Stephen Chan
sychan@sims.berkeley.edu

Visualizing Network Security

Project Goals

Log information is gathered by computer systems constantly, especially alert logs by security tools. These logs are textual and it is hard to get a "big picture" of what is happening and what (if anything) is different from the norm.

The visualization built and tested is a treemap of alerts generated by a network IDS system (Bro). Log are parsed into fields that a treemapping tool can parse and the treemap will be ordered in different ways to see if it provides insight into current or historical activity.

In addition, a database will be used to store historical information (average, mean, etc…) about the number and type of alerts seen in the past, and this will be used to color code the treemap blocks. The goal is to make it easy to spot changes in the type and frequency of alerts. The window of time used to compare against the past will be varied (we may compare the last 1 hours worth of data, last 6 hours, last 12 hours, last 24 hours, etc… against the long term average).

Related Work

Security oriented visualizations are often tied to the topology of the network and its hosts. This visualization is not tied to hosts on the local network, but instead looks at overall alert activity. We believe that this will be useful as part of a broader goal for detection of distributed attacks – either attacks that are distributed over large numbers of hosts, or attacks which are distributed over a relatively large span of time.

For the immediate tool, we are interested in 2 aspects:

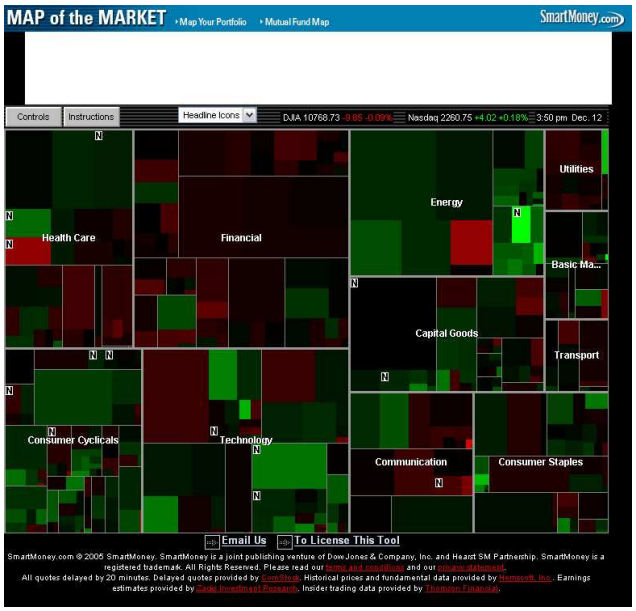
1) A real-time dashboard that shows the current state of the intrusion activity
2) An analysis tool for examining large quantities of data to see if there are trends and patterns that can be identified for developing further intrusion detection tools.

Map of the Market

http://www.smartmoney.com/marketmap/

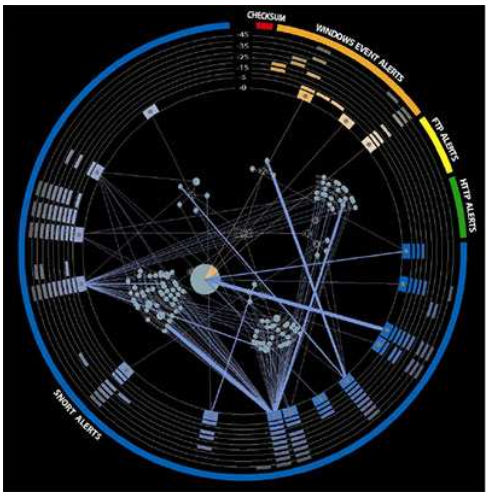
For the prototype, we use a TreeMap based visualization, similar to Map of the Market. The premise is that Map of the Market provides a real time "Dashboard" of market activity, so that users can see changes and be able to zoom in on activity that is of interest.

In the field of computer security, what is interesting are anomalies. Operational computer security is often a game of anomaly detection, and hopefully a real-time visual dashboard can facilitate the timely detection and response to anomalies.



Stephen Chan
sychan@sims.berkeley.edu

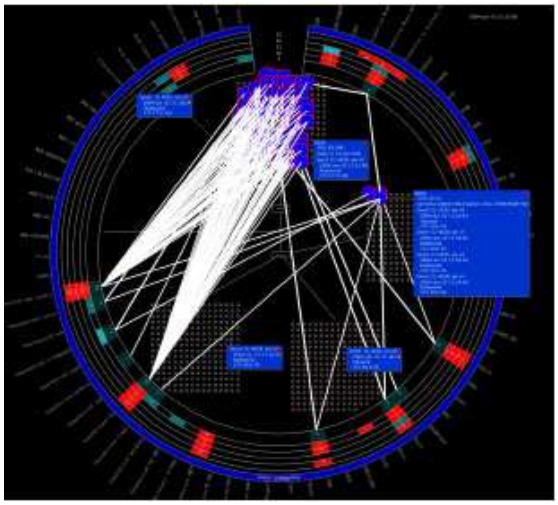
VizAlert http://www.ncassr.org/projects/sift/vizsec/proceedings/2005/paper13.ppt



VisAlert was chosen because is one of the visualizations that my peers have found to be very useful in terms of providing situational awareness. It is in the process of becoming a commercial product.

Tools such as VisAlert give a good sense of the current state of your network, but don't provide a historical baseline against which to compare activity. This tool is targeted specifically at visualizing the current network state, so the design may not provide useful visualization when there are a large number of alerts – such as over a long period of time, or when the network is under heavy attack.

In the second screen shot from VisAlert we see the representation of a large attack. Notice the number of lines and the density of the graphics – on our networks we typically see scans that target thousands of hosts at a time – even if some form of aggregation were to be used, there would be scores of such attacks per day: visualizing this over the span of days, weeks or months would become a distraction.



Ben Schneiderman's talk at Usenix Security 2005

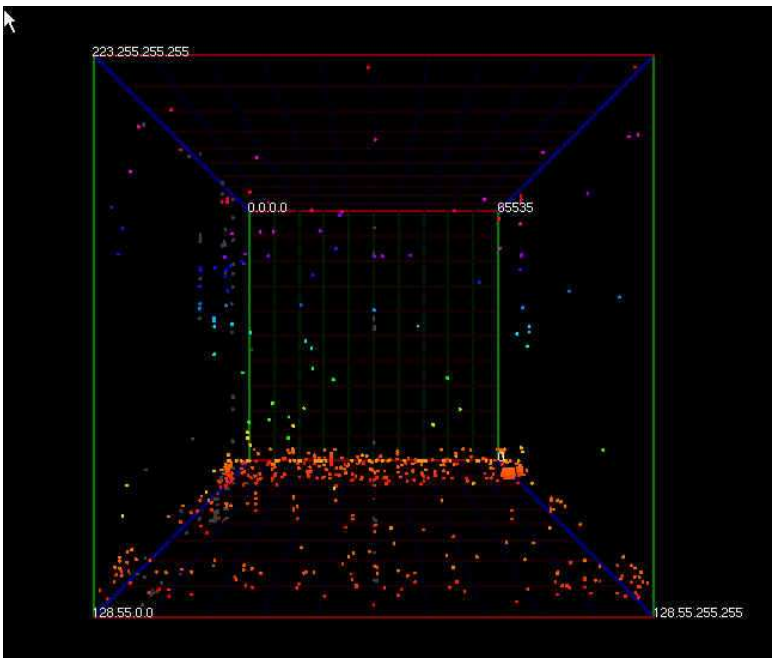
http://www.usenix.org/publications/login/2005-12/openpdfs/sec05summaries.pdf
http://www.cs.umd.edu/hcil/pubs/presentations/SecurityUSENIX1_files/frame.htm

At the 2005 Usenix Security Conference, Ben Schneiderman talked about approaches and opportunities for Information Visualization in dealing with security. He presented TreeMap (among others) as a potential tool for visualizing security issues. While he did not present a specific security tool for visualizing intrusion detection attempts, his talk inspired the security oriented visualization explored here

Stephen Chan
sychan@sims.berkeley.edu

The Spinning Cube of Potential Doom

http://www.nersc.gov/nusers/security/TheSpinningCube.php



The campily named "Spinning Cube of Potential Doom" was a visualization demo by another member of my security team. It was a visualization of TCP connections from the same data source used in this project.

The cube is a representation of 3 parameters: source IP address, destination IP address and port number. Each dot represents a TCP connection that was either attempted, or completed. White dots represent complete connections, while colors dots represent connections that did not complete. A quick viewing of the image shows that most of the connections are incomplete, and further analysis shows that these connections are mostly attempts to scan the network for vulnerable services.

The purpose of the Spinning Cube is not so much to show operational security information, but rather to make it clear how much of the traffic that is found on our networks is malicious. One thing that becomes very clear with this visualization is that there is so much traffic, it is really necessary to apply some very intelligent form of filtering, or else aggregate the information so that if we see a single host scanning 10,000 machines, it is identified as a single event.

Discussion

The "mantra" of information visualization, as Ben Schneiderman describes it in the slide 42 of his presentation above is: *Overview, zoom & filter, details-on-demand* The tool provided, TreeMap fulfills all these requirements:

- The TreeMap is an high level overview
- It is possible to zoom into the treemap
- Filtering is available on any parameter
- Details of the event are available

In addition, my observations of work practices in computer security are that:

1) Security professionals come from very different backgrounds and want to look at different data, or else process the same data in different ways

Stephen Chan
sychan@sims.berkeley.edu

2) In computer security, it is never quite clear exactly what subset of the data you have at hand is relevant for answering a given question: is the IP address, port, time of day, type of attack, etc…
3) There is a broader iterative process of data collection, analysis, generation of tools/procedures as well as specific and very dynamic individual work practices that must be adapted to by the tool.

Consequently, tools that are too rigid in terms of how they represent and filter data, or which impose too many requirements or assumptions about the environment's practices or toolset are problematic for security administrators. The goal is to develop something that is lightweight and flexible, that adapts to the security administrator's work patterns and approach to analysis.

Description of Prototype

The prototype is based on TreeMap 4.1 (http://www.cs.umd.edu/hcil/treemap/) a treemapping tool created by the HCI Lab at University of Maryland. It is a java based tool that reads structured text files and allows the user to create treemaps clustered on any of the fields found in the datafile. In addition, the tool allows control over the treemap layout supporting squarified, strip, and slice and dice layouts. Coloring, sizing and details of border size, fonts and filtering are all supported.

The dataset consisted of 15,271 alerts from our intrusion detection system. These alerts span a 3 month period and provided enough datapoints to perform a basic frequency analysis of the types of attacks and the destination ports being targeted by network attackers. The data was parsed into a structured text file and loaded into a MySQL database, where computations such as frequency, log calculations and other derived or calculated values were generated. The data was then re-exported into a tab delimited file for import into TreeMap. The following table shows the fields in the final dataset:

- Unixtime(integer): the unix timestamp of when the alert was generated, # of seconds since the epoch Jan 1, 1970
- Time(string): textual timestamp
- Type(string): type of alert generated by Bro
- Subtype(string): sub classification of alert
- Srcip(string): source IP of system that triggered alert (attacker)
- Srcport(integer): source port of attacking host
- Destip(string): destination IP of the system that triggered alert (target)
- Destport(integer): destination port of the attacking host
- Numtargets(integer): the number of hosts affected by this attack
- log(numtargets) (float): log base 10 of the # of hosts effected. Needed because the range of hosts targeted ranges from 1 to 65000, making it difficult to use numtargets as a sizing parameter in the treemap.
- Typefreq(float): the relative frequency (from 0 – 1.0) of the alert type
- Typerarity(float): the inverse of the type frequency, used as an indicate of how rare a certain type of attack is

Stephen Chan
sychan@sims.berkeley.edu

- Destportfreq(float): the relative frequency (from 0-1.0) of the port being targeted. The ports being targeted had a power law/zipf style distribution – some ports were targeted very, very often, and other ports were targeted much less frequently.
- Portrarity(float): Because of the distribution of target ports was so heavily skewed, we calculated the rarity as the destportfreq/(average of all frequencies). This calculation is somewhat like using a standard distribution oriented metric (which in hindsight, would have been an interesting calculation).
- Maxrarity(float): The maximum of the typerarity and portrarity values.
- octet1(integer): the first octet of the source IP address
- octet2(integer): the second octet of the source IP address
- octet3(integer): the third octet of the source IP address

The frequency and rarity values were calculated as a way to identify anomalous entries – despite the first level of filtering already performed by the IDS's rule system, there are still too many alerts and it is important to provide some form of filtration for "interesting" alerts. What we find during the user interviews is that users have very custom personal metrics of what is interesting!

In the following images, we show some of the sample treemaps generated from the dataset.

Stephen Chan
sychan@sims.berkeley.edu

Figure 1: Treemap displaying type, subtype and source IP address of attack. Size is related to rarity of the type of attack (1/frequency) and color is based on the timestamp of the alert.

In this visualization, we use size and color to look at roughly one days worth of data about alerts. The typerarity metric is used to size an alert for interest, and the color is used to indicate time (brighter colors are more recent). In this visualization, we are most interested in anomalous entries, based on frequency of the attack type – in this regard, this visualization seems to be successful (at least, *in the abstract*).

Stephen Chan
sychan@sims.berkeley.edu

Figure 2: Map of alerts with sizing based on log(numtargets) and color based on time of day

This map displays the use of the TreeMap to show the attacks, with bias given to attacks that affect a larger number of hosts. It turns out that attacks target large numbers of hosts are typically scans, which are quite uninteresting to security staff because the IDS has identified the attack already and blocked further scanning.

Stephen Chan
sychan@sims.berkeley.edu

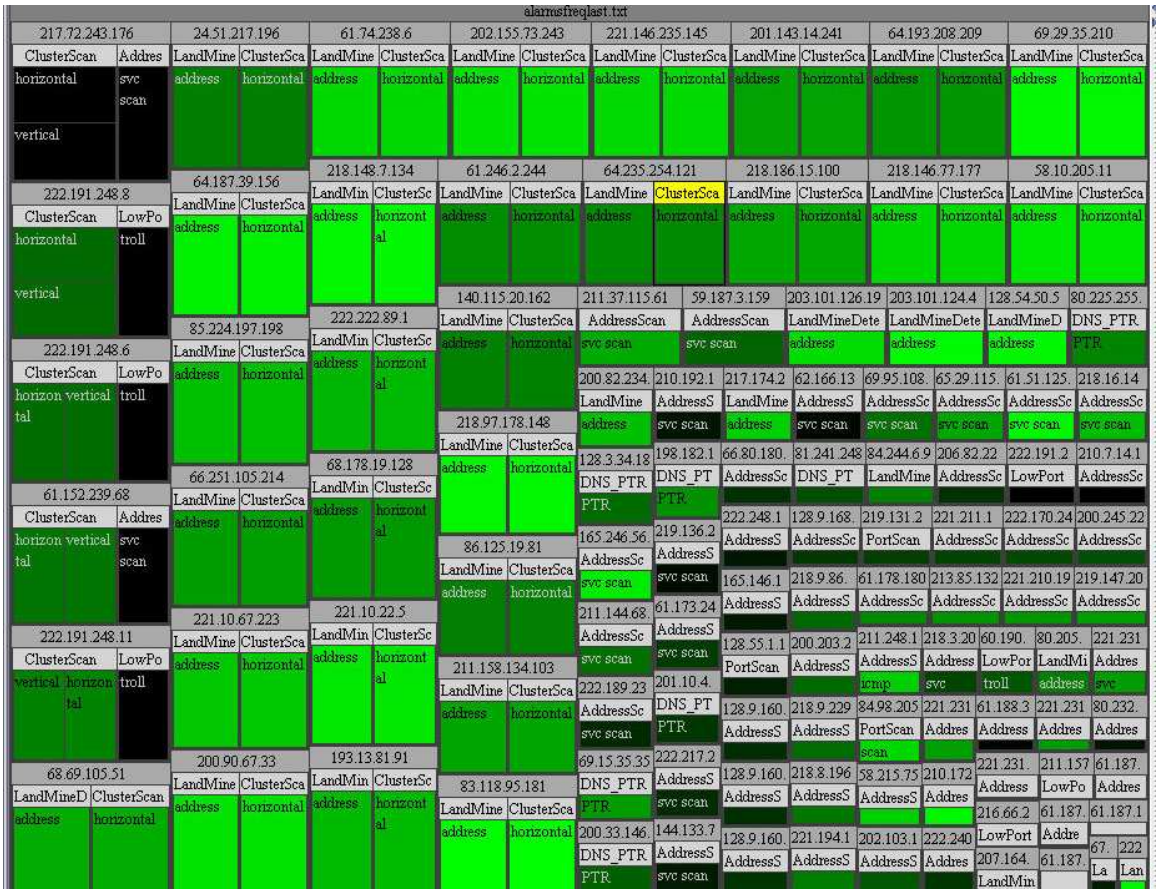


Figure 3: Map that groups attacks by srcip and subtype

This map groups attacks by source IP, to bring out patterns in the types of attacks that any individual host is attempting. We see that hosts that perform scans often use horizontal cluster scans, as well as triggering the “landmine” detection (traffic directed towards IP addresses where there are no hosts).

Stephen Chan
sychan@sims.berkeley.edu

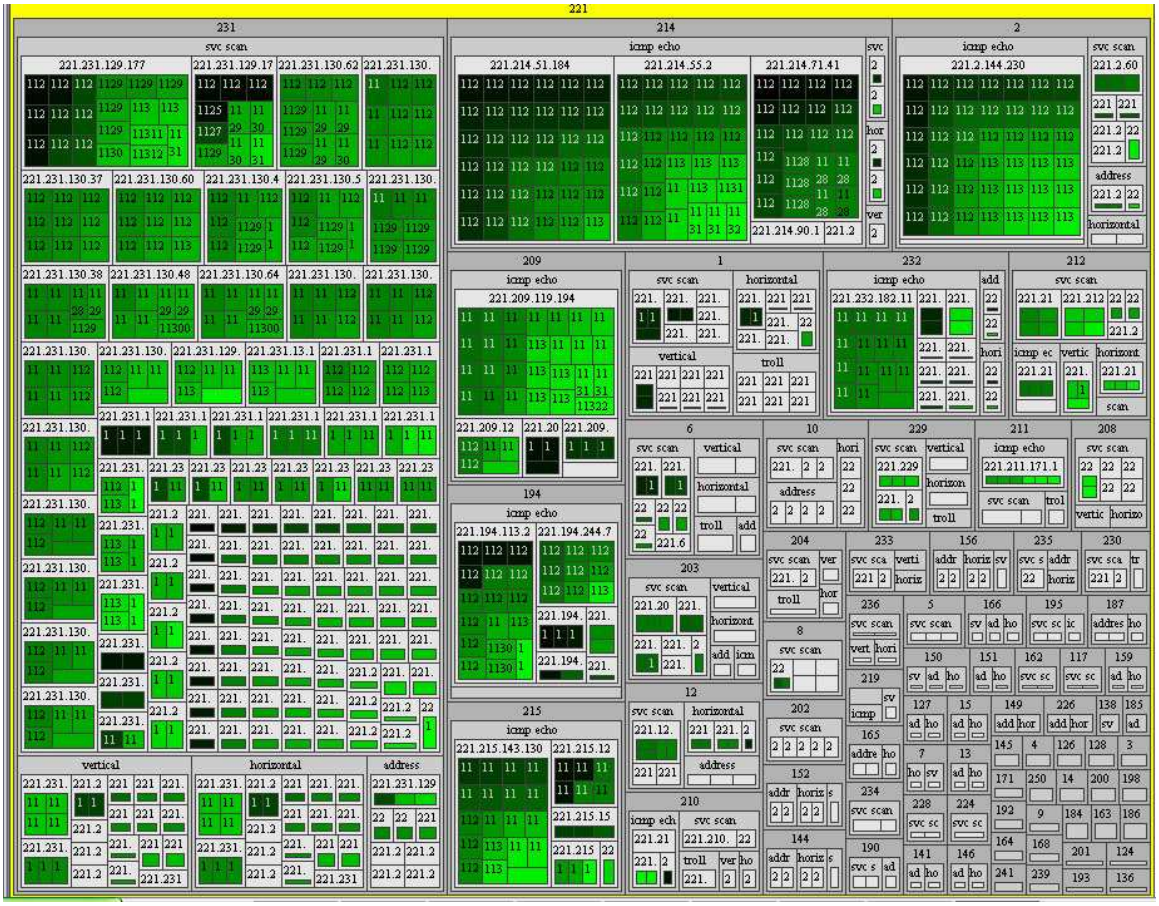


Figure 4: Alarms clustered by first and second octet, than srcip - zoomed into the 221.*.*.* address range

In this final image, we look at the full dataset, clustered by the first and then second octet, then by the src ip address. No special sizing is used, but colors are used to indicate time. The intention is to see if we can spot patterns of attack activity originating from IP ranges on the internet.

What we see here is that the 221.231.*.* IP range is the source of a lot of scanning activity. The colors and structure indicate that very similar kinds of attacks are coming from this address block, and that they occur at very similar times. This is possible indication of a distributed scanning attack from a cluster of hostile machines in that address space.

A whois lookup returns the following information:

```
whois 221.231.
[Querying whois.apnic.net]
[whois.apnic.net]
% [whois.apnic.net node-2]
% Whois data copyright terms    http://www.apnic.net/db/dbcopyright.html

%WARNING:905: fixed lookup key
%
% The key "221.231." has been changed to "221.231.0.0" for lookup.

inetnum:        221.231.0.0 - 221.231.0.63
netname:        YANCHENG-XINXINGCHENGUANG-NETBAR
descr:          YanCheng Xinxing ChenguangNetbar
```

Stephen Chan
sychan@sims.berkeley.edu

```
descr:          Yancheng City
descr:          Jiangsu Province
country:        CN
admin-c:        CH455-AP
tech-c:         NT35-AP
changed:        ip@jsinfo.net 20031028
status:         ASSIGNED NON-PORTABLE
mnt-by:         MAINT-CHINANET-JS
mnt-lower:      MAINT-CHINANET-JS-YC
source:         APNIC

route:          221.228.0.0/14
descr:          CHINANET jiangsu province network
country:        CN
origin:         AS23650
mnt-by:         MAINT-CHINANET-JS
changed:        ip@jsinfo.net 20030630
source:         APNIC

role:           CNCGroup Hostmaster
e-mail:         abuse@cnc-noc.net
address:        No.156,Fu-Xing-Men-Nei Street,
address:        Beijing,100031,P.R.China
nic-hdl:        CH455-AP
phone:          +86-10-82993155
fax-no:         +86-10-82993102
country:        CN
admin-c:        CH444-AP
tech-c:         CH444-AP
changed:        abuse@cnc-noc.net 20041119
mnt-by:         MAINT-CNCGROUP
source:         APNIC

person:         ni tongquan
nic-hdl:        NT35-AP
e-mail:         nitongquan@sina.com
address:        yancheng xinxin city east street NO.17
phone:          +86-515-8511928
country:        CN
changed:        ip@jsinfo.net 20031028
mnt-by:         MAINT-CHINANET-JS
source:         APNIC
```

We see that several networks in China are the source of these attacks. It often turns out that many address blocks on the internet are considered “bad neighborhoods”, and that in fact, some networks (such as those assigned to eastern European countries with a history of being bad network citizens) are frequent sources of network attacks: indicating that a further derived field such as “ip source attack frequency” should be generated and used for either sizing or coloring of treemap fields.

It is actually the case that certain network neighborhoods are “redlined” by network security professionals, and any traffic from those networks is automatically considered suspicious, generating alerts.

Results of User Study

The user study included 3 professional engineers tasked with network security. All were recruited through work contacts(Lawrence Berkeley Lab):

1) Participant A: an engineer who builds and operated network security infrastructure and is on call for security alerts. Participant A works at a

government supercomputer center that serves scientific computing and has over 15 years of work experience. Has an undergraduate degree in physics.
2) Participant B: the lead of the networking and security groups, who is a very strong technologically but is not tasked with immediate network security per se. However, as the supervisor, he is very concerned with how such a tool might improve the productivity and/or effectiveness of the security team. This participant is the manager of participant A, and has over 15 years of work experience as well. Has a Master's Degree in statistics
3) Participant C: the incident response manager for all network security a large government research laboratory. Participant C's job is to monitor and response to network security threats across the entire lab of several thousand people. This participant has over 20 years of experience.

All of the participants are considered senior technical staff and participate in both operational work, as well as development of new tools and techniques for improving security, as well as having published in technical journals. All were skilled in both operations as well as software development.

The user study was originally structured to have the users step through several tasks themselves, however based on the first user studied, it became clear that the user interface offered too many options and responsiveness of the Java based tool suffered when dealing with large datasets. A large period of time was spent learning and navigating the user interface for TreeMap instead of looking at the visualization and how it night fit into the user's workflow. In addition, partway through the first interview, a security incident arose and all staff members became immersed in dealing with the incident.

Consequently, to save time, the tasks of loading and setting up the visualization were taken over by the tester (myself), and exploratory tasks were described by the user, but actually implemented by the tester. The interviews typically lasted from 45 minutes to an hour, and there was far too much information to be described in full in this paper – much of the interviews became discussions of what would be necessary to make such a tool useful for each particular user. Summaries results are described below for each user.

Participant A:

- Initial amount of information was overwhelming (for 15000+ events)
- Idea of TreeMap took some explaining, but eventually was grasped
- User interface was slow and unresponsive for large datasets
- Too many features on the UI – wanted a cleaner, simpler and more responsive interface.
    - Would be satisfied with only the following controls available:
        - User specified clustering/hierarchy
        - User specified coloring, sizing and labeling
- The data presented was interesting and somewhat useful, however wanted to have a query language along the lines of SQL, and access to a large datastore that could be used to generate calculations, and baselining

Stephen Chan
sychan@sims.berkeley.edu

- Some sort of classifier system (perhaps based on Bayesian categorizers) would be very useful for identifying interesting vs non-interesting attacks.
- Visual approach is good, if only it allowed the flexibility to perform ad hoc retrieval and calculations

Overall, the user was very interested in the tool, but needed something that was faster, cleaner and with an ad hoc query language. The idea of having a SQL query engine tied into a large database of all network traffic (as well as the alerts) was very appealing.

When SpotFire and Tableau were shown to the user, the ScatterPlot tool was seen as very interesting.

Participant B

Participant B did not ”drive” the UI directly, in order to factor out UI issues from the usefulness of the visualizations.

- TreeMap was an unfamiliar visualization
- Initial amount of information was overwhelming
- Visualization task needs to have flexible filtering rules – much of what is being displayed is simply not interesting. Rather than have it smaller, would rather have it not be displayed at all
- In discussions about detection of distributed attacks, he pointed out that in attackers cannot randomize all possible parameters in the attack traffic (for example, many IP based header information such as time to live, sizing windows, and other very specific header information), so a tool that shows all the parameters of network traffic would be handy
    - Based on a description of multidimensional viz tools such as a parallel coordinate system viz tool for the detailed header information could be used as a discovery tool for identifying patterns of distributed attacks.
- Tool should be integrated with a workflow, so that clicking on an event should either drill down into further information sources, or link into an incident response system
- Details of the visualization are not so important, as whether it impacts the effectiveness of the security team – as such it needs to be built in a way that ties into the rest of the work of the team.

Participant C

- Too much data, and most of the alerts are non-interesting, so aggressive filtering is necessary
- Metrics of type rarity and port rarity as a start, but further information that might be interesting are:
    - Did any of the scans successfully connect to a target before being shut down? An attack that manages to connect to a target host is interesting irrespective of how frequently seen the attack is

    - Is this alert bracketed by other interesting network events? For example, a "common" scan that is correlated with suspicious traffic such as IRC connections (often used by hackers) from the same host within a time window becomes far more interesting
- Wanted to have open ended ad-hoc filtering rules, as well as methods of having events on the display "click through" to very detailed information sources.
- Some kind of classifier system such as a Bayesian tool or AI-ish analysis to flag the item as interesting vs. uninteresting is needed to make it really useful.
- Tool also needs to tie into his existing toolset (shell scripts, awk, etc…)

Summary

The key issues raised during user testing were:

- UI needs to be faster and cleaner for the user to use it.
- Different users are interested in different filtering rules
- Tool needs to support existing work practices:
    - Trouble tickets/incident response
    - Drill down to get more specific information for triage
    - Be very flexible in terms of data sources and amount of data that can be crunched
        - Data sources might be a SQL database, large flat file databases, command line interfaces into other tools (such as whois)
    - It was a frequently expressed desire that they would like to be able to click on an event that was graphed and trigger some kind of custom action
- The need and desire for such a tool was common to all the participants, however flexibility and modularity are needed to make it genuinely useful.

Results Not Achieved (but would have liked to)

It would have been good to have enough time to allow users to work with different visualization tools such as SpotFire and Tableau. The problem is that visualization is a relatively specialized field, and the tools have non-trivial learning curves (even something as specific as a TreeMap). In discussing and showing some of the other visualizations available, it became clear that the participants found them to have lots of potential, but would have needed far more time than was available in order to evaluate them.

It would have also been interesting to perform some other forms of metrics for interesting vs not-interesting. The type rarity and port rarity were an intriguing start, but it became clear that each user wanted to have some ad hoc capability to generate different metrics and experiment with them.

A tool such as TreeMap that offers a few more types of visualizations (ScatterPlots and Multi-coordinate) but with a fewer controls over the details of the rendering, but which offered rich control over incoming data sources and triggers for

Stephen Chan
sychan@sims.berkeley.edu

events when an event is “clicked on” seems to be the tool that most of the participants were looking for. The statically generated metrics imported into TreeMap were not adequate for their uses – Tableau’s approach of hooking in a SQL backend might be a very good approach.

Since most of these users were competent software developers, if some toolkit like PreFuse exposed a high level interface into a scripting language (such as Perl, or even Groovy) that would allow rapid generation of visualizations from data extracted via the scripts, as well as callouts for external actions when items were “clicked on”, a viable and useful tool for Security visualization would likely result.