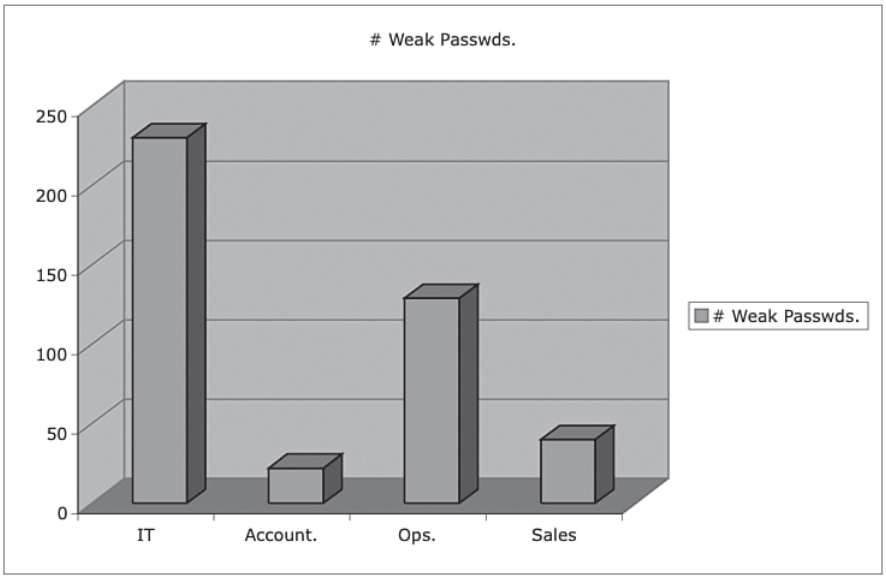


**Figure 6-1** Initial Exhibit for Password Data Series

What is wrong with this picture? All sorts of things:

- Gratuitous 3-D effect
- Abbreviated category names
- Unnecessary legend
- Grid lines add no value
- Distracting shadows and background
- No data labels

Let's clean this up. Figure 6-2 shows a redrawn version of the exhibit. I made quite a few changes:

- Specified a sensible chart title indicating what the exhibit signifies—"Results of Password Audit by Department"—and a relevant time interval—"March 2005."
- Added a y-axis label, "Number of Weak Passwords."
- Eliminated the horizontal grid lines.
- Removed the series legend.
- Added data labels above each bar.
- Removed the tick marks from both the x- and y-axes.
- Removed the series border around each bar and changed the color from lilac to navy blue.
- Harmonized all labels to use the same typeface (Arial instead of Verdana), size (9-point), and style (plain, except for the title in boldface). Also, cleared the "auto-scale" check box for all text items.
- Removed the plot area border and background fill.
- Removed the chart area border and background fill.

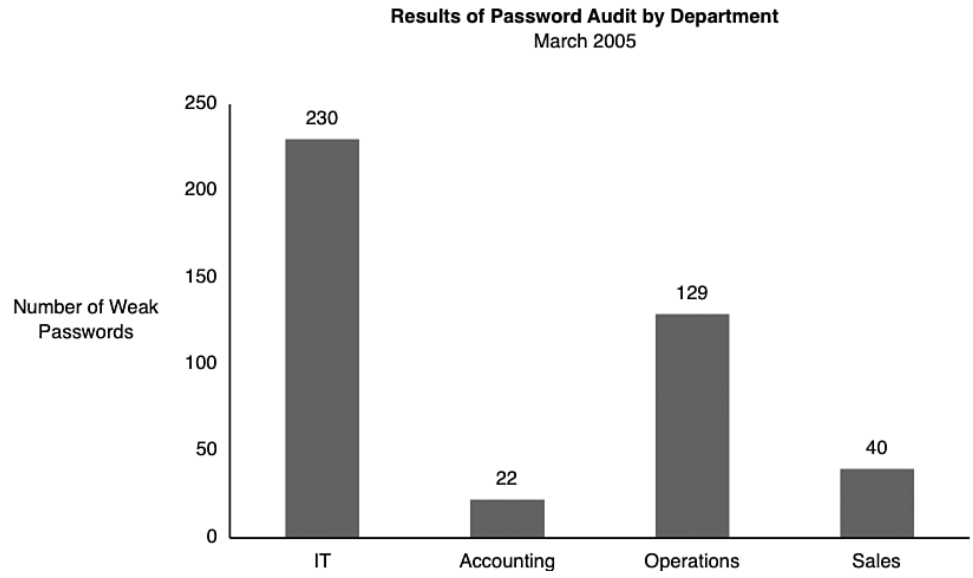


**Figure 6-2** Redrawn Exhibit for Password Data Series

We can still improve this exhibit further by making some additional changes to the format. First, switching the axes provides additional flexibility for the department names and looks more professional. In addition, sorting the departments in descending order of the data points strengthens the exhibit's message. Finally, reducing the exhibit's overall size saves some space. Figure 6-3 shows the chart in its final form.

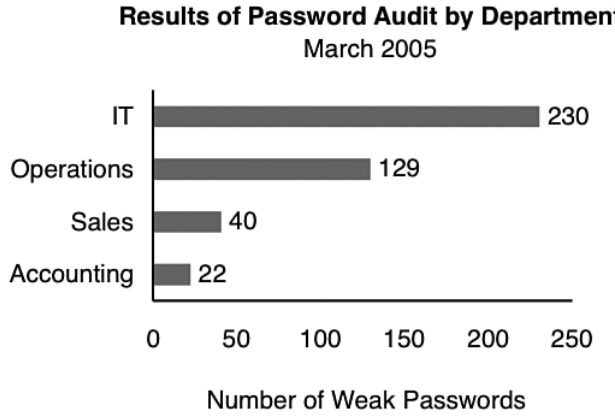


**Figure 6-3** Redrawn Exhibit for Password Data Series (2)

As an alterative form, some business magazines substitute thin tick marks in place of the x-axis line. That looks good too, and proves that judicious use of tick marks can pay off.

## STACKED BAR CHARTS

The preceding section discussed some commonsense guidelines for redrawing a sample exhibit produced using Excel's default settings. The chart format used in the example was the venerable bar chart—a format most readers should be familiar with.

This section introduces a variation on the bar chart—the stacked bar (or column) chart. When comparing categorized data across multiple time periods, stacked bar charts are a reasonable choice, when two conditions hold true:

- You need to analyze more than two time periods
- The number of categories being compared do not exceed about a half dozen

Two variants of the stacked bar chart exist—one that normalizes all values relative to their percentage share of the total, and one that does not. Readers of *The Economist* should recognize the former. Figure 6-4 shows a sample "normalized" stacked bar chart created using a spreadsheet package.

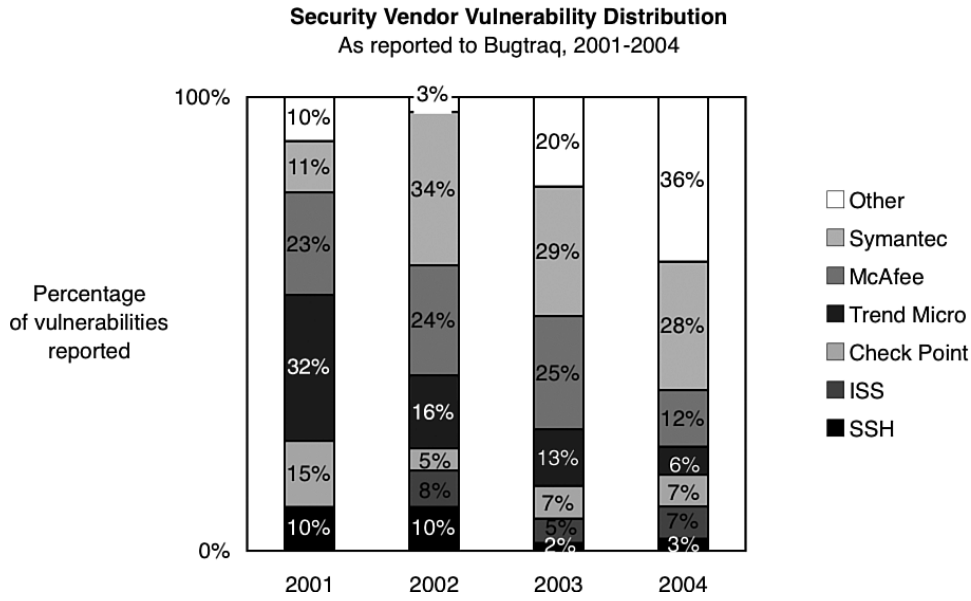


**Figure 6-4** Stacked Bar Chart (Normalized)

To increase readability, I tweaked the chart as follows:

- Grouped all data rows after the top six into a catchall category called "Other."
- Removed the background fill and all tick marks.
- Simplified the y-axis labels so that they show only the minimum (0) and maximum (100%) values.
- Manually reversed (and put in boldface) the series labels that lie on dark-colored backgrounds.
- Stretched the chart legend vertically to better align the legend labels with the bar positions. Note how the chart legend items line up, horizontally, with the items in the stacked bars.

- Manually added an opaque white background to the "3%" label.
- Changed the color of the "Other" category to white (so that it is less noticeable than the named categories).
- Tweaked the color scheme so that the shades of gray alternate between light and dark (to improve contrast between categories, and thus readability).

Figure 6-5 shows the same data, but plotted using the second, non-normalized stacked bar chart variant.

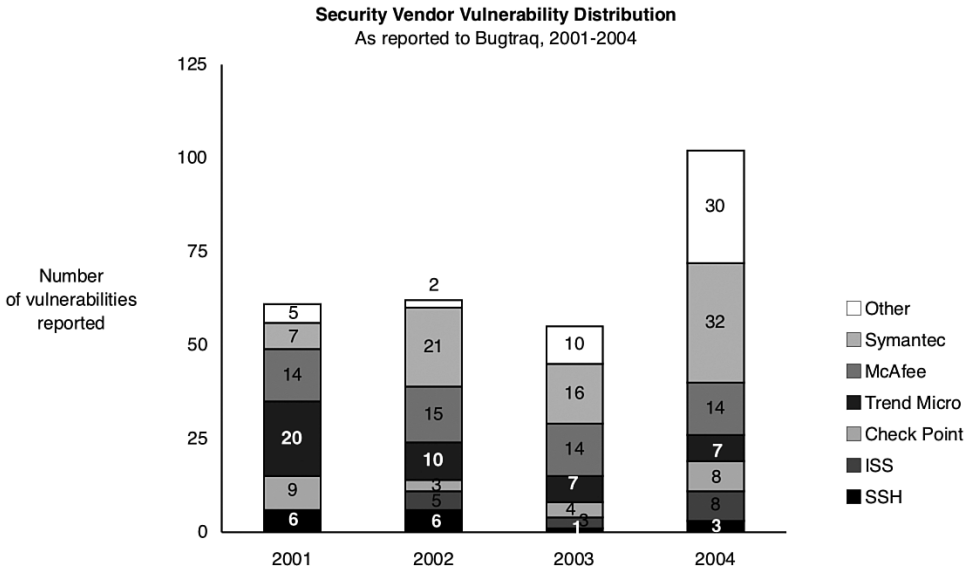


**Figure 6-5** Stacked Bar Chart (Nonnormalized)

I do not use stacked bar charts often, but they can be useful when the number of data series and corresponding points are relatively small. The primary advantage of stacked bar charts is that readers recognize them fairly readily.

## WATERFALL CHARTS

Popularized by McKinsey and Company, the "waterfall chart" provides a flashier alternative to the stacked bar chart. It is best used in relatively simple exhibits where the analyst is trying to show the relative contributions of different factors to a larger total.

For example, in the past I have used waterfall charts with executive audiences to illustrate how particular categories of security vulnerabilities contribute to an overall risk score.

A typical waterfall chart (see Figure 6-6) contains the total number at the top, represented as a horizontal bar. Bars arrayed underneath "explode" the component numbers onto separate rows. Numbers for each row appear to the right of the bar. A dashed line typically separates each bar.

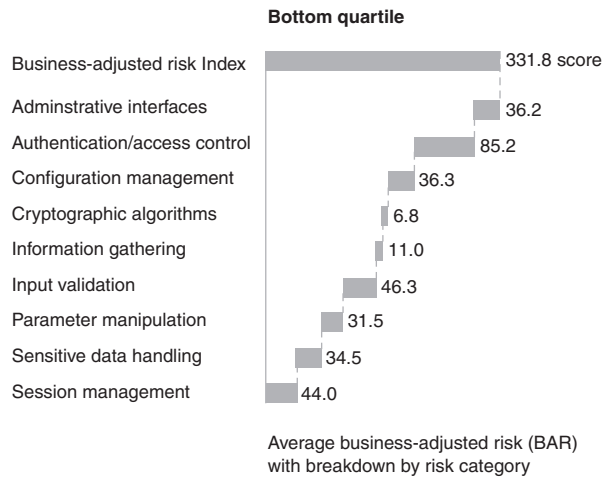


**Figure 6-6** Waterfall Chart

Waterfall charts tend to be more readable and less claustrophobic (or space-efficient, depending on your point of view) than bar charts. Waterfall charts look neater, and facilitate comparisons better, than the equivalent stacked bar chart.

A side benefit of waterfall charts is that they can be used in small-multiple exhibits, but only when done with care, and with multiples of perhaps two at most. Figure 6-7 shows a "side-by-side" small-multiple variation of the waterfall chart that looks good and tells its story well.

Few software packages exist that can create waterfall charts; in most cases, analysts must hand-draw them using a graphics package like Visio, ConceptDraw, Adobe Illustrator, or (in a pinch) a presentation package such as PowerPoint. As an alternative to hand drawing, waterfall charts are also good candidates for automation. A few lines of Perl or Java, for example, can easily generate vector graphics for waterfall charts.

Waterfall charts are good for high-end management presentations, but that is about it. Their legibility degrades quickly after about a dozen rows. For data sets that are highly dense, consider treemaps instead. I discuss treemaps later in this chapter.

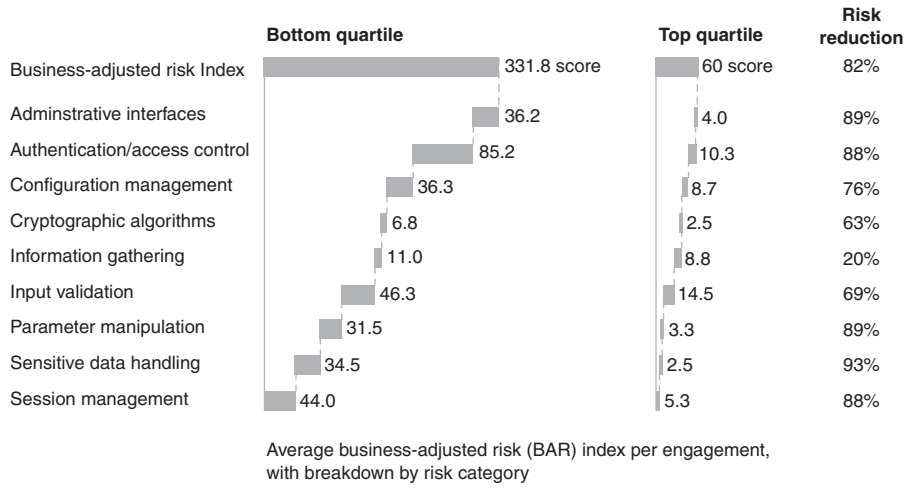


**Figure 6-7** Small-Multiple Waterfall Chart

## TIME SERIES CHARTS

Time series charts are probably the best-known technique for visualizing security metrics. They remain the most common form of exhibit in security information reports, and they figure prominently in products for measuring compliance or tracking vulnerabilities.

### BASIC TIME SERIES CHARTS

Chapter 5 discussed how a *time series* captures a set of consistently measured data records over an interval of time. Each record contains a number of data attributes. Time series charts simply graph an attribute (or set of attributes) over a time interval. The time interval (generally days, months, quarters, or years) serves as the independent variable and usually appears on the horizontal axis. The attribute(s) that vary over time serve as the dependent variable(s) and appear on the vertical axis.

Variations on the basic time series chart exist. Clever analysts occasionally add a second vertical axis on the right side of the exhibit to display a contrasting attribute in the same exhibit. Most readers are likely familiar with financially oriented time series charts (*The Economist*, *Business Week*) that show, for example, interest rates on the left and money supply on the right.

Time series charts accommodate a number of formats, depending on the preferences of the security analyst. Formats that work well include

- Line charts
- Area charts
- Bar charts

Each format has strengths and weaknesses. Personally, I prefer line charts for exhibits used in isolation. In an individual exhibit, the direction and tendency of the series line matters most; bars and colored chart areas distract the reader. But for small-multiple exhibits (discussed later in this chapter), area charts can help imbue individual exhibits with stronger "shapes" that are better distinguished by the eye.

Figure 6-8 depicts a sample time series graphic, drawn as a line chart. It shows the number of infections for the 2001 Slammer worm, based on data from the Cooperative Association for Internet Data Analysis (CAIDA).[4] When I prepared this chart, I wanted the infection trend line to be the most prominent characteristic. Note how the x- and y-axes are relatively plain and thin, while the data series itself appears as a thick line drawn in saturated blue.

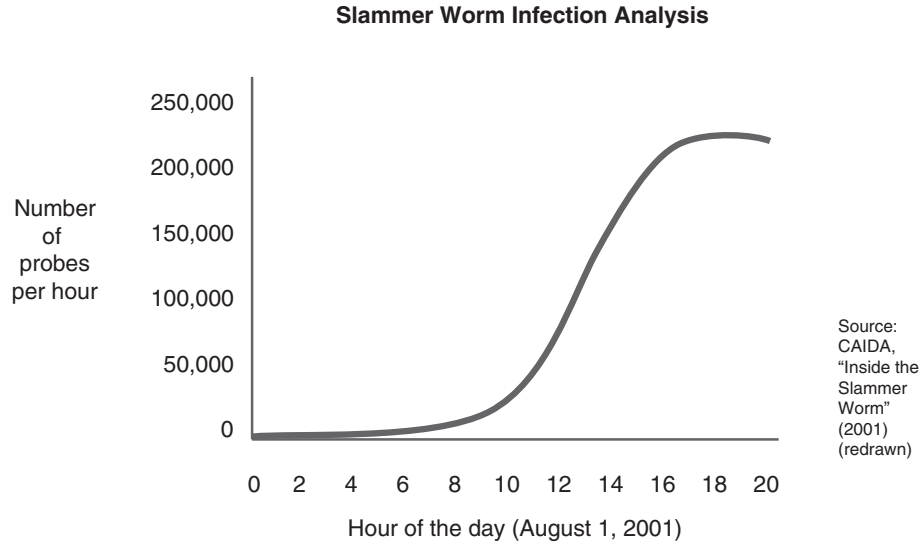


**Figure 6-8** Time Series Chart of the Slammer Worm

[4] See CAIDA's examination of the Slammer worm at http://www.caida.org/outreach/papers/2003/sapphire2/sapphire.xml.

Time series charts are perhaps the most easy-to-understand form of information graphics. Everyone—managers, staff, and laypersons—knows how to interpret what they mean. Every graphics package worth its salt supports one or more of its forms. And unless the analyst commits a horrible labeling blunder, they are nearly impossible to screw up.

### INDEXED TIME SERIES CHARTS

Popularity and wide tool support mean that time series charts make a good starting point for visualizing security metrics. One of the more common applications of time series charts is for displaying improvement over time against a baseline. By "baseline" I mean a set of measurements taken at a particular point in time. A twist on the venerable time series chart, therefore, is an "indexed" version that charts each data series relative to the baseline.

To create the baseline, the analyst selects a starting point in time and normalizes all dependent data series values at that point to some "base" index value. I prefer normalizing to the number 100 because it corresponds to the "report card" or "IQ score" scales that most people are familiar with. As a side benefit, it displays fairly nicely and can show up to two significant digits of precision if required.

Normalization of the data series values to the baseline value produces a chart in which all values emanate from a common baseline origin point and diverge from that point forward. The normalization, in effect, encourages the viewer to trace the pathways of each series over time.

Figure 6-9 shows a sample indexed time series chart depicting the number of security vulnerabilities in several types of software for the period of 2001 through the first quarter of 2005. It uses 2001 quarterly averages as the baseline values. The chart clearly shows that the number of vulnerabilities for Microsoft products dropped well below its 2001 baseline early in 2003 and has not yet returned to that level. In contrast, the number of vulnerabilities for security products increased in early 2005 to nearly 50 percent (indexed value: 151) over the 2001 baseline.

Indexed time series charts challenge readers to compare and contrast rates of change among divergent data series over time. As a result, they are best used to show comparisons of measurements taken over time against understood baselines.

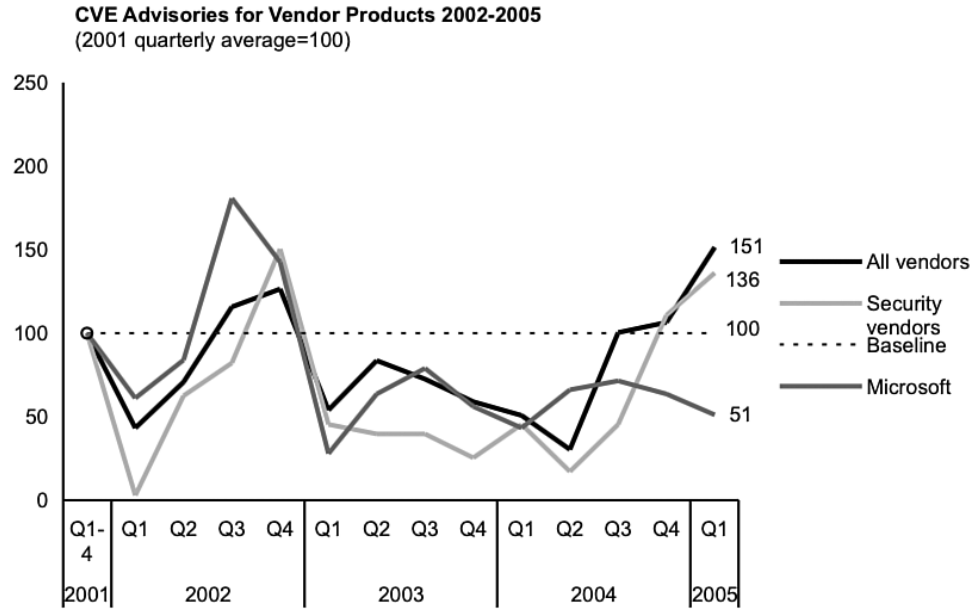


**Figure 6-9** Indexed Time Series Chart

## QUARTILE TIME SERIES CHARTS

Indexed time series charts showcase one way to revisualize sets of time series data by normalizing to a baseline. Another variation on the time series chart, which I refer to as the "quartile time series chart," showcases another technique. It uses quartile information from data sets to show broader measures of performance over time.

As you may recall from Chapter 5, quartiles group data into four bins: the top 25% of the data points in the sample comprise the first or "top" quartile, and the bottom 25% form the fourth. The last element in the second quartile, in fact, is the *median* data point in the set.

To create a quartile time series chart, the analyst calculates the first, second, third, and fourth quartile numbers for each time interval in the data set. The resulting exhibit simply graphs the first, second, and third quartiles. Figure 6-10 shows a sample quartile time series chart. Notice how the exhibit omits the fourth quartile; since it represents the upper bound of the data set, including it would only add visual noise.

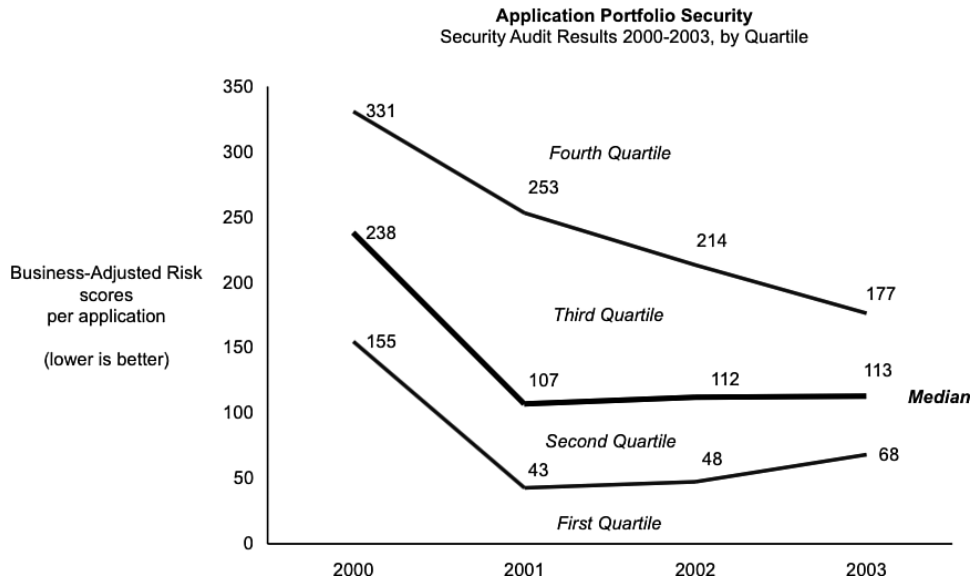


**Figure 6-10** Quartile Time Series Chart

The way to read the exhibit is straightforward: the thick line represents the median values that separate the second and third quartiles. The thin line below the median separates the first and second quartiles, and the thin line above the median separates the third from the fourth. Based on the positions of the lines, viewers can quickly identify the correct quartile that any other data point falls into. Although the time series interval in the example I have provided is fairly broad (yearly samples), the broad headlines from this exhibit announce themselves:

- The period from 2000 until 2001 saw the most dramatic improvement (a 50% drop in median scores).

and

- Since 2001, median scores have stayed fairly flat.

but

- The worst applications (fourth quartile) demonstrated continuous improvement through all periods.

and

- All quartiles appear to be converging, which means that application security scores are generally improving across the board (specifically, the difference between the first and third quartile lines decreases over time).

although

- The first quartile has worsened in the most recent year (2003) relative to the previous one.

This chart contains two minor graphical refinements worth mentioning. First, all the quartile lines appear in the same color (black). However, the line that arguably provides the most context—the median—was drawn thicker (a 3-point line instead of 1-point). Second, I have added free-form text labels (italicized) to clearly establish the territories occupied by each quartile and to identify the median (italicized and bold).

In addition to these refinements, analysts can plot additional data points to show which quartiles they fall into. This is extremely useful for answering a common question from management about a particular item (namely, "How did we do?"). In fact, an analyst could combine the quartile time series line plots with a scatter plot showing the scores for selected (or all) data points in the set.

Alternatively, the analyst could create what I refer to as the "You Are Here" benchmarking chart by adding a horizontal line representing the score for a particular data point being benchmarked. The line crosses the y-axis and extends the width of the chart. When I was a consultant at @stake, for example, we used this technique to show how a client's freshly assessed application scored relative to our first/second/third/fourth quartile benchmarks. Clients liked the "You Are Here" chart because it showed how their applications ranked—that is, which quartile they fell into. From the consultant's point of view, the "You Are Here" chart helped drive business because it made the point that the client's application would have ranked better (or worse) in different periods.

Quartile charts excel in revealing how data change over time. They dig below the surface by graphing more than just simplistic averages or means. I rarely see them used, and that is a shame. Make them part of your toolset.

## BIVARIATE (X-Y) CHARTS

As noted earlier, time series charts are the most common information security visualization technique. The unadorned time series chart is like an old reliable friend who speaks plainly and always shows up on time. Everyone knows what to expect, and he rarely

disappoints. But he is not too bright, and his insights are rarely very penetrating. His slightly more flashily clad brothers—the indexed and quartile-based time series charts—offer a bit more excitement but not necessarily any extra insight.

In contrast, bivariate charts—which show how two variables interrelate—resemble cranky uncles more than old friends. They offer lots more insight and wisdom but require readers to take more time to understand their unique qualities (or eccentricities, if you prefer). You can't just inflict an uncle upon the uninitiated.

Bivariate charts gain their power from the often-unexpected linkages one finds by plotting two variables from the same data set on the same page. Each variable corresponds to an attribute in the data records being analyzed. When charted together, cause-and-effect relationships often emerge. Note that bivariate charts are time-independent; that is, they do not show temporal relations in data the way time series charts do. In most cases, bivariate charts display data for a given time interval; make sure to note the relevant interval in the chart's title.

Let us try a security example. Recall the application security defects data set discussed in the preceding chapter (in Table 5-2). The data set contained instances of security defects in a developed application. Each record in the data set contained these attributes:

- Application
- Owner
- Defect
- Date
- Exploitability
- Business impact
- Business-adjusted risk (BAR)
- Engineering fix hours

An analyst could use a bivariate chart to show how two of these attributes relate. A hypothetical chart might show one of the following:

- Exploitability versus business impact
- Business-adjusted risk versus engineering fix hours
- Business impact versus engineering fix hours

Figure 6-11 shows all three of these charts using the sample data set from Table 5-2. To show the relationship between the x- and y-axis variables explicitly, I have added a logarithmic regression line for the latter two charts.

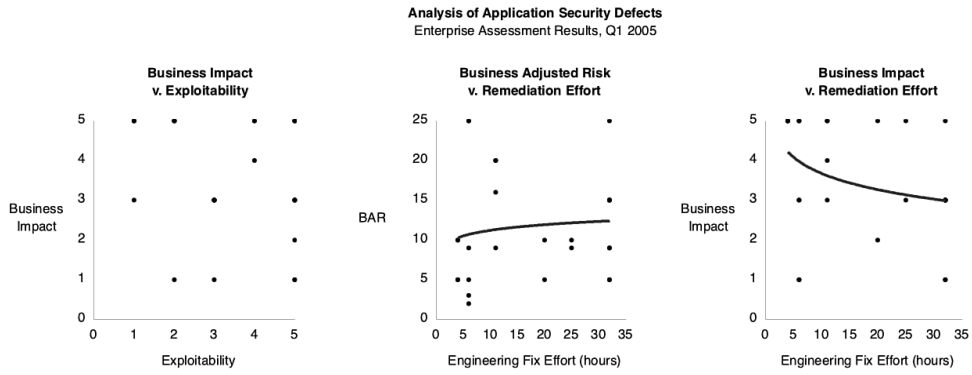


**Figure 6-11** Sample Bivariate Charts

Casual examination of the two rightmost charts suggests that a weak relationship exists between remediation effort and either business impact or business-adjusted risk. That fits; one can reasonably expect that more serious security flaws will take longer to fix. In contrast, the left chart suggests that no obvious relationship exists between business impact and exploitability. This, too, seems to align with expectations—exploitable security holes do not possess any intrinsic qualities that would cause them to also be high-impact.

Exploring relationships between variables in a graphical way can help confirm or deny an existing hypothesis. For example, an analyst reviewing the exhibits in Figure 6-11 would not be able to make strong, definitive statements about cause-and-effect relationships between business impact, exploitability, and remediation effort.

Some bivariate charts show much stronger relationships. Figure 6-12 shows a fictitious bivariate chart that displays the relationship between end-user training and password strength, as measured by a password-cracker like John the Ripper. In this case, the relationship between the cause (how long since the last user training session) and the effect (the relative security of passwords) is much clearer. I have added a logarithmic trend line to highlight the relationship; a linear trend line works well also.

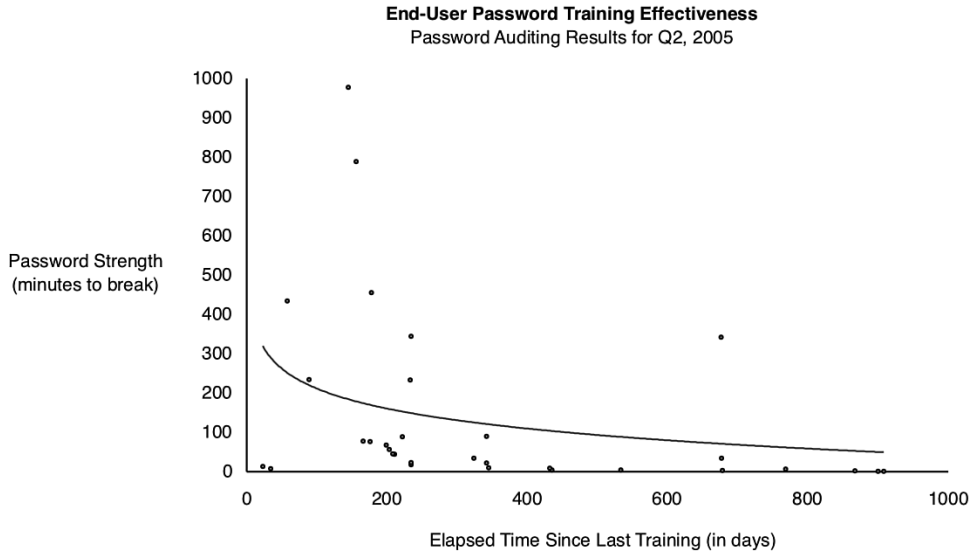


**Figure 6-12** Password Effectiveness Bivariate Chart

## TWO-PERIOD BIVARIATE CHARTS

Although bivariate charts cannot display temporal relationships as well as time series charts can, they can show comparative data in a limited way. A variation on the standard bivariate chart, the "two-period" bivariate chart, plots each period's data series and connects interperiod points with thin lines. Different markers distinguish the "before" and "after" points. The overall effect resembles a football or basketball chalkboard diagram. Figure 6-13 shows an excellent two-period chart from *The Economist* of a Boston Consulting Group study of investment banking revenues and corresponding value at risk (VAR).[5]

Two-period bivariate charts are a relatively specialized breed; they do not work well with sets larger than about a dozen pairs of data points. In addition, mainstream desktop packages like Excel cannot create them, so they must be hand-drawn.

[5] *The Economist*, "Happy Days," Dec. 29, 2004.

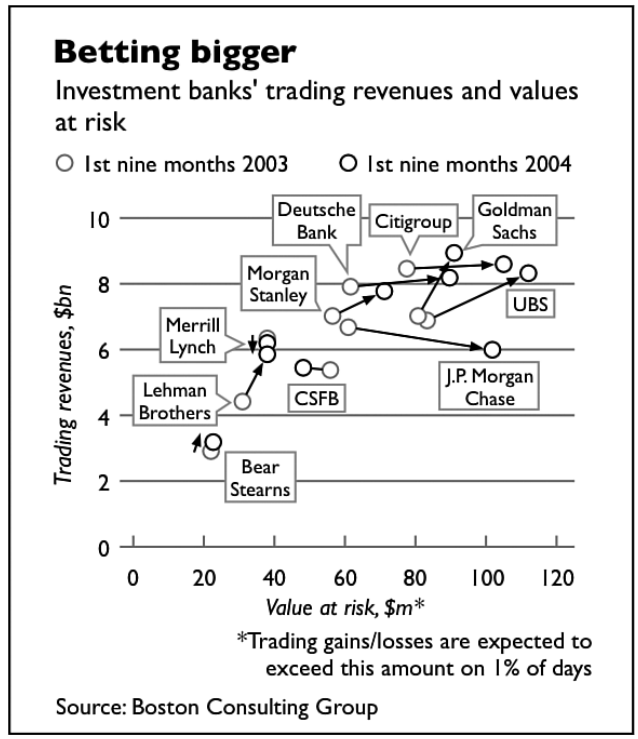


**Figure 6-13** Sample Two-Period Bivariate Chart (Redrawn)



## SMALL MULTIPLES

Curious people like to probe, ask questions, and understand why something is the way it is. One of the most powerful ways to satisfy a person's curiosity is to provide ways to compare and contrast. People instinctively know how to compare before with after, apples with oranges, and like with unlike. Why do they do this? In part so that they can understand relationships by spotting the differences.

In the field of information graphics, a concept called *small multiples* provides a natural, instinctive way to show how things relate and, more importantly, how they differ. First popularized by Edward Tufte, small multiples plot several cross sections of data in separate mini-charts and then combine them into a single exhibit. As a result, readers can—at a glance—quickly sweep back and forth across the exhibit, looking for patterns, similarities, and differences. An important quality of the small-multiple chart is that the

axes remain constant with respect to their units of measure and scales. Only the data cross sections being plotted change.

Figure 6-14, a screen capture from the distributed network intrusion detection project DShield, shows how small-multiple exhibits work. The small multiples are in the column labeled "Activity Past Month." They show the relative number of hostile scans encountered for the network services enumerated in the "Service Name" column. Although the x- and y-axis labels are not shown, it seems clear enough what they must be: the vertical axis scale starts at 0 and increases at a linear—or possibly logarithmic—rate to maxima held constant in all graphs. The x-axis shows how the relative number of scans varies over time. Minor quibbles about labeling aside, the use of small multiples in this exhibit enables the reader to quickly get a sense of which ports are most likely to be scanned. In this case, they are 445 and 135—two ports associated with Windows services that are often prone to compromises. A network administrator running an all-Windows environment, for example, might see this exhibit and decide to push out a group policy temporarily restricting access to these ports.

| Service Name | Port Number | Activity Past Month | Explanation |
|---|---|---|---|
| epmap | 135 |  | DCE endpoint resolution |
| microsoft-ds | 445 |  | Win2k+ Server Message Block |
| --- | 1026 |  | |
| icq | 1027 |  | icq instant messanger |
| ms-sql-s | 1433 |  | Microsoft-SQL-Server |
| ms-sql-m | 1434 |  | Microsoft-SQL-Monitor |
| radmin | 4899 |  | Remote Administrator default port |
| netbios-ssn | 139 |  | NETBIOS Session Service |
| --- | 1028 |  | |
| smtp | 25 |  | Simple Mail Transfer |

**Figure 6-14** Sample Small-Multiple Exhibit[6]

[6] This chart was obtained from DShield, http://dshield.org/topports.php.

One can easily imagine how this exhibit could be enhanced. Instead of simply showing the "top 10" most-scanned ports, we could show the top 100, or a subset of the most common well-known ports. Doing so would require some graphical nips and tucks. The "Explanation" column would need to vanish, and we would want to combine the "Service Name" and "Port Number" columns. From the point of view of aesthetics, representing the scan results as solid filled area charts on a white background (instead of black) could increase the small-multiple format's readability.

An intriguing small-multiple format that would work well here is the *sparkline*—a minimalist "simple, intense, word-sized graphics" format invented by Tufte.[7] Figure 6-15 shows a fictitious redrawing of the preceding exhibit using sparkline format, constructed using Excel. Each mini-chart includes a dark gray line to show the trend for each cross section, as well as a light gray band denoting the "normal" range—that is, the mean value plus or minus the standard deviation. So that the reader can understand the plot lines in context, the final data point in each series is highlighted with a red marker and numeric label.

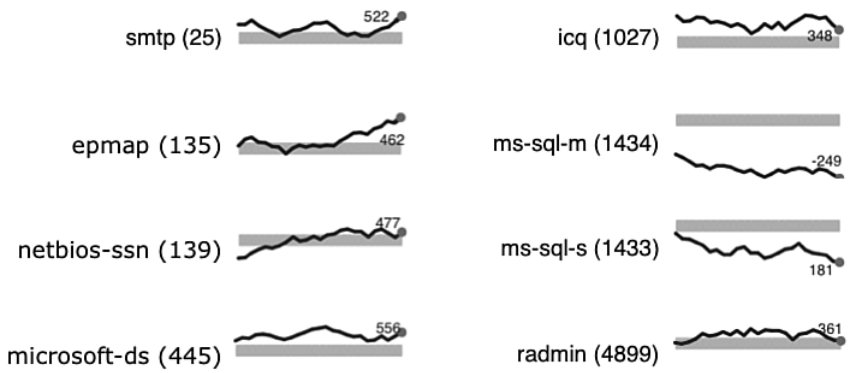


**Figure 6-15** Sample Small-Multiple Exhibit (Sparklines)

## QUARTILE-PLOT SMALL MULTIPLES

The time-series-oriented small multiples in Figures 6-14 and 6-15 help the reader understand the relative magnitude of activities over time. But time series charts are not the only type of graphic that can be used as a small multiple. Figure 6-16 shows a hand-drawn small-multiple exhibit using bar charts that compares and contrasts the

[7] See E. Tufte, *Beautiful Evidence*, Graphics Press, 2006.

distribution of security flaws across nine different application security areas for a selected group of applications.[8] Each multiple contains a vertical bar chart displaying the area's first and fourth quartiles in the data set.

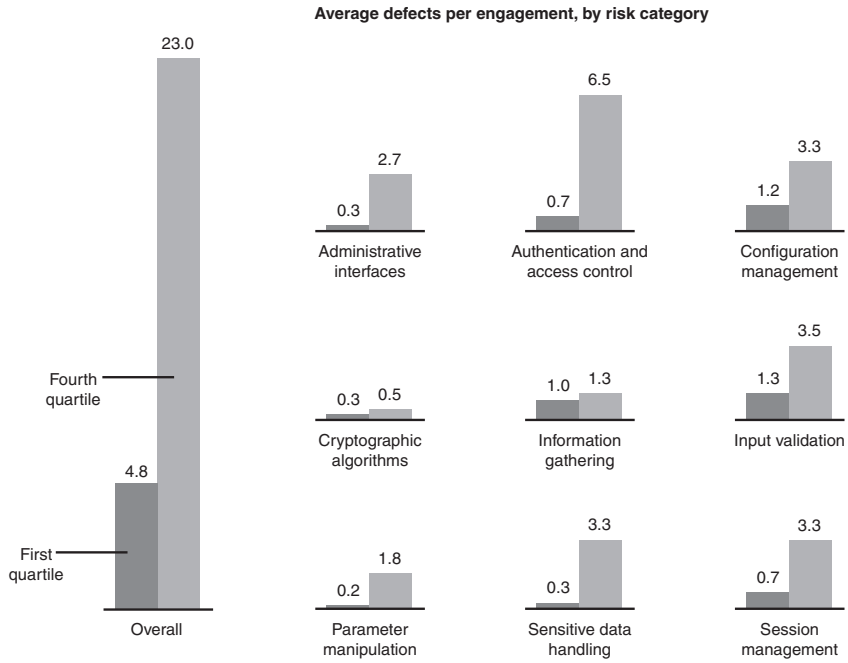


**Figure 6-16** Sample Small-Multiple Exhibit (with Quartiles)

The combination of the small-multiple format with a first-versus-fourth comparison yields an extremely powerful graphic. A simple glance at the exhibit reveals the headline: fourth-quartile applications are much worse than their first-quartile counterparts in some areas, but not others. For example, the "best" applications contain 90% fewer authentication defects, have 90% fewer problems related to sensitive data handling, and suffer from 80% fewer session management issues. In contrast, the number of cryptographic issues are few across the board, and the difference between the best and worst applications is not large.

The exhibit is interesting for another reason: it sports a "layered" macro/micro design that shows both the overall total (on the left) and the contributions made by individual

[8] A. Jaquith, "The Security of Applications: Not All Are Created Equal," @stake, Inc., 2002.

multiples. The scaling factors for the y-axis remain the same for both, and the quartile labels on the "overall total" graph serve as a key. Not all small-multiple exhibits lend themselves to such an elegant format, but it is nice when they do.

Small multiples, while powerful, are not well-supported by mainstream spreadsheet packages. For example, due to lack of better methods, a security analyst would need to hand-draw Figure 6-16 using Visio or a similar drawing package. Some careful spreadsheet jockeying in Excel might also work, although to do so would require the analyst to painstakingly format and align each multiple down to the pixel—and pray that Excel doesn't move or reformat it. However, in most cases analysts would do better to generate the individual multiples using a script and then stitch them together programmatically into a web page or PDF.

## TWO-BY-TWO MATRICES

A special form of the bivariate chart, two-by-two (2×2) matrices became popular in the 1960s and 1970s with the advent of modern management consulting. Sometimes also called quadrant charts, two-by-two grids cluster the data in an X-Y plot into four boxes, divided by crosshair-like grid lines at the center of the chart. Two-by-two matrices abound in management circles and have even spread to the lay public. Steven Covey's "Seven Habits" Urgency/Importance grid, for example, introduced the technique to millions of readers. And the Gartner Group's "Magic Quadrant" 2×2 matrix is well known to IT professionals.

As a management consulting device, the 2×2 matrix offers a powerful framework for analyzing business problems. The Boston Consulting Group's Growth-Share Matrix, shown in Figure 6-17, was one of the early 2×2 matrices. Its success with clients helped the firm establish its reputation as a leading strategic adviser. The technique is fairly simple: the analyst plots the company's set of products according to the growth rate of each product's market segment (y-axis) and its share of the market relative to competitors.[9] Products with high relative shares in fast-growing markets are designated "stars." Those in slow-growing markets are considered "cash cows" if they command a significant share, and "dogs" if not. Finally, slow-selling products in high-growth markets are called "question marks" because they could transform into stars if their market shares go up, or wither into dogs if the overall market cools.

[9] Variations of the Growth-Share Matrix include an optional "bubble plot," in which the diameter of each product scales in proportion to its revenues. To keep the example simple, I have excluded this feature.

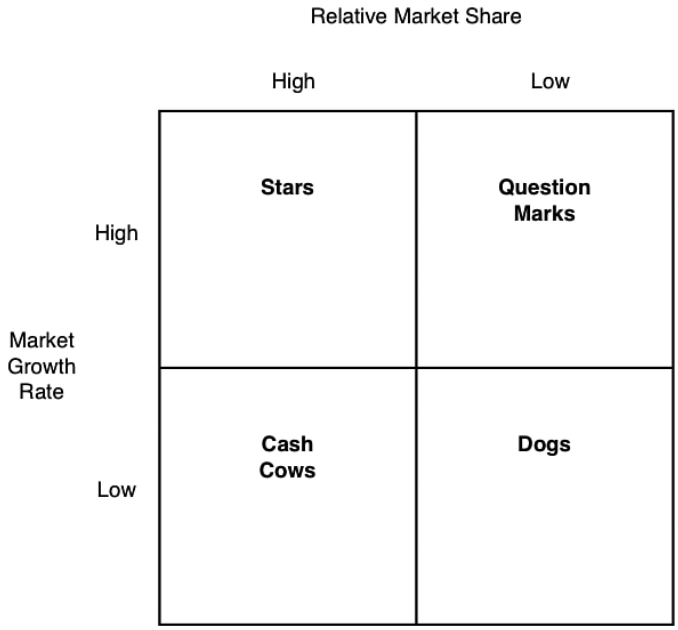


**Figure 6-17** The Boston Consulting Group Growth-Share Matrix (2×2 Matrix)

The Growth-Share Matrix might seem simple—deceptively so, in fact. Yet this form of 2×2 grid has proven to be tremendously resilient because it:

- **Speeds comprehension by grouping data into simple buckets.** Most good managers are pattern-finders; they want to make sense of things. Classifying a potentially large data set into quadrants speaks to this natural human impulse.
- **Facilitates cross-sectional comparisons.** Within each quadrant, the reader can perform rapid comparisons among data points as a way of understanding how quadrant members are alike (or different).
- **Exposes the analytical process.** Bivariate plots (plots containing two variables) such as the BCG matrix use two labeled, quantitative axes to display results. The reader understands the explanation for each point's plot position simply by looking at the axis labels.
- **Presents a small, logical set of management options.** In addition to the designated label, each quadrant contains straightforward advice. In this example, the options are to invest in stars, maintain cash cows, selectively invest in question marks, and divest the dogs.

These benefits accrue to all 2×2 grids, not just the BCG Growth-Share matrix. In fact, the logic and power of 2×2 grids have proven sufficiently compelling that authors have devoted entire books to the subject.[10]

In addition to these explicit and obvious benefits, 2×2 grids contain an unintended—rather subtle—consequence. By its nature, the 2×2 grid constrains the decision space. It is a compartmentalized box; all the data in the analysis *must* sit in one of the four compartments. By carefully choosing the quadrant boundary values and labeling scheme, the analyst can literally frame the decision-making process.

I can personally testify to the power of having a constrained decision space by relating a story about @stake's signature exhibit, the Business Impact Matrix. Early in the spring of 2000, I successfully sold and managed @stake's first contract, a $35,000 engagement to assess the security of a business-to-business commerce website. Of course, since this was the first engagement in the company's history, we needed to invent everything from scratch—the document template, the graphic design, boilerplate text, and exhibits. Fortunately, we had budgeted sufficient time to prototype most of the essential parts of the document.

On the day before the due date, my technical team was busily writing up the detailed findings portion of the document per our agreed-upon formats. I took responsibility for writing the Executive Summary. As overall project leader and lead consultant for the firm, I was painfully aware that presentation and conciseness mattered. In particular, I wanted to make sure that the Summary came in under two pages, including a nice snappy graphic that summarized our technical findings. Figure 6-18 shows the final exhibit for the engagement—a classic 2×2 grid that I later named the Business Impact Matrix.

The Business Impact Matrix displays three attributes for each security defect: the degree of exploitability (y-axis), cost to fix (x-axis), and business impact (size of bubble). All attributes are normalized to a 1-to-5-point scale.

Although I am naturally biased on the subject, it is fair to say that the Business Impact Matrix contributed more to the early success of the firm than anything else we did. Of course, it helps to understand what the sleepy security consulting world was like in 1999 and 2000. Most of our competitors—Big Five accountancies—tended to send in kids with network scanners, who would drop off phone-book-sized reports at the end of engagements. Some of the newer, pure-play consultancies like Guardent (now part of VeriSign) and Rampart Security (later renamed Foundstone, now part of McAfee) appeared to do much the same thing.

Against this backdrop, the Business Impact Matrix represented genuine innovation. Our clients loved it; sometimes they would literally tear off the first few pages of the

[10] The 2×2 grid proved so ubiquitous that a pair of business consultants decided to write a book on it. See A. Lowy and P. Hood, *The Power of the 2×2 Matrix*.

report (with the exhibit) and send it to their bosses. Prospects loved it, too, because they could instantly see exactly what they would get from an @stake engagement. Although Symantec retired the Business Impact Matrix shortly after acquiring @stake, it proved its worth in hundreds of client engagements.

When creating bivariate exhibits, consider the 2×2 matrix as an alternative method of display. Ask yourself whether the axes can be sliced into quadrants and labeled. If so, you may be on the verge of discovering something innovative yourself!

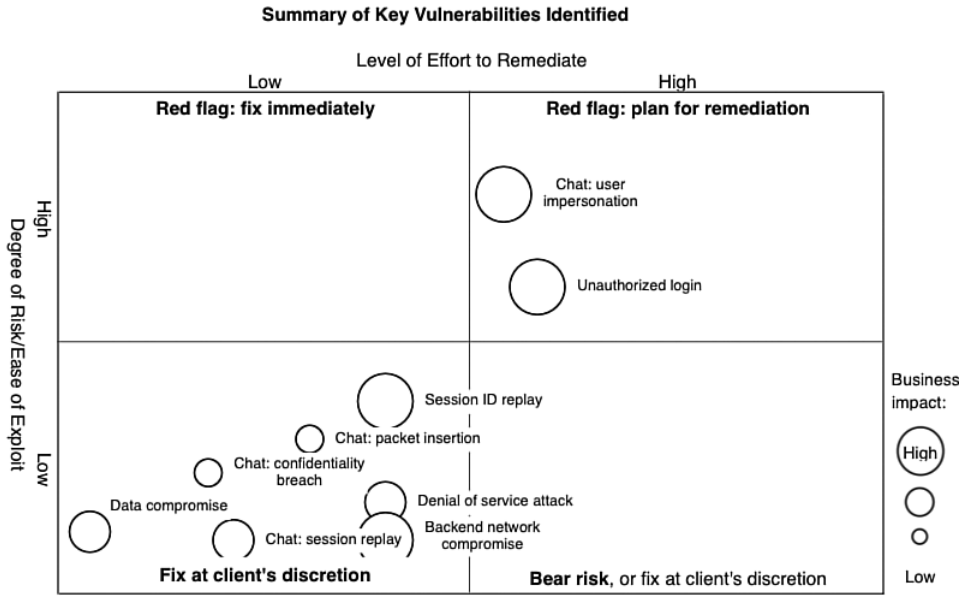


**Figure 6-18** @stake Business Impact Matrix (ca. 2000, Redrawn)

## PERIOD-SHARE CHART

A clever variant of the bivariate scatter plot, the period-share chart, shows how competitor market shares change when measured at two periods in time, typically on a year-over-year basis. I have seen this technique used only rarely, but its properties make it a natural exhibit format for analyzing security data.[11]

[11] The term "period-share chart" is my own designation for this exhibit format. I have not seen it formally named in any business or statistical literature.

To create the chart, the analyst plots each competitor's market shares relative to the leader for both periods, with the previous period on the x-axis and the current period on the y. A position above the diagonal designates the competitor as one who gained share relative to the previous period; below the diagonal, as a share loser. Figure 6-19 shows an exhibit from a Boston Consulting Group study of capital markets.[12]

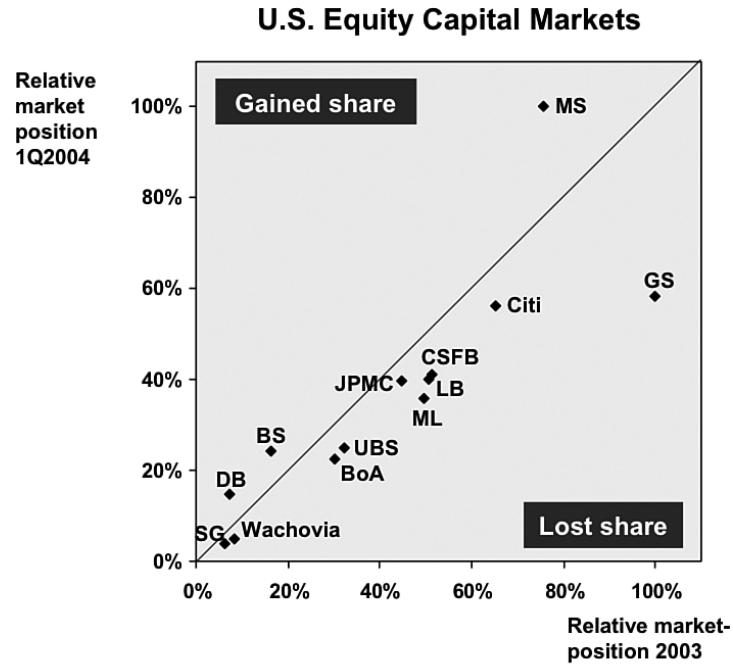


**Figure 6-19** Sample Period-Share Chart (BCG Investment Banking Study)



Examine the chart carefully; its simple appearance belies a sophisticated analytical approach. The use of ratios relative to each period's leader is the key. By using these

[12] S. Ivanov, L. Kuebel-Sorger, B. Rauls, *Investment Banking and Capital Markets: Fourth Quarter 2004 Edition*, The Boston Consulting Group, 2005, http://www.bcg.com/publications/files/Q4_2004_Market_Report_BR_TM_NYC_10Mar05.pdf.

measures as the x- and y-axis values, readers can clearly see the relative period-over-period increase (or decrease). The beauty of this approach is that firms whose positions change relative to the lead firm "automatically" appear above or below the diagonal. The period leader always falls either along the top or right of the chart. If a firm holds the leader position for both periods, it appears in the upper right, at the top of the diagonal line.

Mechanically speaking, spreadsheet software such as Excel can create period-share charts, but only after the analyst applies a little persuasion. A standard X-Y scatter plot supplies the base. To give the chart a bit of extra headroom, I recommend increasing the axis maximums a little past 100% (1.0)—for example, 110% (1.1). Period-share charts should always have an enclosing plot frame. The analyst can use one of two methods to create the diagonal line. The quick-and-dirty way involves overlaying a diagonal line over the chart. However, this means that the chart cannot be resized without also moving the diagonal line and that printed versions of the chart may suffer from line alignment problems. Therefore, I recommend the following technique:

- Create a new data series containing exactly two data points: (0,0) and (110%,110%).
- Plot the new series on the chart.
- Add a trend line for the series (this adds the diagonal).
- Change the line's thickness and color so that it matches the plot frame.
- Hide the markers for the series (leaving just the diagonal).

Shifting the focus away from general use cases for a moment, how might one use a period-share chart to represent security data? One way might be to replace the concept of "market shares" with "vulnerability shares" or "spending shares." In other words, use the period-share format to show how the distribution of security flaws or spending priorities changes over time.

Figure 6-20 shows a concrete security example. On the chart, I have plotted the "vulnerability shares" of flaws found in major security vendors' software packages for the years 2003 and 2004.

Another security example might be a plot of spending priorities year-over-year, or changes in the types of threats detected by an organization's perimeter controls. I leave these as an exercise for the reader.

Period-share charts efficiently show the answer to a basic question: what's hot this year (quarter) compared to last? In that respect, they work as well as—and maybe better than—stacked bar charts or area charts. But period-share charts work less well in certain situations. For example, the period-share chart (in Figure 6-20) says nothing about the absolute number of vulnerabilities found for either the leader (Symantec) or all companies in the data set.

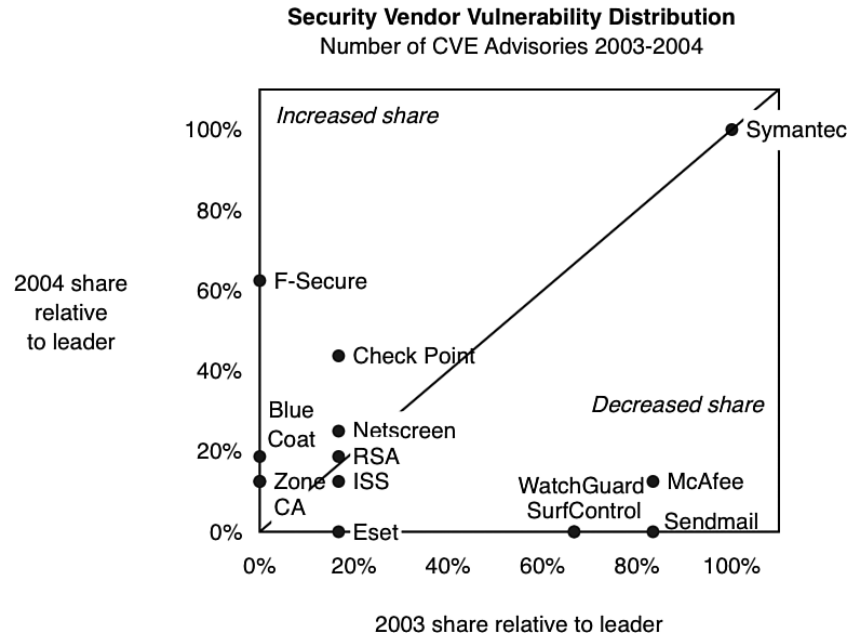


**Figure 6-20** Period-Share Chart (Security Example)

Therefore, as a rule of thumb, avoid period-share charts when

- You need to show absolute share values.
- You need to know the absolute size of the market.
- Data sets are thin, leading to a situation where multiple points "cluster."
- The number of competitors exceeds fifteen or more.
- The desired periods for analysis exceed two.

In many cases, alternative chart formats (such as the stacked bar chart or the Pareto chart) are a better choice.

## PARETO CHARTS

If period-share charts show how relative leadership positions among data categories change over time, Pareto charts help viewers understand each category's contribution to the total. Analysts commonly use Pareto charts to determine whether a small number of categories contribute disproportionately—that is, whether the data implies an "80/20 rule."

The Pareto chart requires a data set aggregated by category and sorted in descending order. Then, for each record in the data set, the analyst calculates the cumulative total as an absolute number and (optionally) as a percentage. Table 6-2 shows the sample security-related data set used in the period-share discussion, enhanced with calculated values for the Pareto chart. I have sorted the 2004 vulnerability counts and added two calculated columns: "Cumulative" and "Cumulative %."

**Table 6-2** Pareto Chart Data Set (Security Example)

| Vendor | 2004 | Cumulative | Cumulative % |
|---|---|---|---|
| Symantec | 32 | 32 | 28.3% |
| McAfee | 14 | 46 | 40.7% |
| F-Secure | 11 | 57 | 50.4% |
| Check Point | 8 | 65 | 57.5% |
| ISS | 8 | 73 | 64.6% |
| Trend Micro | 7 | 80 | 70.8% |
| Zone Labs | 6 | 86 | 76.1% |
| OpenSSL | 5 | 91 | 80.5% |
| Sophos | 5 | 96 | 85.0% |
| Sygate | 5 | 101 | 89.4% |
| Webroot | 4 | 105 | 92.9% |
| Panda | 3 | 108 | 95.6% |
| SSH | 3 | 111 | 98.2% |
| Kaspersky | 2 | 113 | 100.0% |
| IronPort | | 113 | 100.0% |
| Total | 113 | 113 | 100.0% |

Figure 6-21 shows a naive, sample Pareto chart for this data set, created using a standard combination-chart wizard and heavily reformatted. Even though we've followed our graphics guidelines—the chart looks neat enough—clear readability issues emerge.

To begin with, we needed to stretch the chart horizontally quite a bit in order to fit everything in. But because Pareto charts by definition attempt to show 80/20 distributions, we can safely cut the low-scoring items out of the list and save space. Focusing on the top 10 vendors, rather than including all of them, follows the 80/20 rule. Second, turning the chart on its axes helps quite a bit, although it requires some spreadsheet hackery to do.

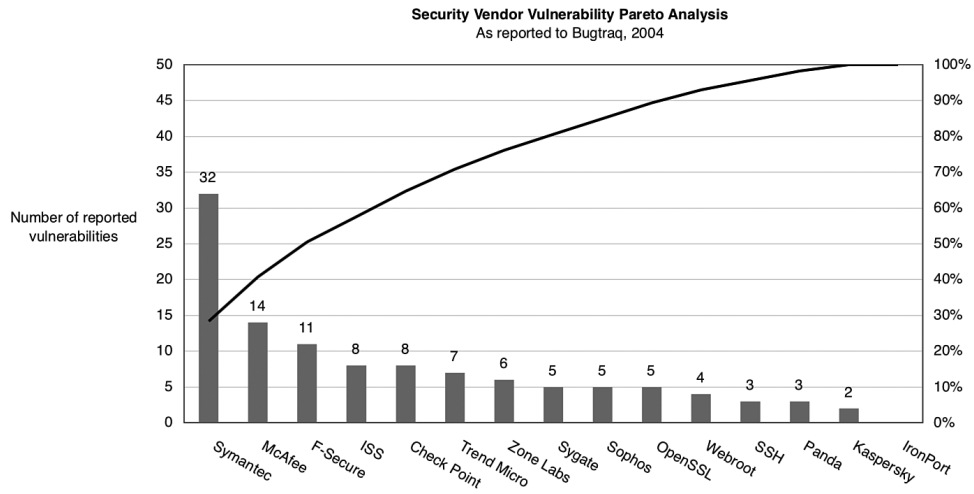


**Figure 6-21** Sample Pareto Chart

Figure 6-22 shows the redrawn version of the Pareto chart. To create the chart, I plotted the two data series (one for the absolute vulnerabilities per vendor, the other for cumulative percentages) initially as two horizontal, overlapping bar graphs. The secondary vertical axis (right side of the chart) contains the cumulative percentage bars. I hid the secondary vertical axis' tick marks and labels. Then, I set the fill color and line for the second series to "no fill," rendering them invisible. Last, I added a polynomial trend line with a polynomial regression with an order of "6." This adds the red "cumulative %" line to the chart. Why do this? Because Excel cannot display a horizontally oriented line chart that uses categories (although it will do so with bar charts, which is how the first data series appears). Unfortunately, until Excel's charting capabilities improve, analysts will need to resort to hacks of this sort.

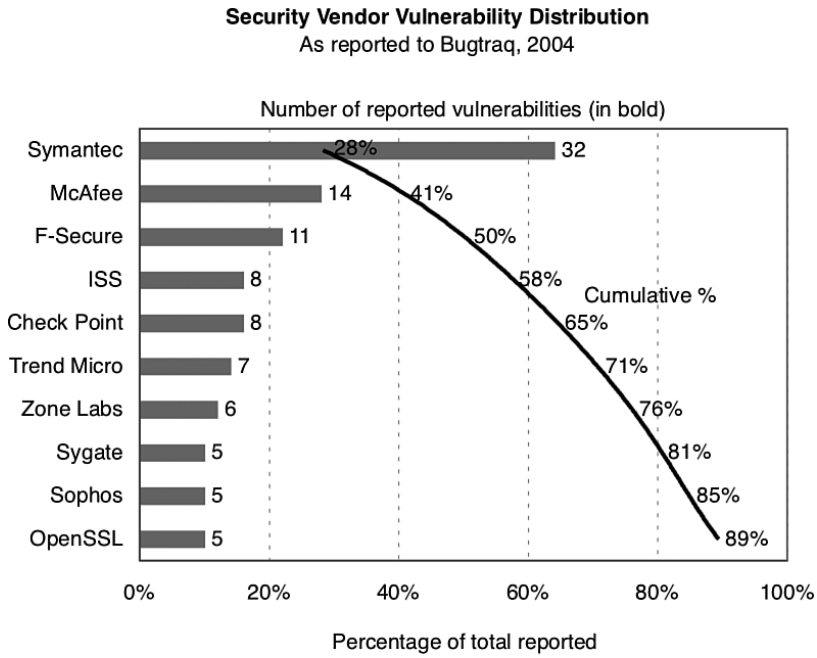


**Figure 6-22** Pareto Chart (Redrawn)

## TABLES

The last technique I will discuss will probably not seem like much of a technique at all: the humble table. Sometimes charts and fancy graphs are overkill. Tables are typically a better choice when

- **The size of the data set is thin**—less than a dozen data points, and spanning a single series.
- **The data set contains many distinct series,** no one of which dominates.
- **The data are imprecise**—designed to provide relative measures rather than precise, empirical absolutes.
- **The idea conveyed in the exhibit cannot be explained by numbers alone**, and depends on additional, textual exposition.

For example, financial report exhibits typically contain many data series. Each line item (for example, EBITDA) is a series unto itself. The audience might want to first look at

EBITDA, then glance upward at the revenue line, and then back down to the GSA expense line. None of these series dominates, and each one needs some explanatory text (the left column) in order to be understood. Therefore, a table is a natural choice for this type of exhibit.

Evaluation matrices are also a classic table application. Figure 6-23 is an example of an exhibit that shows two data series: degree of trust and data sensitivity. Both are on a 1-to-3 scale. Neither set requires particular precision. Note the use of focused use of saturated blue and thin horizontal rules; they make the data "pop" while allowing the reader's eye to sweep sideways for context.

**Network zones: order of implementation**

**Minimize risks from untrusted sources first, then secure sensitive data.**

Implementation order

| Zone | Traffic type | Degree of trust*** | Risks | Data sensitivity | Dependencies |
|---|---|---|---|---|---|
| DMZ* | Many-to-many | ● | Unquantified risks, denial of service | ● | Feeds from outside Framingham |
| Data | Many-to-few | ● ● ● | IP theft, reputation | ● ● ● | Applications, cultural |
| IT Admin | Few-to-few | ● ● ● | Rogue admins, denial of service, eavesdropping | ● ● | Hardware upgrades, cultural |
| Serverland | Many-to-few | ● ● ● | Fraud, IP theft, denial of service | ● ● ● | App-to-app communications |
| Userland** | Many-to-many | ● ● | IP theft, denial of service | ● ● | Cultural, function-specific protections |

**Figure 6-23** Sample Table

As with other types of exhibits, tables should be relatively free of ornament. In the example in Figure 6-23, created in a presentation package (PowerPoint), all the vertical table lines and margins have been erased. This is in contrast with the default PowerPoint layout, which puts solid black lines around each cell and a 2.25-point black line around the perimeter.

Sometimes decoration has its uses, however. Consider the famous *Consumer Reports* circular icons used in product ratings: black circles for "poor," lightly stroked white circles for "average," and red bull's-eyes for "excellent." In addition to being universally understood (always a good thing), the *Consumer Reports* icons retain their meaning even

when reproduced in black and white. This is a subtle but often overlooked property worth emphasizing. Figure 6-24 shows an example done in a similar style, which packs quite a bit of information into a relatively small space.

**DRM Protection Effectiveness**

| Criteria | | Myth TV | Macro Vision | MS MP | OMA | TiVo 2 | RIM | DTV |
|---|---|---|---|---|---|---|---|---|
| Protection against… | Hardware attacks | ● | ● | ● | ◎ | ◎ | ◎ | ○ |
| | Software super-cracks | ● | ◎ | ○ | ●/◎ | ◎ | ◎ | ○ |
| | Cryptographic breaks | ● | ● | ○ | ◎ | ○ | ◎ | ○ |
| Effort for successful attack | | ● | ● | ◎ | ◎ | ◎ | ◎ | ○ |
| Quality of extracted content | | ● | ◎ | ● | ○ | ●/◎ | ○ | ◎ |
| "Water-tightness" of protections | | ● | ● | ● | ● | ● | ● | ● |
| Assessment of controls for: | Entitlements (CAS) | ● | ● | ◎ | ◎ | ○ | ◎ | ○ |
| | Copy control (DRM) | ● | ● | ◎ | ◎ | ◎ | ● | ◎ |

**Key message**: Hardware-based protections present the highest barriers, and can make conditional access systems very effective. However, even the best protections will not prevent *some form* of medium- or high-quality content leakage, after access has been granted.

**Symbol key: low** ● medium ◎ high ○ (Note: Quality of Extracted Content symbols reversed; **h** ● m ◎ l ○)

**Figure 6-24** Sample Decorative Table

This exhibit works well for a number of reasons. First, the icons are much easier to read than a sea of numbers, which are not easily distinguished. In addition, the saturated colored circles make the relatively "good" ratings really stand out, which in this case is what we want. Notice the subdued gray grid in the background and the thicker line separating the totals rows from the main table body. Imagine what this would have looked like if it had been done using default settings—that is, with a heavy grid, and with numbers instead of icons. Sometimes, it pays to spend a little extra time redrawing the table.

## TREEMAPS

The visualization methods I have described in this chapter show how to effectively display data over time, in cross sections, and using systems of one or more variables.

The vast majority of the security metrics discussed in the preceding chapters lend themselves well to these methods. However, all these methods assume a data set whose records are structured in a relatively simple manner. For example, department *X* has value *Y*, or activity *A* has value *B* at time *T*. We assume that the independent variables—departments or activities—iterate through flat lists of values.

But what if the data set is not flat? Metrics visualization for security often requires the ability to roll up, or drill down into, a data set. In these cases, containment and hierarchy relationships establish vital context for the viewer. Perhaps the best way to view the data is to show the hierarchy as part of the exhibit. For example, one could show the roll-up structure for departments, sites, and business units, or the containment relationships of business processes.

"Hmm," says the IT analyst as he strokes his goatee. "Graphical displays of hierarchy… isn't this what network diagrams do?" Yes, in part. Technology architects have been drawing network diagrams for an eternity, and these show containment relationships quite well. However, network diagrams are rarely suitable for metrics visualization because they are

- **Too technical:** Managers don't care about TCP/IP addresses.
- **Too literal:** Only a small number of security metrics make sense on network diagrams.
- **Space-inefficient:** Lots of white space, low density of nodes per inch.

Fortunately, recent innovations in data visualization outside the information technology field mean that security analysts need not rely on network diagrams to show containment-oriented data sets. There is a better alternative: the *treemap*.

Little known outside of academe, treemaps are used with hierarchical data structures that can be aggregated. The core data elements of a treemap are rectangular nodes that, when rendered, appear as a patchwork of rectangles. The arrangement of the rectangles shows the containment hierarchy, in the same way a Bento box does. The size (area) of each rectangle represents the node's "weight," while its color or brightness displays attributes such as relative importance, criticality, or membership within an arbitrary category. Treemaps possess four properties that make them extremely useful for large-scale data visualization:

- Simple visual paradigm
- Extremely space-efficient
- Naturally suited for aggregation
- Excellent for high-resolution data display

Originally developed by Ben Schneiderman, a professor in the University of Maryland's Department of Computer Science, treemaps are easily the most innovative data visualization technique to emerge from the research world in the last ten years. Although they are not yet mainstream, many companies have created compelling, rich information graphics with them. For example, SmartMoney.com's Java-based Map of the Market applet, shown in Figure 6-25, features a treemap that shows near-real-time stock activity. The size of each block represents the relative market capitalization of the sector or company; the color shows whether prices are increasing (green, rendered here as light grays) or decreasing (red, rendered as darker grays). What is particularly clever about this example is that it precisely illustrates the micro/macro visualization qualities of treemaps. The reader sees the overall sweep and scope of the market, and he or she also sees how the different blocks relate to each other—and can dive into one of the individual data points, too.

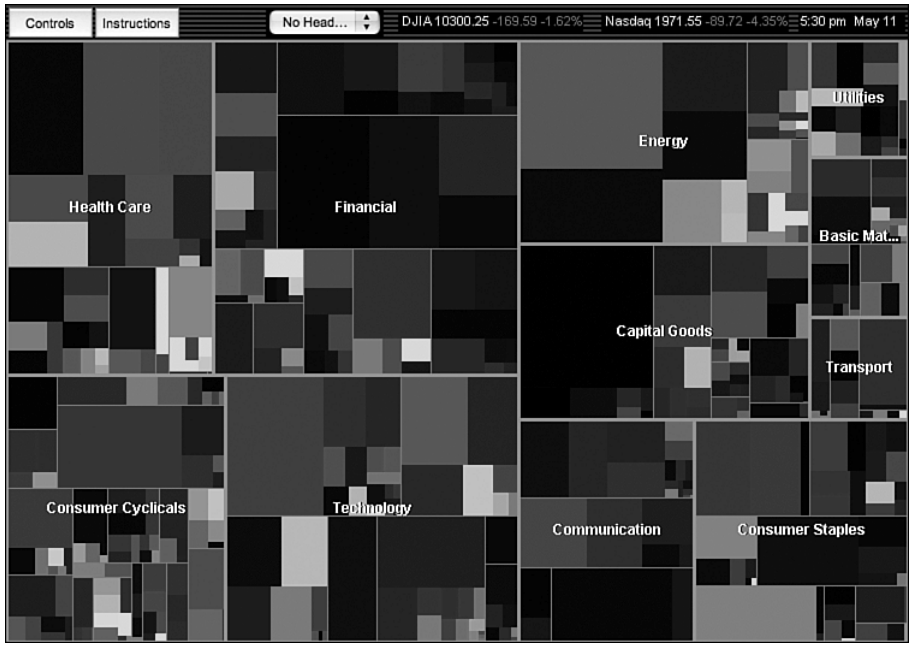


**Figure 6-25** SmartMoney.com Map of the Market

## CREATING TREEMAPS

Standard office productivity suites cannot create treemaps; instead, analysts must rely on specialized toolkits. Many treemap packages exist, including an open-source implementation I wrote for my own use called JTreemap. Let's walk through a simple treemap example using this tool, available on my website at http://www.freshcookies.org/jtreemap.

To construct a treemap, the security analyst identifies data attributes that supply:

- The size of each rectangle (size of deployed base, dollar value of asset, number of lines of code)
- The saturation value for each rectangle (criticality, priority, business impact)

and optionally

- The containment hierarchy (top-level category, department, business unit)

Next, the analyst formats a data set, loads it into JTreemap, and plots the results. JTreemap accepts a tab-delimited input file; after parsing the input, it creates a graph of the treemap. Table 6-3 shows a sample input file containing action plan data for an assessment of an e-commerce application. The field order for the file is as follows:

- NAME: Displayed as text within the node. Here, we'll use the name of the action item.
- DESCRIPTION: Displayed in the tool tip when the mouse pointer hovers over the node.
- BRIGHTNESS: The node's relative saturation, ranging from 0.0 (transparent) to 1.0 (fully saturated). In this example, we'll supply the item's priority, with the highest values representing the most important items.
- AREA: The amount of space to allocate to the node, relative to all others in the treemap. For the area, we will specify the amount of effort required to implement the action item.
- CATEGORIES[0..*n*]: The categories to use for this node (separated by tabs), with the highest-level categories first. An arbitrary number of categories may be specified, although in practice most simple applications will not need more than three or four. Each top-level category will be given its own color; in this example, there is only one top-level category. For this example, we will simply supply the name of the responsible business group ("E-commerce security").

**Table 6-3** Sample Treemap Data File

| Name (Action Item) | Description | Brightness (Priority) | Area (Effort) | Categories (Application Name) |
|---|---|---|---|---|
| Password policy | For end users | 0.9 | 4 | E-commerce security |
| Secure coding practices | For developers | 1 | 8 | E-commerce security |
| Identity management | Centralized account management | 0.6 | 12 | E-commerce security |
| Website server configuration | To be done by the systems group | 0.7 | 5 | E-commerce security |

To ensure that nodes are arranged sensibly and in a manner pleasing to the eye, treemaps typically support one or more *layout algorithms*. The first algorithm, originally developed by the University of Maryland, is the "strip" layout. However, at present prevailing consensus holds that J.J. van Wijk's "squarified" layout algorithm[13] provides the best balance of structural fidelity and aesthetics. This is the one I use in my own package.

A single command from the console produces an interactive dialog box containing the treemap:

```
java -jar freshcookies-treemap-0.3.jar test.tab
```

Figure 6-26 shows the resulting JTreemap dialog for our sample data set.

The preceding example, while simplistic, shows how treemaps work. The "Identity management" rectangle dominates because it requires the most effort to fix; the saturated color of "Secure coding practices" (rendered as a lighter gray) shows that it is the most important priority. Since the business group "eCommerce security" will fix every action item, all items receive the same color (red, rendered as gray here).

Treemaps can support much higher data densities than in our simple example. Figure 6-27 shows action items mapped to the ISO 17799 security framework. In contrast to the previous example, which contained only one level of containment (the group name, eCommerce security), this example contains three. These levels correspond to the first three levels of the ISO topic hierarchy. Each rectangle is equally weighted (all have areas of 1) but contains different saturation values. In all, Figure 6-27 displays 556 data attributes (139 topics times 4 attributes: area, saturation, top-level grouping, and name).

[13] See Van Wijk's explanation at his website, http://www.win.tue.nl/~vanwijk/rhd.pdf. The visual style of JTreemap follows the one used by Marcos Weskamp's stunning NewsMap application (http://www.marumushi.com/apps/newsmap/newsmap.cfm).


**Figure 6-26** Sample Treemap


**Figure 6-27** ISO 17799 Treemap (Displaying Three Levels of Hierarchy)

Figure 6-28 shows the same data again, but aggregated so that the lowest level "rolls up" to the top two.

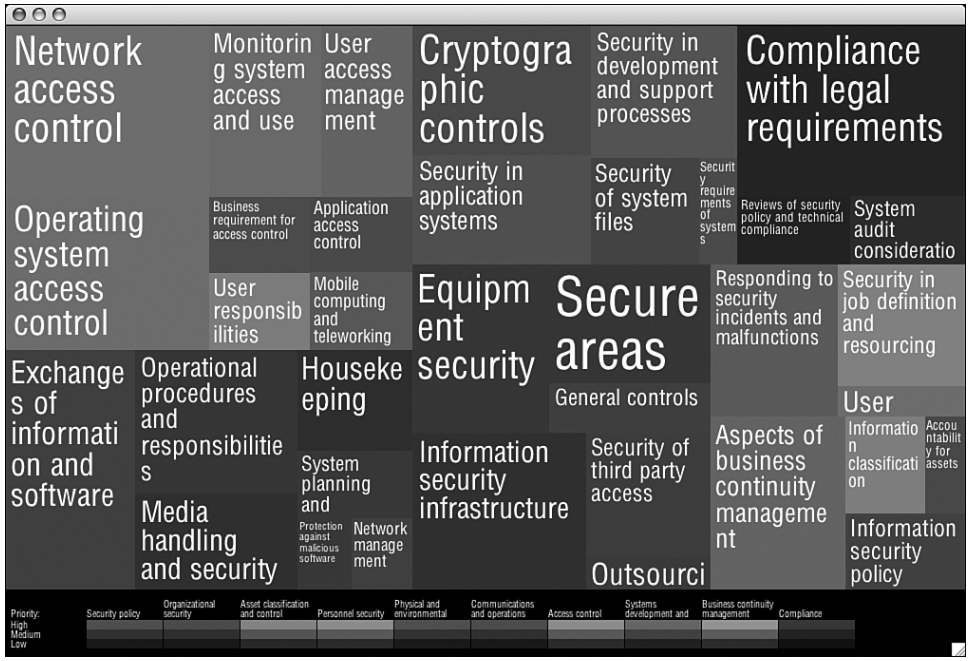


**Figure 6-28** ISO 17799 Treemap (Displaying Two Levels of Hierarchy)

Treemap styles often vary from the one I have presented here, which is my own implementation. Some include text in individual nodes; others do not. Other implementations feature clever shading or border-rendering algorithms, drilldown capabilities, and more. The University of Maryland's treemap website (http://www.cs.umd.edu/hcil/treemap-history) contains links to other implementations, including a wide variety of commercial packages.

In summary, treemaps add another tool to the security analyst's bag of tricks. Treemaps provide an effective way of visualizing highly dense, hierarchically structured data. Although treemaps are not yet implemented in commercial office productivity packages, implementations exist that can help you today. Get to know them, and watch your colleagues' jaws drop.

## THINKING LIKE A CANNIBAL: THE CASE FOR REDRAWING

Good visualization flows naturally from good design. And good design results from clear thinking about desired objectives; the format should always strive to answer key questions. Analysts should, on a regular basis, revisit existing exhibits to determine whether the questions they answer are the right ones. Equally important, analysts should ask themselves whether the design, format, and details of existing reports answer those questions as well as they could. When an exhibit falls short, consider revisioning and redrawing it.

Redrawing an existing exhibit format requires a certain amount of courage, particularly when an existing process or shipping product depends on it. Let us try an example or two.

### A PATCH JOB FOR ECORA

Ecora Patch Manager's Reporting Center produces a patch-management status chart[14] and associated table that summarizes the effectiveness of the patch application process (see Figure 6-29). The chart shows the number of patches available, plus the total number available, and groups these statistics according to the severity of the patch. Unfortunately, the chart does not tell us much other than aggregate statistics, and the exact question it answers seems vague. Questions this exhibit should answer include these:

- What patches are the most troublesome (hardest to apply)?
- How effective or efficient is the patch management solution overall?
- How large is the window of exposure for systems that have not yet applied patches?

We can tell that the chart is not doing its job well just by looking at it. The dead giveaway is that two bars use the same unit of measure (one for total number of patches available, and one for installed patches). Bar charts are typically good candidates for redrawing using a bivariate chart.

Redrawing this chart using two variables improves matters somewhat. Let us assume that effectiveness, or lack thereof, matters most. Therefore, the chart ought not to emphasize applied patches, but *missing* patches—in particular, the percentage of missing patches. Assume for the moment that the number of overall required patches matters to the reader, too, because it implies the overall effort required to apply them. Therefore, the two axes show the number of available patches (on the x-axis) and the percentage of these that were not installed (on the y).

[14] Ecora Patch Manager, Installed Patch by Severity Report, http://www.ecora.com/ecora/sample_reports/patchmanager/installedPatch.asp.

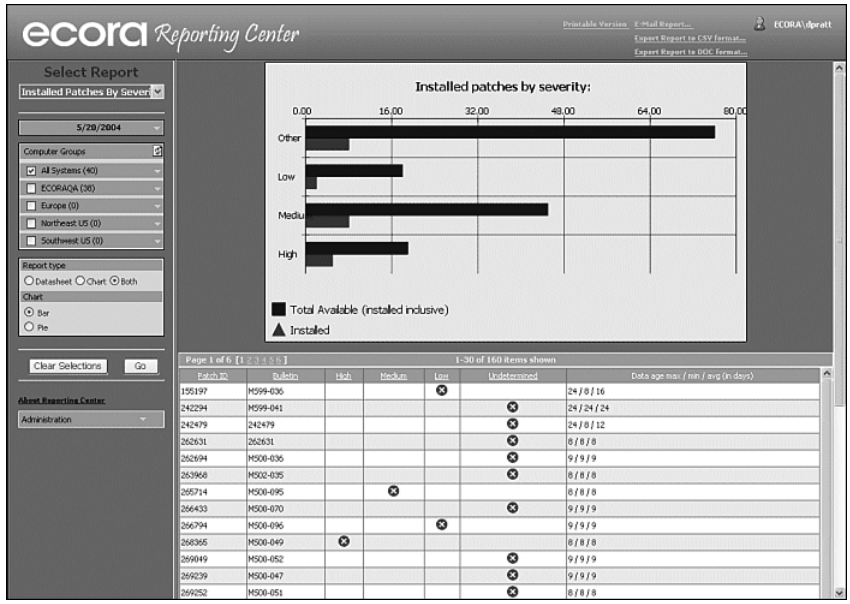


**Figure 6-29** Redrawing Candidate: Ecora Patch Manager



An initial redrawing of the exhibit results in the example shown in Figure 6-30. Notice the change in terminology from "available" patches to "required," which better reflects the mandatory nature of patch management; we *want* available patches to be installed, don't we?

Although this bivariate chart improves on the original, it still presents problems: does the reader care about the number of required patches in aggregate? Probably not. The chart becomes more relevant when expressing the number of required patches on a per-machine basis. For the sake of simplicity, assume that the number of affected machines in the sample is five. Figure 6-31 shows the second revision for the chart.

The new chart improves on the two previous iterations, but we are not done yet. Examining the data presented in the table below the original Ecora graph, we notice additional data that might make the exhibit even more relevant: per-patch performance data, expressed in terms of the minimum, maximum, and average days required to apply each patch. A stock-chart style exhibit provides an ideal way to express this concept; a secondary table underneath provides the per-machine missing patch statistics.

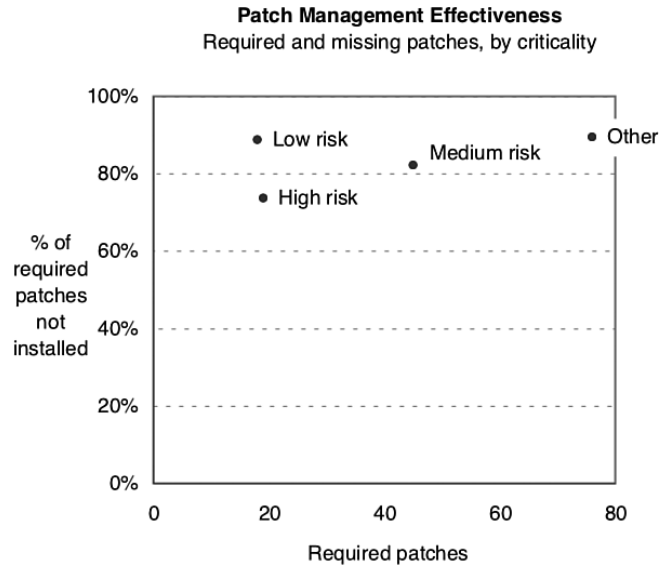


**Figure 6-30** Ecora Patch Manager Exhibit (Redrawn)

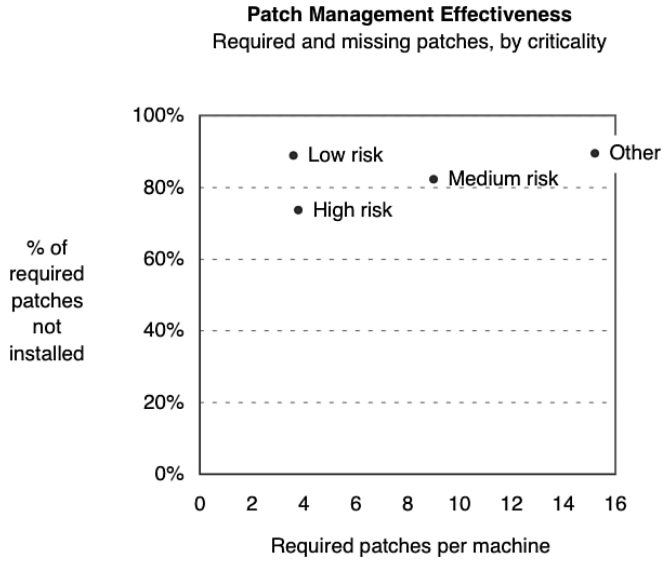


**Figure 6-31** Ecora Patch Manager Exhibit (Second Redrawing)

Figure 6-32 contains the final redrawn patch management exhibit (using hypothetical min/max/average data, since we don't have the actual figures). The crossbars for the average patch time appear as opaque white-colored crossbars, which have the effect of "erasing" part of the hi-low bars.

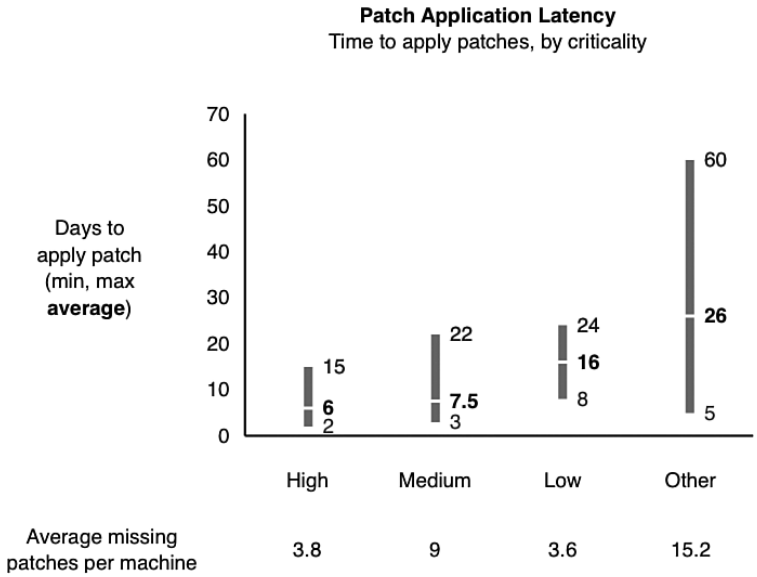


**Figure 6-32** Ecora Patch Manager Exhibit (Final Redrawing)

All in all, the final exhibit answers our key questions much better than the original. It shows the average, minimum, and maximum patch latency metrics for four classes of patches, as well as the average number of missing patches per machine in a mini-table below the chart. The revised chart contains twice as many data points as the original, and the data are more revealing. The vertical axes show mean "per patch" and "per machine" performance statistics instead of simpleminded, non-normalized raw numbers. And aesthetically speaking, the new chart's relatively unadorned, functional format focuses the viewer's eye on the data.

The revision process revealed four lessons. Analysts should always:

- **Question the exhibit format** when the complexity of the underlying message exceeds the chart's ability to communicate it faithfully.
- **Dig deeper** for richer, more relevant data to answer key questions.

- **Consider nontraditional formats,** such as our "stock chart" adaptation.
- **Use iterative revisions** to zero in on the right design for the exhibit.

## REORIENTING SECURCOMPASS

It seems safe to assume that no security analyst would intentionally create a bad graphic. But sometimes reality challenges even the most cherished assumptions. Consider the "benchmarking" chart shown in Figure 6-33, created using a security assessment tool from Espiria called SecurCompass®. The exhibit, which appeared in the *Computer Security Journal*, shows the average score (and ranges) by industry for security compliance.[15] But you would never know it—the chart is literally incomprehensible:

- **Nondescript title:** The title "Industry Benchmarks" sounds terrific, but what is the "benchmark," and how was it derived?
- **Poor series labeling:** The words "low" and "high" suggest relative measures—but relative to what?
- **Poor axis labeling:** What unit of measure does the vertical axis represent?
- **Mysterious methodology:** From whom was the data obtained, and over which periods?
- **Obnoxious formatting:** Readers must crane their necks sideways to read the category labels, and the data series bars (formatted in 3-D, naturally) seem far too narrow.

This exhibit is a real howler. Ironically, the authors of the exhibit actually have interesting data to present. For example, in the narrative support immediately preceding the exhibit, the reader learns that the "low" and "high" data points refer to the minimum and maximum scores observed in each industry. Four pages prior to that, the reader learns that the numeric scale (0 to 5) refers to scores on a self-assessed security compliance survey. Most significant, the first page of the report—thirteen pages before the exhibit—implies that the sample size for the study exceeded 350 organizations. All these details matter, and should have appeared in the exhibit. It is almost as if the authors were ashamed of their good (?) work, and wanted to bury the evidence.

[15] J. Heimerl and H. Voigt, "Measurement: The Foundation of Security Program Design and Management," *Computer Security Journal*, Volume XXI, Number 2, 2005.

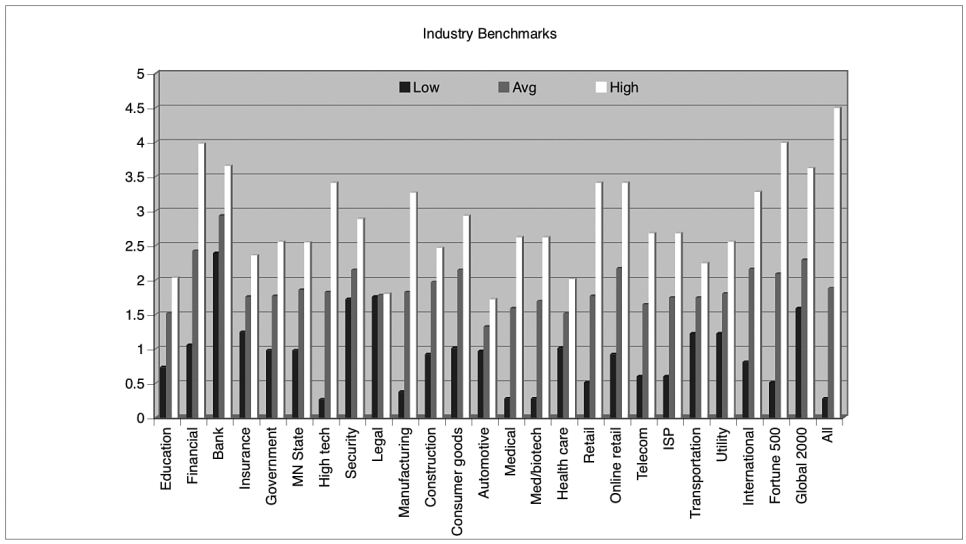


**Figure 6-33** Redrawing Candidate: SecurCompass



Cleaning up the SecurCompass exhibit requires a change in format. A "hi-lo-close" stock chart gives us an effective way to show the minimum/maximum/average data points. Changing the format to vertical permits the viewer to read all the industry labels. A few additional chart tweaks help, too:

- More honest chart title (these are scores for self-assessment, not benchmarks)
- Inclusion of sample size (350 companies) and sample interval (2001 to 2005)
- A clearer label for the dependent axis ("mean compliance score and range," plus some minimal examples)
- Data sorted by mean industry score
- Softened grid lines

Figure 6-34 shows the redrawn exhibit. Look at the difference. The headline pops right out. Banks and financial services companies score highest, health care and automotive the lowest. The data also suggest that some industry scores vary more than others; compare insurance, for example, with high tech.

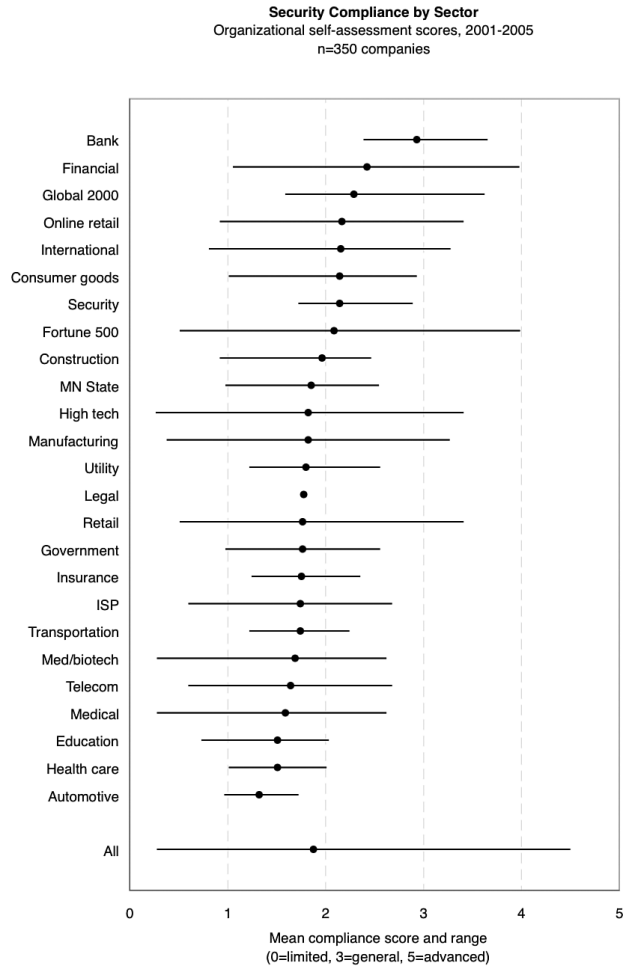


**Figure 6-34** SecurCompass Exhibit (Redrawn)

Although redrawing the exhibit provides much more immediacy and impact, it also highlights potential shortcomings in the data. For instance, why do the legal scores vary so little? (Likely answer: a very small sample.) In addition, the total number of companies in the study (more than 350) seems low compared to the number of industries

(25, which equals a mean 14 samples per industry). It would have been better to condense the industries, perhaps by half. Increased sample density would have made it possible to include additional statistical information like standard deviations or quartile ranges—crucial for understanding variances within industries.

Finally, readers will note that I have changed the title to de-emphasize benchmarking. There is a simple reason for this. Benchmarks imply point-in-time measures, but the sample interval covers a relatively long period of time (four years). Security practices change quickly; scores averaged over a four-year period cannot serve as credible "benchmarks"—although one-year averages can. The alternative title I supplied ("Organizational self-assessment scores, 2001-2005) is more honest and sidesteps the benchmarking issue.

The SecurCompass example teaches several lessons:

- **Charts should explain themselves.** If the reader can't figure out a chart without reading the surrounding narrative, it is a bad chart.
- **Not all data points require labels.** For charts whose primary function is to compare cross sections, simply sorting the data works better than labeling every point.
- **Good charts never "bury the lead."** If the interesting data from the chart aren't intuitively obvious, redraw the chart.

## MANAGING THREATS TO READABILITY

The final candidate for vivisection in this chapter is Symantec's DeepSight Threat Management System (TMS). The service aggregates and summarizes a wealth of information about Internet-based threats gathered from distributed sensors across the globe. One of the tabs on the management console labeled "Firewall Statistics" shows the most popular network ports subjected to hostile probing. Figure 6-35 shows a sample screen capture taken during the spring of 2005. Nice-looking graphics—but wait! Where is the legend that tells us what the colors on the horizontal bars mean?

Oh, there it is—at the very bottom of the web page, accessible only after scrolling past several completely unrelated tables. Although it is not a fatal flaw, it certainly detracts from readability, because it requires readers to move their heads back and forth between the exhibit (at the top) and the legend (at the bottom).

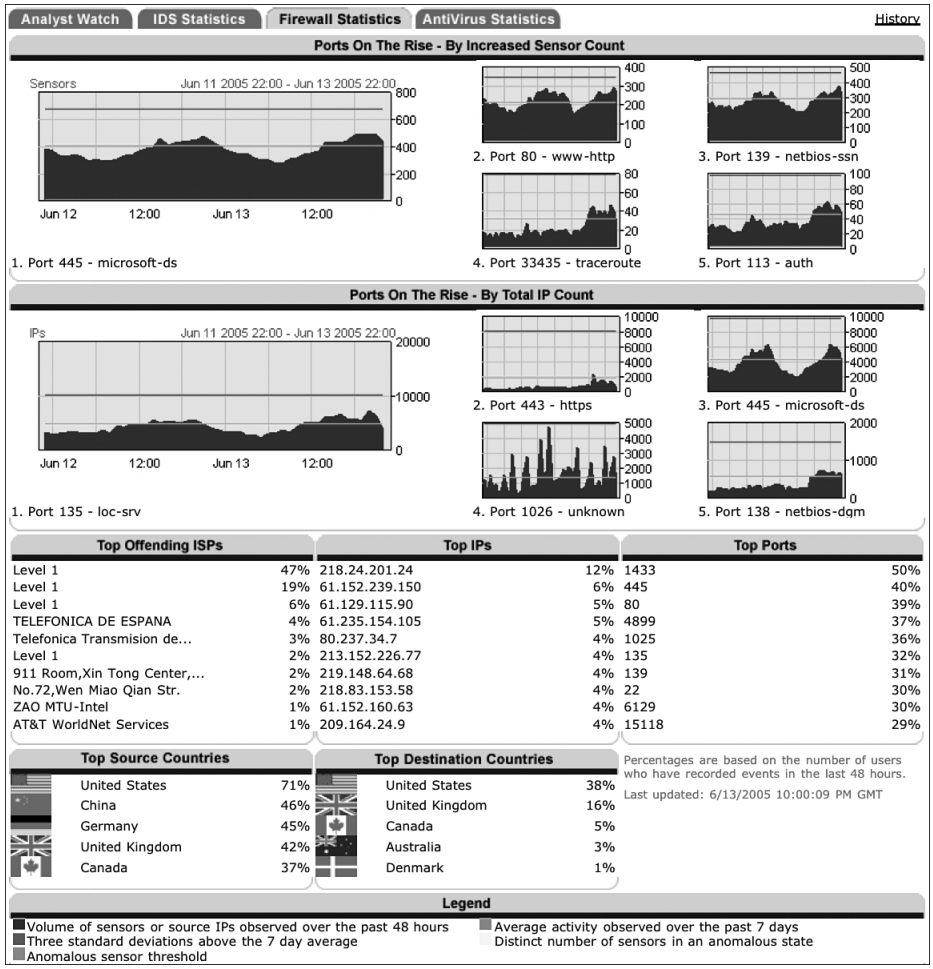


**Figure 6-35** Symantec Threat Management System



So what is wrong with this exhibit? Nothing fatal; most readers will find the exhibit—which appears to be a variant of a small multiple—to be perfectly adequate. But upon closer inspection, certain qualities of this exhibit seem odd:

- Why the extra prominence given to the leftmost graphics? Do they summarize the four (times two) small multiples to the right? (Answer: The leftmost graphics are not summaries. They are larger merely because they show the number-one ranked ports.)

- Corollary question: does the extra width of the leftmost graphic signify anything important? (Answer: No.)
- Why don't the legend keys seem to make much sense? What does "Volume of sensors or source IPs observed . . ." mean? (Answer: It is poorly worded.)
- Some of the graphs on the right side of the ". . . By Increased Sensor Count" exhibit, upon first glance, show volumes that look higher than the leftmost graphs. Is this accurate? (Answer: No. The vertical scales are different, which distorts the visual impression. The scale for the graph of the #1 port is a ridiculous eight times bigger than #5.)
- The red horizontal lines indicate port probe volumes three standard deviations above the seven-day trailing average. Is that an appropriate threshold? (Answer: Probably not. Recall from Chapter 5 that for normal distributions, data points within one standard deviation ($\sigma$) on either side of the mean covers two-thirds of the points. Two standard deviations covers 95%, and three covers 99.7%. The TMS data might be distributed normally; even if it is not, three standard deviations seems excessive.)

Based on the concerns I have expressed regarding format, scaling, and labeling, the reader can probably guess how I chose to fix the exhibit. Figure 6-36 shows a redrawn version that includes these changes:

- The leftmost graphics' size and scale match those on the right.
- All the small multiples' baselines align.
- Vertical axes maintain consistent scales.
- Grid lines vanish.
- The number of standard deviations in the "threshold" line (red) decreases from three to two, and its graphical representation changes to a gray band extending on *both* sides of the mean.
- Plain-English service names replace the cryptic, IANA-style service names used in the original.
- Chart legend labels vanish; the one relevant legend item (regarding the threshold line) morphs into annotations applied directly to the leftmost multiples.

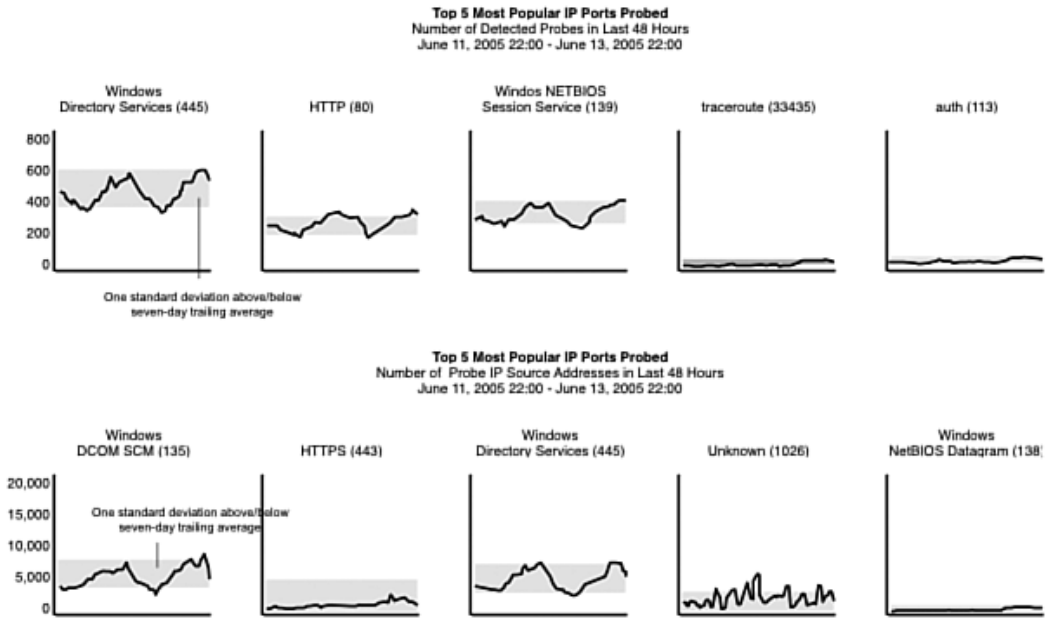


**Figure 6-36** Symantec Threat Management System (Redrawn)

The redrawn version communicates the facts more truthfully. Due to the consistent scaling for the y-axis, the reader immediately notices the Pareto-like properties of both the number of detected probes and source IP addresses: after the third port, the numbers drop off sharply. The rescaled version also removes an inadvertent distortion. Instead of an epidemic of `traceroute` attacks (top row, #4), one sees that the number of detected probes for this port is in fact extremely low relative to the top three.

In addition, the combination of the tightened "threshold line" from three standard deviations to one, and its corresponding extension to below the mean, yields an interesting insight. Most of the time, the levels of probe traffic seem to fall within fairly predictable bands.

The redrawn exhibit could arguably benefit from further improvements. To keep things simple, I omitted time markers on the horizontal axes. Were this exhibit used for forensics rather than (as I imagined) a heads-up dashboard, it would require x-axis markers. And form factor of the exhibit may seem excessively horizontal.

Symantec's Threat Management System provides valuable information, but its slapdash execution makes it less effective than it could be. Its mistakes teach us several lessons:

- To minimize distortions, small multiples should maintain consistent scales for the x- and y-axes.
- Overly wide anomaly thresholds (more than two standard deviations) may not provide enough visual cues to allow viewers to spot unusual behavior. Narrower bands (one or two standard deviations) may provide more value.
- Chart legends should stay as close to the point of use (the charts!) as possible. If possible, consider eliminating the legend in favor of on-chart annotations.
- Superfluous data merely adds clutter. Remove an extra data series not directly related to the main point of the exhibit.
- Small-multiple titles work better when expressed in plain English.

## SUMMARY

If a picture is worth ten thousand words, it follows that an ugly picture is worth ten thousand ugly words. With information security graphics, clarity, taste, and restraint can help ensure that an analyst's graphically conveyed *magnum opus* beautifully expresses the story he or she intended.

You can keep your information graphics lean, trim, and elegant by following six basic principles:

- **It is about the data, not the design.** Resist urges to add shiny backgrounds and decoration, or anything else that detracts from the data.
- **Just say no to 3-D.** Fake depth distracts the reader. Unless you are a NASA scientist trying to visualize global warming, you do not need it.
- **Do not go off to meet the wizard.** If using Excel, prepare for radical surgery after clicking Done.
- **Erase, erase, erase.** Get rid of all grids, tick marks, shadows, and superfluous plot frames. Not all data points require labels. For cross-sectional charts, sorting the data works better than labeling every point.
- **Reconsider Technicolor.** Mute the colors, or use a monochrome palette.
- **Label honestly and without contortions.** Pick a meaningful title that summarizes the exhibit, label units of measure clearly, use consistent fonts of the same size, cite the data source, and avoid abbreviations. Chart legends should stay as close to the data as possible; consider eliminating them in favor of on-chart annotations.

Good charts never bury the lead. If the interesting data from the chart are not intuitively obvious, redraw the chart. If the reader cannot figure out a chart without reading the surrounding narrative, it is a bad chart.

The analyst's graphical toolbag includes a wide variety of exhibit formats, each of which has strengths and weaknesses, depending on the nature of the data and the intended message. These formats include:

- **Stacked bar charts,** which show the contribution of each data series over multiple time periods to an absolute total. Stacked bar charts can also be "normalized" to show each series' relative contribution on a percentage basis.
- **Waterfall charts,** which show how multiple categories accumulate to form an overall total, generally for a single period. Waterfall charts are not especially dense but can make for effective management presentation formats because of their association with consulting.
- **Time series charts,** which show how one or more series vary over a given time interval: hours, months, quarters, or years.
- **Indexed time series charts,** which express each data point as a multiple of its starting value. Typically, the starting points are normalized to a value of 100. Indexed time charts work well for analyzing relative, rather than absolute, performance over time for a group of comparable series.
- **Quartile time series charts,** which plot quartile values for a data series over time. Typically, quartile charts plot three series: the median values, the values separating the first and second quartiles, and the values separating the third from the fourth.
- **Bivariate charts,** which show how two variables behave relative to one another. These charts can help analysts understand relationships between pairs of variables, such as potential cause-and-effect relationships. A variation on the bivariate chart, the **two-period bivariate chart,** resembles a basketball chalkboard diagram and helps viewers understand period-to-period changes in relationships.
- **Small multiples,** which plot several identical charts on the same canvas, allowing the eye to quickly sweep back and forth across the exhibit, looking for patterns, similarities, and differences. The axis scales remain constant, but the cross sections change from chart to chart. Small multiples are one of the most powerful ways to visualize cross-sectional data.
- **Quartile-plot small multiples,** which combine the comparative power of small multiples with the insights of quartile analyses. Particularly popular in management consulting circles, this chart format visually isolates factors that separate the best and worst performers.

- **Two-by-two matrices,** which extend the bivariate plot by grouping results into quadrants. Another favorite of management consultants, the 2×2 matrix enables an analyst to frame the terms of debate by categorizing and naming the results sets: for example, "quick hits," "strategic initiatives," "discretionary fix," and "bear risk."
- **Period-share charts,** which plot winners and losers over two successive periods in a square plot. Winners who increase share appear above the diagonal; losers fall below it. Period-share charts work best when the number of participants does not exceed fifteen and where plot positions are dispersed.
- **Pareto charts,** which present as a bar graph a range of sorted values from largest to smallest. On a secondary axis, a line plot shows how the cumulative addition of values converges on 100%. Pareto charts help analysts understand whether a data set follows the 80/20 rule.
- **Tables,** which show data values in a familiar grid layout. Small splashes of color and careful use of icons, such as those familiar to readers of *Consumer Reports*, can enhance table readability.
- **Treemaps,** which show hierarchical relationships in data sets as a series of recursive rectangles. The relative size or percentage of each data point determines the rectangle's size. Importance or criticality determines the rectangle's color saturation; "hot" items appear more saturated.

With all these exhibit formats to choose from, analysts may sometimes find that choosing the right format is not always easy. Analysts should always question the exhibit format when the complexity of the underlying message exceeds the chart's ability to communicate it faithfully. Dig deeper for richer, more relevant data to answer key questions, and use iterative revisions to zero in on the right design for the exhibit.

In the last few chapters, we have discussed what metrics to get ("Diagnosing Problems and Measuring Technical Security," "Measuring Program Effectiveness"), what to do with them once we've got them ("Analysis Techniques"), and how to show them off to their best effect (this chapter). But so far, we have furiously waved our hands over the "getting" part.

I shall wave my hands no longer. Next up is Chapter 7, "Automating Metrics Calculations," which shows you how to obtain and transform raw data from sources such as firewalls, antivirus logs, and third-party reports.