Security Visualization:

What you don't see can hurt you

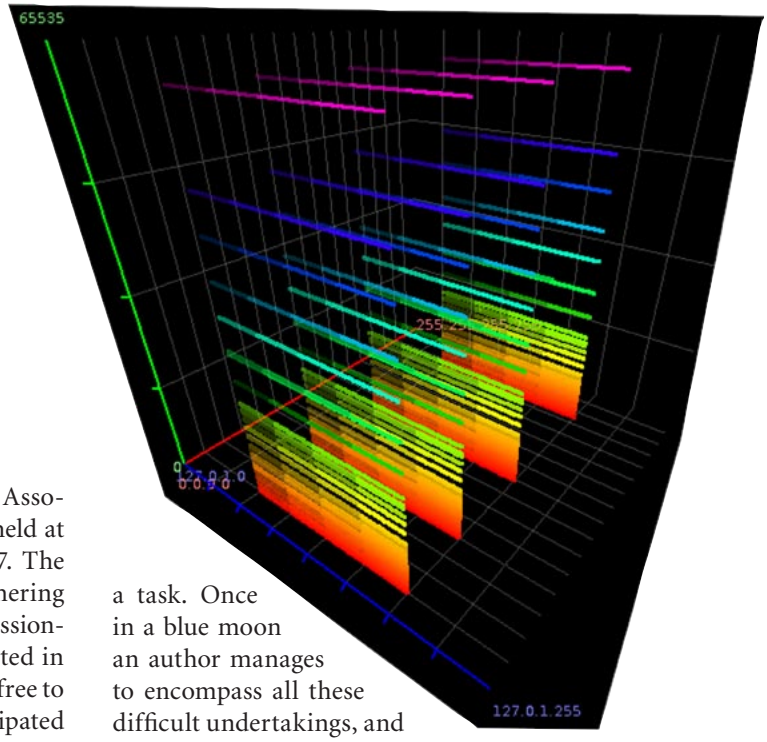
By Russ McRee – ISSA member, Puget Sound (Seattle), WA, USA chapter

Prerequisites

Java for TNV 0.3.8
libpcap or winpcap
Windows XP is best for InetVis if running on Windows

Similar Projects

AfterGlow[1]
Google Visualization API Gadget Gallery[2]
Treemap[3]



I was privileged to attend a Cyber Conflict Studies Association (CCSA) workshop on Response Options, held at the University of Washington this past March 27. The workshop served as the pre-event to the quarterly gathering of the Agora, a northwest regional association of professionals working in the cybersecurity field. If you're interested in learning more about either the CCSA or the Agora, feel free to email me or check out the CCSA.[4] The section I participated in described "potential intrusion and attack sensor options for the private sector." As discussion ensued, it occurred to me that, as attacks evolve, it will be essential to look beyond the norms of intrusion/extrusion detection and deterrence. "Brilliant, Russ," you say to yourself, shaking your head at a statement of most obvious proportions. Bear with me. We know that in most environments, time and resources necessary to truly absorb and analyze log data is challenging at the least.

To these challenges, may I suggest that you add security data visualization tools to your process. Greg Conti, in his groundbreaking gem, *Security Data Visualization: Graphical Techniques for Network Analysts*, sums it up eloquently on the back cover. "A picture is worth a thousand packets."

Allow me to give you a quick review of the book. Many a nonfiction author sets out with the ambitious goal of conveying his topic with clarity, precision, the appropriate amount of detail, and a voice that keeps the reader engaged. Doing so, while negotiating technical topics, is all the more daunting a task. Once in a blue moon an author manages to encompass all these difficult undertakings, and not only meets the challenge but exceeds it. Dr. Conti has achieved just such success with *Security Data Visualization*. Anyone with the slightest drive to "see" information security with different eyes will find this book impossible to put down. Additionally, it's an exceptionally well-crafted book, simply beautiful in its quality of paper and print. You won't regret visiting your favorite book venue for this one. I asked Dr. Conti what we might expect of him in the future:

"I'm currently researching the application of visualization to assist file-level analysis. While there are special case analysis tools for executables, a great deal of work still occurs using the venerable hex editor. Unfortunately, the tiny textual window of a hex editor provides only a small glimpse into a file's internal structure, and fails to provide any big picture context. I plan to release an open-source tool at Black Hat this summer that combines many of the strengths of the hex editor with useful visualization techniques and improved navigation. My initial results show that files of 100MB or more can be quickly analyzed visually, but similar analysis would be extremely difficult using a traditional hex editor."

Dr. Conti's feedback validates my point, albeit a simple one. With resources like his book and websites like SecViz and VizSec *there should be nothing holding you back from taking*

1 http:// afterglow.sourceforge.net.
2 http://code.google.com/apis/visualization/documentation/gadgetgallery.html.
3 http://www.cs.umd.edu/hcil/treemap.
4 http://www.cyberconflict.org.

the leap into the study of security data visualization. Inspired at the CCSA workshop by the desire to view all the data we must consume as information security professionals in a different light, I set out learn more about the security data visualization (secviz) practice. It is my hope that what I've learned be conveyed to you in a manner that leaves you inspired too.

Security Data Visualization Tools

I will likely upset some existing secviz fans as I will leave AfterGlow[5] out of this discussion, having covered it recently in November's discussion of Argus. Nonetheless, it is truly excellent and you should explore it yourself. The other tool many consider essential that I'll not be discussing here is Treemap,[6] a "space-constrained visualization of hierarchical structures," but like AfterGlow, don't leave it off your list.

In the hopes of enticing you to undertake the practice of secviz, I'll provide details on three specific offerings: InetVis, Rumint, and TNV.

InetVis is a 3-D scatter-plot visualization for network traffic; Rumint is an open source network and security visualization tool; and TNV is a visualization tool for analyzing network packet capture (pcap) data.[7]

As I researched each of these tools, I decided to use a common network capture in order to show the same data via various visualization opportunities. The resulting capture was taken over five minutes post-infection of a sandbox victim with a typical Storm variant. Given how extraordinarily noisy such infections can be, this capture made for great visualization opportunities.

In order to make the experimentation process simple for you, dear reader, I've posted *ecard.cap* to my website.[8]

InetVis

I have a particular fondness for the visual output created by InetVis. Not to distract from any other offerings, but we likeses the pretty pictures, my precious. InetVis is adopted from Stephen Lau's *Spinning Cube of Potential Doom*. Network packets are plotted by:

- Destination address (home network) plotted along blue x-axis (horizontal)
- Source address (external Internet range) plotted along red z-axis (depth)
- Ports (TCP and UDP) plotted along green y-axis (vertical)
- ICMP traffic plotted below TCP/UDP cube grey/white ICMP plane[9]

The InetVis package conveniently includes all its dependencies, so you needn't hunt down any other applications to achieve a working state quickly. I will suggest that you use InetVis on a Windows XP installation, rather than Vista.

Getting started is as simple as executing *inetvis.exe* from your installation directory. You'll immediately be introduced to the InetVis Control Panel. To get started you can leave everything default in this UI, but explore View, then Plotter Settings (see Figure 1).

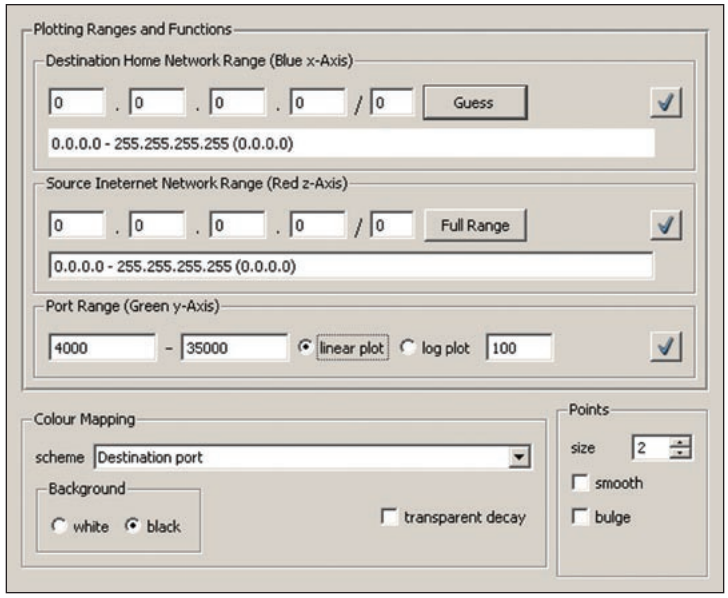


Figure 1 – Setting up InetVis

As you begin to work with your own captures, you'll want to experiment with how you set the *Home Network Range* (Blue x-axis). InetVis will bug you incessantly to set this, but it isn't mandatory; just click OK each time it asks. With *ecard.cap*, used for all the tools discussed this month, I found that not defining the home network gave me better results. I found this was simply a function of the fact that InetVis sees the home network as the destination and all Internet addresses as the source. But as well we know, when viewing a capture from a Storm infection, our perspective is really the opposite, where the infected host is the source and all the other P2P participants are the destination. Our screenshots will prove this out. See Figure 2.

I also narrowed the port range a bit. Where default is 0-65535, Storm tends to chat a great deal between 5000-30000 give or take, so I set the *Port Range* to 4000-35000 to create a more focused view. You'll find that the resulting visualization shows a single point of reference along the red z-axis (source) as the infected host is utilizing a Class C address of 192.168.248.105. You'll also note that it sends a prism of visualized disease across the blue x-axis (destination) given the hundreds of unique hosts in the botnet's P2P mesh that my victim chatted with over the four minute capture. Finally, you'll note the full rainbow across the y-axis (green) that represents the port range as this particular capture spread the whole spectrum.

5 http://afterglow.sourceforge.net.

6 http://www.cs.umd.edu/hcil/treemap.

7 http://www.vizsec.org/applications/open-source-applications.

8 http://holisticinfosec.org/toolsmith/files/pcap/ecard.cap.

9 http://www.cs.ru.ac.za/research/g02v2468/inetvis.html.

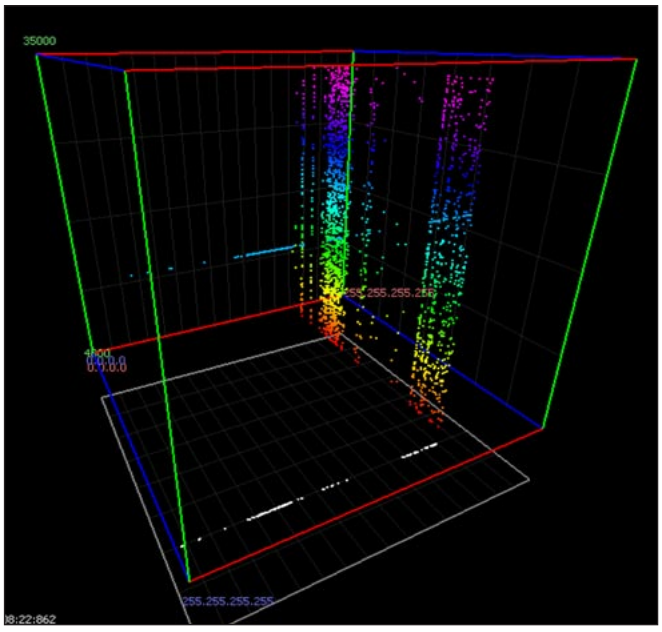


Figure 2 – A single Storm infection as visualized with InetVis

What makes me giddy about InetVis is the ability to rotate the cube. Hold your left mouse button down on the left side of the view and drag to your right; the cube will rotate in the direction your pointer travels. Remember Dr. Conti's claim that a picture is worth a thousand packets? 8000+ in this case, but you definitely get the picture.

Rumint

You'll find Rumint easy to install and run as well. Execute *Rumint_214.exe* from your install location and you're off. The UI is entirely obvious; select File, choose Load PCAP Dataset, and select the capture of your choice. As we're using my *ecard.cap* example, I selected it accordingly and chose my favorite options in the View menu before selecting Play. *Text Rainfall* is interesting if there's anything of use being passed in the clear, but *Parallel Coordinate Plot* is where the magic awaits you. If you limited yourself to one view, this one is it. Before you begin, increase the number of axes to 6. You can choose as many as 19, but for screen capture's sake I narrowed it to manageable settings. Further, six axes allowed me to select ideal parameters for visualizing Storm, as follows: *Packet Length, Source IP, Dest IP, UDP Source Port, UDP Dest Port,* and *TTL*. Click *Play* and watch the parallel coordinate plot begin to flesh out. By just the 40th packet, watch as the UDP destination port axis begins to explode, as well you'd expect given the attributes of the malware traffic we're analyzing. With the slider bar in the primary UI, you can replay each packet and watch it visually as it plays across your chosen parameters. See Figure 3.

Experiment with the parameters you set as axes. You'll quickly find varying levels of relevance, but once you tune and optimize, I promise you, something will leap out you that you might otherwise have missed.

TNV

TNV (The Network Visualizer or Time-based Network Visualizer) requires Java, but as such, runs anywhere. On Windows, execute *tnv_startup.bat* to begin. Define your home network IP address, then select a database type (embedded is easiest). The default display settings will give you an ample glimpse at your data, but give the sample capture we've been discussing some time to load. Once loaded though, you'll have some interesting opportunities to manipulate the data. Be cognizant of the time slider at the bottom of the UI, while the other feature to take immediate advantage of is in the View menu; specifically, *View All Packet Details*. You can then right-click on an entry of your choosing to (you guessed it) view packet details. See Figure 4.Be patient with TNV, it's the slowest of all the apps tested, but worthy of a look. There's a great *start* read available.[10]

10 http://tnv.sourceforge.net/start.php.

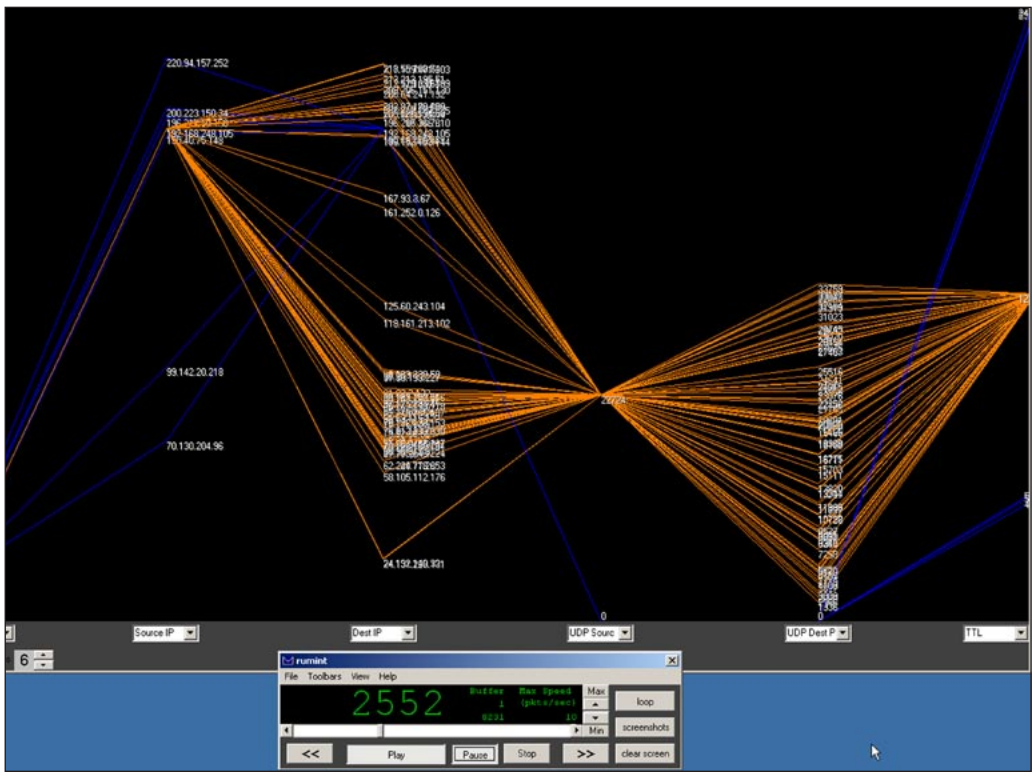


Figure 3 – RUMINT's exquisite detail

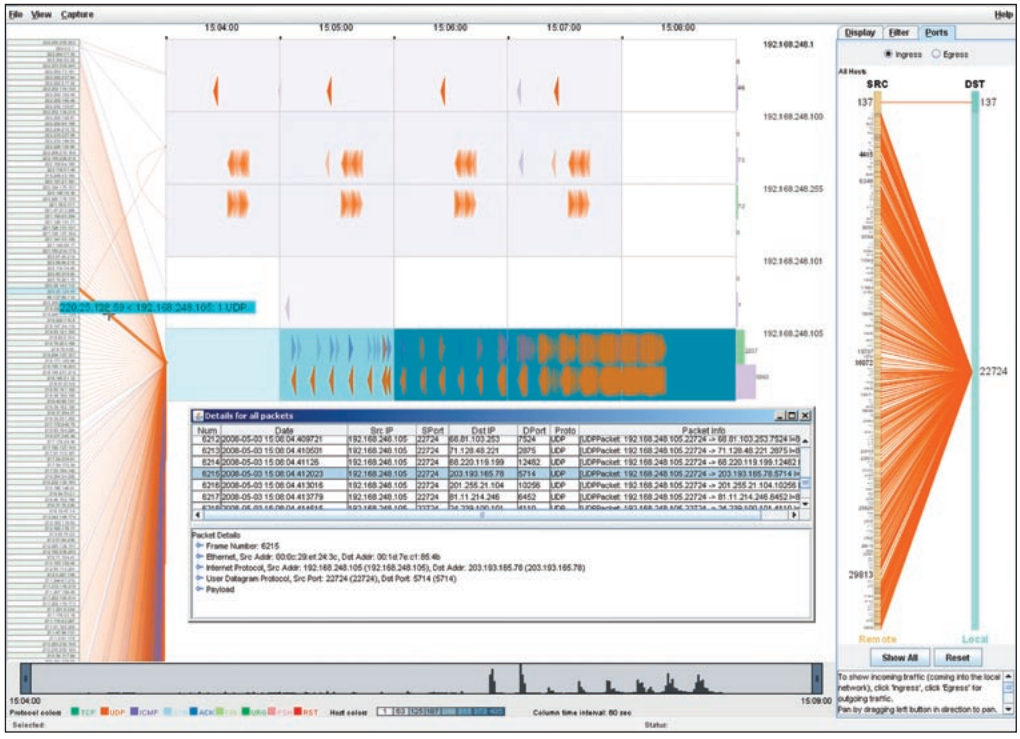


Figure 4 – TNV in action

DAVIX

Raffael Marty, the author of AfterGlow, and Jan P. Monsch are releasing DAVIX (Data Analysis and Visualization Linux), a security visualization LiveCD, at BlackHat this August.

From Jan: "The idea behind DAVIX is to provide an integrated out-of-the-box environment for data and visualization analysis. For the first release, scheduled for Black Hat/DEFCON, we have assembled a set of 25+ visualization tools and some basic tools for fiddling with log files. The long-term goal is to provide a good set of tools supporting the complete process of visual data analysis and possibly a software component which allows easily integrating all these different tools."

Dr. Conti's offered words of advice on this very subject: "For InfoSec practitioners just starting out in visualization, the best place to start is by experimenting with existing tools. Unfortunately, finding and correctly installing these tools can be a tricky process. That is why I'm excited by the DAVIX project. It integrates a wide range of tools into one easy to use distribution."

Benefits and drawbacks

To really draw out (pardon the pun) the full features of these apps, you'll need to dedicate some time to the cause. But you can't go wrong. I've already enjoyed two specific occasions where the use of security data visualization tools provided me with discovery I'd have otherwise likely missed, literally striking the very chord I was yearning for at the CCSA workshop.

Remember that these tools allow you to capture data from your network interface, an aspect we didn't cover here at all.

Read Dr. Conti's *Security Data Visualization* and peruse the two secviz websites; you'll not be disappointed.

In conclusion

A few final details, if I may. Raffael Marty also has a new book coming out July 30 via Addison Wesley called *Applied Security Visualization*.[11] I'm definitely looking forward to its release.

You also enjoy an article called What a Botnet Looks Like.[12]

Researcher David Vorel mapped interconnected, bot-infected IP addresses and created a nice visualization. Only one problem: he didn't use AfterGlow! ;-)

Given word and space limitations, I've done the secviz discipline limited justice here, but I truly hope you'll give it its rightful due. You cannot stop what you cannot see. Cheers, until next month…

Acknowledgments

My great thanks to:

—Greg Conti, PhD, Assistant Professor, LTC, U.S. Military Academy at West Point, author of *Security Data Visualization*.

—Raffael Marty, Security Visualization: http://secviz.org and raffy.ch/blog.

—Jan P. Monsch, Senior Security Analyst and DAVIX Initiator Jean-Pierre van Riel, InetVis.[13]

Resources

—CCSA, http://www.cyberconflict.org

—RUMINT, http://rumint.org

—SecViz, http://www.secviz.org

—Vizsec, http://www.vizsec.org

About the Author

Russ McRee, GCIH, GCFA, CISSP, is a security analyst working in the Seattle area. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.

11 http://www.informit.com/store/product.aspx?isbn=0321510100.

12 http://www.csoonline.com/article/348317/What_a_Botnet_Looks_Like.

13 http://www.cs.ru.ac.za/research/g02v2468/inetvis.html.