Global Information Assurance Certification Paper

How Can You Build and Leverage SNORT IDS Metrics to Reduce Risk?

GIAC (GCIA) Gold Certification

Author: Tim Proffitt, tim@timproffitt.com
Advisor: David Shinberg

Accepted: August 19th 2013

Abstract

Many organizations have deployed Snort sensors at their ingress points. Some may have deployed them between segmented internal networks. Others may have IDS sensors littered throughout the organization. Regardless of how the sensor is placed the IDS can provide a significant view into traffic crossing the network. With this data already being generated, how many organizations create metrics for further analysis? What metrics are valuable to security teams and how are they used? What insights can one gain by good metrics and how can that be used to reduce risk to the organization? The paper will cover current technologies and techniques that can be used to create valuable metrics to aide security teams into making informed decisions.

1. Introduction

Metrics are used in many facets of a person's life and can be quite beneficial to the decision making process. Is a car still getting the miles per gallon it should be? Are invested stocks increasing in value and at a rate that is desirable? What should the thermostat be set to this summer to minimize the amount of energy consumed? How much weight can one loose each week to get ready for trips to the community pool? Can a family afford to lease a car over the next three years with a variable based income? Technologists should be asking in the same vein, questions about IDS activities that can be answered by metrics. Is the sensor operating as intended? Is the sensor alarming on the correct events? Is the IDS seeing an increase or decrease in events and should this matter? What events should an analyst find interesting? Is a sensor being managed in a manner that offers the best protection for the organization? Is the organization being attacked and not aware of it?

The Snort IDS, created in 1998, has seen a very large deployment with a long history. The open source community has richly supported this tool and offers additional GUI tools that generate graphical views and metrics. Several tools such as SQUIL and Snorby have emerged to provide a nice platform for analysis by a security team.

Regardless of the technology utilized to generate the sensor alarms, security teams can create processes that will generate valuable data. Utilizing statistical techniques against collected data will allow security teams to build metrics. These metrics can then be used to in decision making by management to reduce risk to the organization.

2. Creating Metrics

"On any given network, on any given day, Snort can fire thousands of alerts. Your task as an intrusion analyst is to sift through the data, extract events of interest, and separate the false positives from the actual attacks." (Beale, Baker, et al, 2006)

The term "metrics" describes a broad range of tools and techniques used to evaluate data (Greitzer 2005). The evaluation of that data is then used as a measurement compared to one or more reference points to produce a result. A simple technology

security example would be to collect incomplete 3 way TCP handshake packets to a destination, over a period of time, with the intent to show a trend. This extremely simple example is one of many situations where technology metrics can help a manager make informed decisions about their security infrastructure. A good security team is concerned if the above IDS metric was trending upward by a factor of two every month. What if the same metric trended downward by half every month? In the first case one could have an IDS showing that a resource is under a prolonged attack. In the second case the IDS could have a rule misconfiguration allowing conversations to be conducted but not monitored. Either way this would be valuable data to a decision maker or at least a situation that would need attention by a member of the team responsible for the IDS.

The technology auditing focused organization ISACA defines information security as the protection of information assets against the risk of loss, operational discontinuity, misuse, unauthorized disclosure, inaccessibility, or damage (Brotby 2009). Technology security is concerned with the potential for legal liability that entities may face as a result of information inaccuracy, loss, or the neglect of care in its protection. A more current definition from CSO management circles describes information security as the triad of confidentiality, integrity and availability. This definition can cause an issue for security teams. How do security teams go about measuring confidentiality or integrity? One can measure availability as it pertains to networks outages and systems uptimes but how can metrics be applied to availability as it pertains to technology security? These are very difficult questions to answer. The above simple metric of the sensor recording the TCP 3 way handshake does not answer these questions, at least not standing on its own. A metrics program needs to develop sound metrics to answer these questions and others that executive management will need for steering an organization.

2.1. What makes a good metric?

Bad metrics can be found most everywhere. Vendor dashboards are littered with them, presentations contain them and security teams expect management to make decisions off them. Take a traditional, out of the package IDS metric that shows the number of signatures being seen by the sensor. This can be valuable data, especially for the IDS team, intrusion response team or the individuals responsible for hardening infrastructure. Knowing that a SYN flood is being executed against a critical web server

is important but the metric says little of the overall security of the organization. Are the intrusion sensors in their current configuration protecting the organization? Is the protection the security team provides now better or worse than last year? Can the budget being allocated on managing the IDS be utilized better in a different control? Smaller, technical metrics should be rolled up into a more comprehensive security picture if security teams are going to be successful in creating good metrics and getting the point across to the upper management of the organization. A good start on metrics, measurements and monitoring information can be summarized as being manageable, meaningful, actionable, unambiguous, reliable, accurate, timely and predictive (Brotby, 2009)

To create quality metrics security teams should strive to:

1. Develop a set of metrics that are repeatable and automated where applicable
2. Create baselines or timelines from the repeatable metrics
3. Have actionable enough metrics to make decisions
4. Be meaningful for management decisions

Teams should constantly be asking what needs to be measured and why. If there is not a good answer to "why", the team should consider whether this would make a good metric. Could the metric be used with other metrics to produce an aggregate picture of an overall security control? Many organizations have multiple technologies to combat malware; often at the end point, mail gateway, the firewall and server. Each of these technologies can produce metrics that can be grouped or aggregated to produce a metric that can show insight into the organizations ability to combat malware.

2.2. Statistical Techniques

There are several commonly used techniques for analyzing data that can be applied to create IDS metrics. Mean, median, aggregation, standard deviation, grouping, cross sectional, time series, correlation matrix, quartile analysis and Statistical Process Control can each be leveraged to build meaningful security metrics offering visibility into large data sets. Many of these techniques can be used in conjunction with one another to build more complex and often more insightful metrics.

Tim Proffitt, tim@timproffitt.com

The mean, or average as it is commonly known, is a standard aggregation metric. The average is the easiest of these techniques to compute. Add the elements in the data set and divide by the number of elements in the set. It should be pointed out; averages can be a poor choice for highly variegated data sets as they can obscure hidden spikes that might be interesting. A data set containing the number of thousands of SYN connections per hour {10,10,10,10,10,10,10,10,10,10} has the same average as the data set {1,1,1,1,90,1,1,1,2,1}. The second data set has a significant deviation (90) that could show interesting activity that might otherwise have been missed if the averages technique was utilized to show this data set's activity.

The median of a data set is the number that separates the top half of the set from the bottom half. The data set's mean will highlight where half the elements are above and half the elements are below. Medians can help particularly with measuring performance. A median metric can aid IDS management in understanding performance or relevance. When a particular signature can be counted by number of instances fired, an analyst can rate his response based on whether the signature is above or below a calculated median.

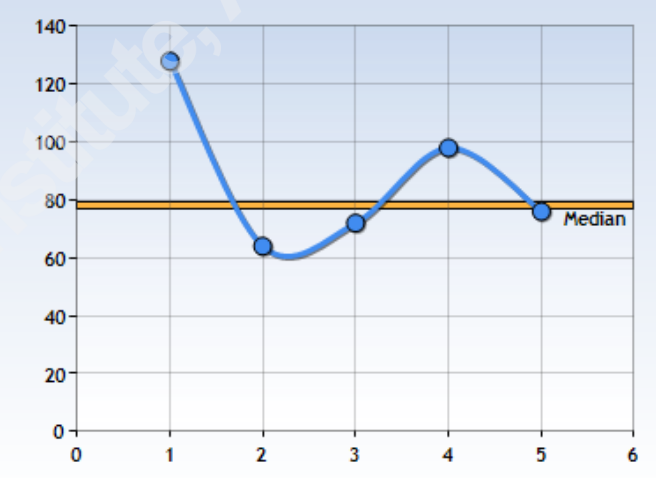


Figure 1: Median Statistical Example

Aggregation is a popular technique for consolidating records into some type of summary data. Common to aggregation statistics are sum, standard deviation, highest, lowest and count. In most cases aggregation involves averaging numeric values and counting nonnumeric values such as signature descriptions or severity. Highest and lowest aggregation values will allow analysis on top seen data elements and the least seen data elements. Aggregation is heavily used in technology metrics. Top 20 alerts, Total

number of High ranked signatures, and number of denied signatures are often generated for intrusion sensors dashboards.

Standard deviation measures the dispersion of a data set from the mean. This analysis technique can show if the data set is tightly clustered or wildly disperse. The smaller the standard deviation the more uniform the data set will be. A higher standard deviation would indicate an irregular pattern. One can calculate the standard deviation by first calculating the mean of the set. Then, for each element square the difference between the element and the mean. Adding up the squares, divide by the number of elements in the set to produce the variance. The variance provides a measure of dispersion and the root of the variance produces the standard deviation. [1] This type of statistical analysis could be used to show the types of TCP socket connection attempts to an organization's internet accessible assets and whether that could be considered normal.

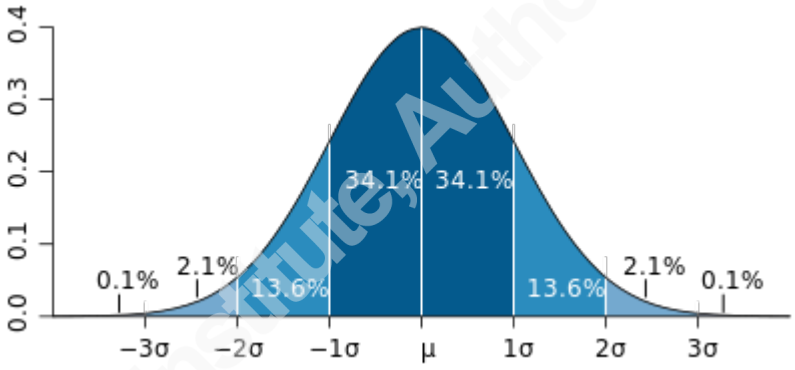


Figure 2: Example Standard deviation chart

Time series analysis is the technique of understanding how a data set has developed over time. This technique is a series of recordings, during regular intervals, of a data set over a period of time. After grouping and aggregating the data set within the desired interval the metric is sorted typically in ascending order. A time series technique can be a powerful tool in determining the current state of a technology versus how it has operated in the past.

> "Time series analysis is an essential tool in the security analyst's bag of tricks. It provides the foundation for other types of analysis. When combined with cross-

[1] If the reader is interested in further discussion of standard deviation they can visit (http://www.mathsisfun.com/data/standard-deviation.html).

sectional analysis and quartile analysis, it provides the basis for benchmarking" (Jaquith 2007).

The time series technique will generate metrics for sensor behavior over a specified reporting time window. Have the sensors alarmed on more events today than what was seen last year? Has the number of incidents investigated decreased in the last 6 months?

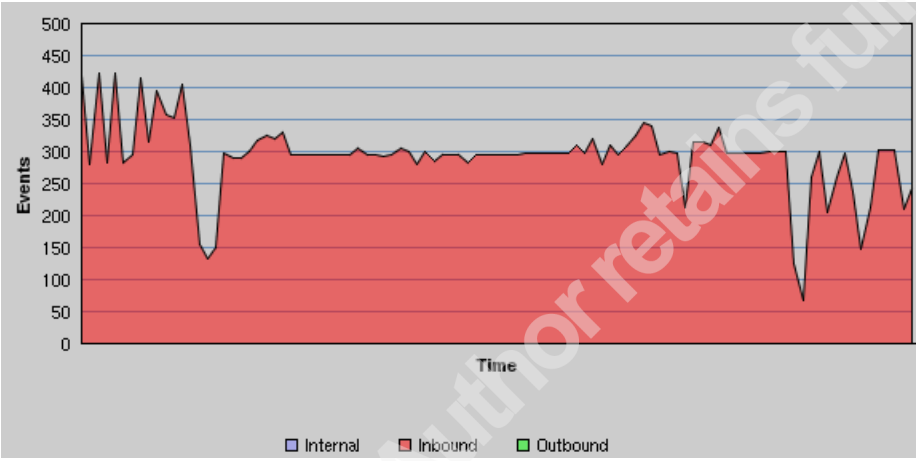


Figure 3: Example Time Series chart

Cross-sectional analysis is a technique that will show how an attribute in the data set will vary over a cross section of comparable data. The technique will take a point in time and compare "apples to apples". For example, analysts may want to measure current high, medium and low ranked signatures in a data set. One could draw a sample of 1,000 alarms randomly from that population (also known as a cross section of that data set), document their attack type profile, and calculate what percentage of that data set is categorized as attack signatures. Twenty percent of the sample could be categorized as attacking the organization. This cross-sectional sample provides one with a snapshot of that population, at that one point in time. Note that an analyst does not know based on one cross-sectional sample if the attack alarms are increasing or decreasing; it can only describe the current proportion and what it could mean to the organization.

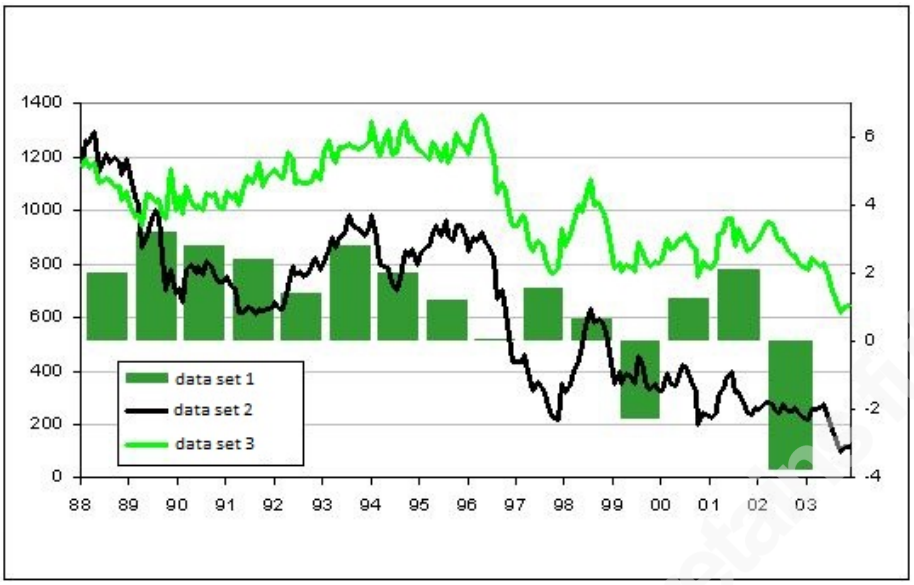


Figure 4: Cross Sectional Analysis Example

Quartile analysis shares several traits with cross-sectional analysis. Each requires a collection of attributes to examine and the analyst will identify a grouping and aggregation technique. In quartile analysis the aggregate is broken into quarters: first, second, third, fourth. The first quarter represents the top (or best) 25% of the aggregation. The fourth is the bottom 25%. By ranking each attribute into quartiles, the analyst gains an understanding of which section each item falls into. This type of analysis can be used to determine how well sensors are being managed, false positive acceptance rates, and aide in determining outliers (i.e., items in the first and fourth quartiles).

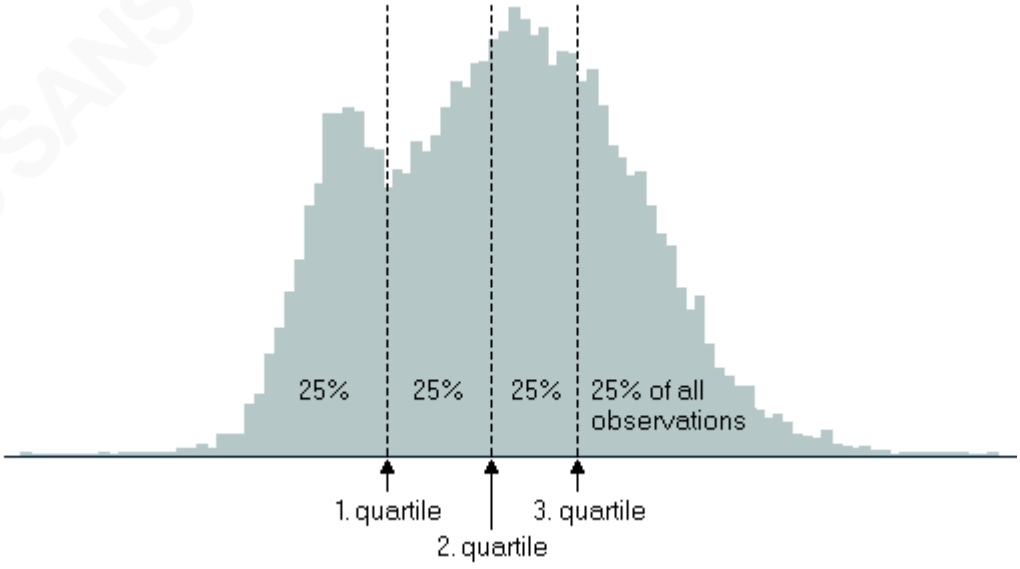


Figure 5: Quartile Analysis Example

Tim Proffitt, tim@timproffitt.com

Statistical process control is a technique that is used for determining scenarios outside normal operating patterns and thus establishing the concept of a baseline. As an example the security team is recording events from a tuned, managed sensor. A series of line graphs or histograms can be drawn to represent the data as a statistical distribution. This can be a picture of the behavior of the variation in the measurement that is being recorded. If a process is deemed as "stable" then the concept is that the sensor generated alarms in statistical control. If the distribution changes unpredictably over time, then the process is said to be out of control. The variation may be large or small but it is always present. Statistical process control can guide a team to the type of action that is appropriate for trying to improve the functioning of a process being monitored. When the data set is charted and falls outside the statistical upper control limit or below the lower control limit, than a security team can investigate what caused the change and implement changes to remediate what changed the sensor's statistics.

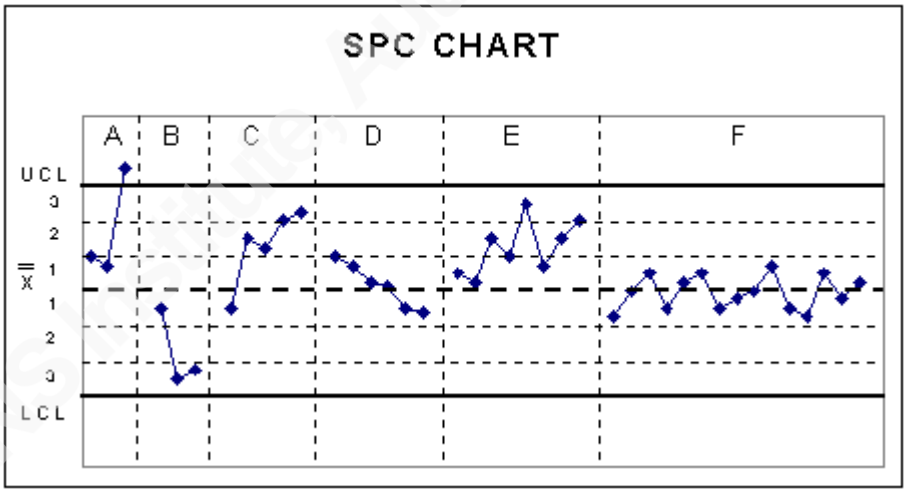


Figure 6: Statistical Process Control Chart Example

For organizations with extremely large data sets and complex reporting requirements, advanced tool sets from Informatica, Siebel analytics, Microsoft Business Intelligence or Information Builders may be warranted. For most organizations, the common Excel or Open Office spreadsheet can provide plenty of processes for carving through data sets to produce metrics. Fortunately for security teams, there are common statistical analysis tools than can make creating the analytics easier. For example, Microsoft Excel can calculate:

- Standard Deviation =STDDEV.P(),
- Absolute Deviations = AVEDEV
- Mean =AVG()
- Median =MEDIAN()
- Quartile analysis =QUARTILE.EXE()

By utilizing spreadsheets, analysts can automatically generate line graphs, scatter charts, bar graphs and most visual graphics needed to generate metrics.

2.3. How can metrics identify an incident?

An intrusion will typically start off with a series of unsuccessful attempts to compromise a host. Due to the current complexity of authentication systems, clandestine attempts at intrusion generally take considerable time before the system gets compromised or damaging change is affected to the system giving administrators a window of opportunity to proactively detect and prevent intrusion (Pillai). Therefore monitoring IDS patterns can be an effective way of identifying possible attacks. However, an IDS system can show an attack attempt, but often has no way to validate that it was successful. A host's logs may show a new administrative user being added, but has no way to determine if this was done maliciously. Yet the sequence of alarms, followed almost immediately by the creation of an admin account, is an event that shouts 'successful attack' quite clearly.

Cross-technology correlation between a host event and a monitored event can be a straight forward piece of evidence. Attacks against a host known to have a service or vulnerability present can be correlated into metrics. Does the organization have a system that must run in a vulnerable state? Any alarms against this system should be interesting, but when the alarms are coming from several sources and are multiplying an analyst should be notified. In the case of low and slow reconnaissance scans, many organizations will miss the activity. IDS sensors are typically not configured to escalate on "slow and low" single-packet probes, complex bounce or idle scans. If the signature event is not a critical or high ranking and the number of packets is only a few an hour, many times this will not stand out from the potentially millions of events generated for that day. A reconnaissance scan of 20 sessions a day may not meet the threshold for an analyst's

attention but after 90 days the metric can show 1800 sessions to a resource which may be interesting. Data collected over time can generate metrics that will show reconnaissance attacks from source and/or destinations when the metrics are built.

Baselines produce a powerful advantage from existing metrics. A baseline can be defined as a normal condition. A data set can then be measured, typically against the baseline to show deviations. Most baselines are established at a point in time and serve to continue to track measurement against the reference point. By utilizing baselines with sensor metrics, security teams can develop key performance indicators (KPI) or leading indicators to identify when an incident may be occurring. When creating a baseline for total signatures recorded over a time period, the metric with a baseline can quickly show where signature deviations have occurred. In figure 7 a baseline has been applied to an aggregate of signatures over time. An interesting indicator is present between 7/23/2013 and 7/24/2013 where the signature count was significantly higher that the baseline.

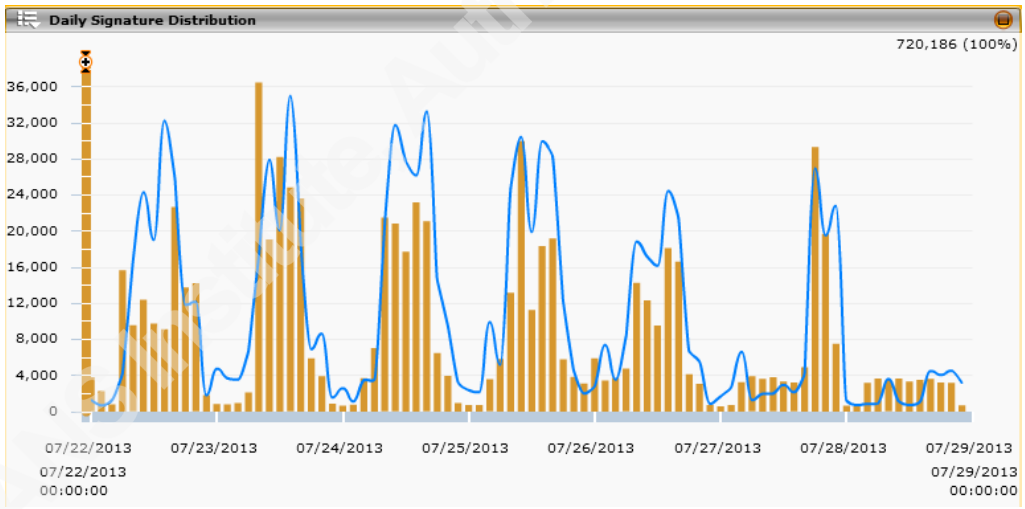


Figure 7: Baseline Example

2.4. What metrics should security teams build?

Metrics can be built that provide visibility into trends, configuration errors and risks to the organization. By utilizing statistical techniques teams can build several metrics that can help reduce risk to the organization by providing data to investigate. There are a number of simple metrics, utilizing the above statistical techniques, which a security team can easily build and put into practice will little effort. These simple metrics can not only provide insight into how the sensors are being utilized but can also lead to the building of more complex metrics as the program matures.

Tim Proffitt, tim@timproffitt.com

Top 20 Alarming Signatures (ordering, highest count,**)** - This aggregation metric can be used to baseline traffic patterns, identify DoS scenarios, highlight misconfigurations and will show the top talking signatures the sensor is processing. The top 20 signatures can be an insight into what the state of the network is in and possible tuning of the sensor.

Top 20 Alerts by Date Metric (ordering, highest count**)** - This aggregation metric can be used to highlight traffic patterns and potentially identify malicious activity during weekends or holidays. High numbers of failed authentications during the beginning of a work week could be considered normal but the same number of authentication failures on a weekend may be interesting.

Alerts by Source IP (aggregation, grouping**)** - Creating data sets by source IP can allow for identifying low and slow reconnaissance over time. Additionally, a sharp trending upward from a few IP addresses can identify a DoS attack. Similar to the Top 20 Alters by Date, this metric will allow teams to tune their sensors by eliminating false positives. Alarming or blocking the organization's vulnerability assessment scanners is most likely counterproductive and is wasting visibility into the true top source IP the analyst would be interested in.

Alerts by Destination IP (aggregation, grouping**) -** Data about destinations can highlight where outside entities are looking to or have found weakness. A high number of alerts, from differing sources, to a single destination should be very interesting to an analyst.

Alerts Categorized by Severity (aggregation, grouping) - By breaking the generated alerts up by severity this metric can allow an analyst to focus on the more risky events first and remediate lower risked events when applicable. Reconnaissance alarms can be a lower ranked severity and can be displayed when sorted from the higher categories.

Number of Alerts by Signature (aggregation, grouping) - By creating a metric for the number of signatures by a specific alarm, an analyst can identify brute force attacks against a single or multiple destinations. This metric can also be used for tuning IDS signatures to reduce the amount of noise they are recording.

Alerts by Source Port (aggregation, grouping) - Collecting data by source port can highlight attacks specifically designed to compromise a specific vulnerability. Analyst

should be interested if 75% of the alarms are packets destined for port 1337. This metric could be showing activity to services that are not allowed by the firewall. An attacker may have found a vulnerability from a misconfiguration.

Alerts by Destination Port (aggregation, grouping) - Destination port metrics are similar to source port but will provide data on systems that may be compromised. An analyst may see a large amount of signatures firing for events destined for port 51001. Similar to the metric for Alerts by Source Port, destination port metrics can show attacks to services. There has been a significant increase in the amount of outbound beaconing from compromised hosts in an effort to defeat inbound defenses. Metrics created from identified incidents can uncover network sessions sharing common characteristics that recur at regular intervals. Teams can seek additional indicators of malware infection to support proactive incident detection as well as to supplement incident response efforts (Balland 2008).

Source IP by Country (aggregation, count, sort) - Geolocation data can be a very powerful indicator if used in conjunction with other metrics data. If an organization does not conduct business outside the United States, an analyst should be interested in any established conversations from a source IP from outside the country. With the rise of the advance persistence threat (ATP), teams can create metrics for connections from countries that have known nation state APT programs.

Total Number of alarms by hour / day (aggregation**,** sum, sort**)** - Creating baselines for the number of alarms over a time period can highlight patterns of an attack. Activity levels registered during off hours can be interesting for the security team. More activity on the weekends than during the traditional 9am to 5pm shift can be an interesting indicator. Using statistical techniques against the total number of alarms by hour will yield data that paints a picture around how the sensors are operating. This metric can also be used as a baseline for the creation of additional metrics to show activity outside of normal operating parameters.

Number of denied conversations (aggregation, count) **-** Denied connections can yield information about several scenarios. Using baseline data this metric can show whether the organization's resources currently under attack with a denial of service. The sensors may be recording a large number of denied connections to a destination that does

not exist. A high number of denied connections can show a misconfiguration, provide indicators of reconnaissance or a brute force attack. By grouping denied conversations and crafting a baseline, metrics can show when these scenarios have initiated.

Number of failed login attempts (aggregation, count) - Metrics on failed logons over time can show indicators of password cracking attempts, policy issues and denial of service attempts. An organization can expect a reasonable number of failed logons due to human nature and a proliferation of devices but when that baseline is doubled in a day that metric can be interesting to an analyst.

IDS Throughput (aggregation, count, quartile) - This metric defines the level of traffic up to which the IDS performs without dropping any packets. Is the IDS performing as it should? Does a sensor need to be replaced or load balanced to meet peak levels of traffic? Can the IDS inspect all traffic it is expected to during a high traffic attack? Dropped packets mean conversations can go unnoticed and place the organization at risk. Teams that understand how their infrastructure is performing will have better information to act on and therefore reduce risk.

Daily Sensor Baseline (quartile, statistical process control) – The sensor baseline metric for each type of sensor (over time) is a visual intended to highlight the sensor operating outside the normal historical patterns. Data sets that are on the highest quartile or outside SPC should be interesting to an analyst. The data points outside the sensor baseline can indicate denial of service, reconnaissance or an attack of some sort.

Signatures with Invalid Destinations Baseline (Aggregate, Time Series) – This baseline offers visibility into conversations seen by the organizations sensors that are not specifically destined for the organization's assets. Many organizations will craft custom signatures to fire when traffic is seen outside what is expected for conversations incoming or exiting the firewall. A metric with a baseline for invalid destinations can provide visibility into an indicator of compromise.

2.5. What metrics should security teams build for management?

Management teams need quality metrics. Consistently, security departments are taking directions from management teams that may have very little knowledge about the attacks the environment endures on a daily basis. In some cases IDS metrics will need to

be combined with qualitative or quantitative data, such as the number of investigations assigned to a security team, to create a better picture of the security stance in the organization. By creating metrics designed for this audience, the security team can equip leaders to make informed decisions about the security of organization.

Depth of System's Detection Capability. A detection capability metric can be defined as the number of attack signature patterns and/or behavior models known to a sensor technology. This metric can indicate if the IDS infrastructure is identifying all that it is expected to identify. This can show data about the team missing attacks because the IDS capability is lacking. The decision could be made to investigate newer technologies to increase the visibility expectations. Additionally, insight can be gained into how the sensor technology is currently providing security and can be contrasted with other competing products can be shown. Depth of Systems Detection Capability can also show where the vendor will issue an update and time for the corporation to implement the vendor updates.

False Negative Ratio. The false negative ratio metric is the ratio of successful attacks not detected by the IDS. "The most basic and commonly used metrics are true positive rate (TP), which is the probability that the IDS outputs an alarm when there is an intrusion, and false positive rate (FP), which is the probability that the IDS outputs an alarm when there is no intrusion" (Gu, 2006). Similar to the false positive rate, the security team may be reacting to incidents not captured by the sensor infrastructure. The organization may have other security technologies in place such as anti-malware, firewalls, data leak prevention, application whitelisting or a honeypot that has revealed an intrusion. As part of incident response tasks, security teams can research whether the sensors were capable of identifying or producing an alert for the intrusion as it crossed the sensor. This data can be collected and be used to generate the False Negative Ratio. This metric can visualize if the current IDS is the correct solution for the environment, if the team is utilizing the technology correctly, or if more security staff should be monitoring the sensor data.

Reliability of Attack Detection. The reliability metric can be defined as the ratio of false positives to total alarms raised. An analyst may be researching incidents to

determine later that the event was a false positive. The data can be collected from the team's ticketing system and used to produce a metric. This can help identify if the sensor infrastructure is to undergo a reconfiguring exercise, determine if the IDS solution is correct solution for the environment or have staff be allocated to perform sensor tuning to an acceptable level.

Compromise Cost Analysis. The compromise metric is the ability to report the extent of damage and compromise due to intrusions identified by the security program. Time is being spent on remediating successful intrusions and a monetary figure could be calculated to aide management in decision making. This monetary figure could be shifted elsewhere to better protect the organization, possibly with intrusion detection.

Intrusion Timeliness. Intrusion timeliness metric is the average or maximal time between an intrusion's occurrence and it being reported. The organization's response time could be in hours or days after an intrusion is successful. The timeframe may be acceptable to the organization or could show a deficiency. The intrusion timeliness metric can show an increasing or decreasing picture for the organization. This metric can show management whether more staff is warranted, the incident response program is operating at an intended level, and are intrusions being reported within a timely manner.

3. Conclusion

As security practitioners, analysts can create metrics from constantly generated data from security systems including intrusion detection sensors. Properly crafted metrics can be a powerful tool in a security team's arsenal and should be leveraged to the organization's advantage. Many metrics can be created beyond what was covered and organizations should strive to continually find new ways to measure data and improve processes. Analysis of the security metrics that are produced can be a critical component for the security team's success. Creating a metrics program is a step in maturing a security program and reducing risk for the organization. "It is impossible to eliminate risk. Instead we must learn to manage risk and to do that we need to measure it, and we need to decide how much we are willing to accept and how much we can afford to mitigate" (Hayden 2010).

Tim Proffitt, tim@timproffitt.com

4. References

Greitzer Frank L. (2005). Methodology, Metrics and Measures for Testing and Evaluation of Intelligence Analysis Tools. Retrieved from www.pnl.gov/coginformatics/media/pdf/iamethodology_paper.pdf

Pillsi, Krish. (2010). Network Intrusion Detection System – A Novel Approach. Department of Computer Science, Lock Haven University of Pennsylvania Lock Haven, PA. Retrieved from http://www.iiis.org/CDs2012/CD2012SCI/IMETI_2012/PapersPdf/FA702DM.pdf

Security Onion. Security Onion Wiki 2010 Online. Retrieved from http://code.google.com/p/security-onion/w/list

Greene Jr., Richard. (2001). Using Snort v1.8 With SnortSnarf on a RedHat Linux System. SANS Reading Room. Retrieved from http://sans.org/rr

Brotby, Krag. W. (2009). Information Security Management Metrics. A Definitive Guide to Effective Security Monitoring and Measurement. Auerbach Publications. New York.

Beale, Jay, Baker, Andrew, Caswell, Brian, Poor, Mike, Foster, James, Beale, Jay, Baker, Andrew, Esler, Joel, Kohlenberg, Toby, Northcutt, Stephen, Rash, Michael, Orebaugh, Angela, & Turnbull, James. (2004). Snort 2.1 Intrusion Detection, Second Edition. Syngress Media Inc.

Hayden, Lance. PhD. (2010). IT Security Metrics. A Practical Framework for Measuring Security and Protecting Data. McGraw-Hill Companies. New York.

Jaquith, Andrew. (2007). Security Metrics. Replacing Fear, Uncertainty and Doubt. Addison-Wesley. New York.

B.C. McDaniel. (2011). Statistical Process Control: Introduction and Background. Retrieved from https://www.moresteam.com/toolbox/statistical-process-control-spc.cfm

Balland, Peter. (2008). Analysis of Network Beaconing Activity for Incident Response. FloCon2008. Retrieved from Site:www.cert.org/flocon/2008/presentations/balland_flocon2008.pdf

Communication Network and Information Security Research Group. (2012). Sriwijaya University, Palembang, Indonesia. Retrieved From Http://comnets.unsri.ac.id/?p=154

Guofei Gu, Prahlad Fogla, David Dagon, Wenke Lee. (2006). An Information-Theoretic Measure of Intrusion Detection Capability. College of Computing, Georgia Institute of Technology, Atlanta GA. Retrieved from http://www-static.cc.gatech.edu/~guofei/paper/Gu_asiaccs06_cid.pdf

Tim Proffitt, tim@timproffitt.com

5. Appendix A

Basic Analysis and Security Engine (BASE) (http://sourceforge.net/projects/secureideas),

SGUIL (http://sourceforge.net/projects/SGUIL/)

OSSIM (http://communities.alienvault.com/),

PLACID (http://freecode.com/projects/placid)

ARK. Website. https://alienvault.bloomfire.com/series/1471/posts/556521 presentation.

Tim Proffitt, tim@timproffitt.com

6. Appendix B

Snort's IDS system is designed to run in different modes. In sniffer mode the program will read packets off the network medium and display them to standard out. While in packet logger mode the IDS will analyze the packets and log them to local storage for later analysis. In intrusion detection mode the IDS will monitor traffic and analyze it against a set of defined snort rules and then perform actions based upon specific rules. The generic built-in metrics that Snort provides is through rule profiling. Rule profiling allows for rudimentary statistics (see table 1). Although these statistics can be useful, they are not going to help a security team reduce risk or protect an organization from an attack. These metrics are more for maintenance of the sensor.

Checks	# of times snort checks for rule options after the engine completes analysis
Matches	# of times that snort finds traffic matching all rule options
No Matches	# of times that snort finds no traffic matching all rule options
Average Ticks	AVG time that snort takes to check each packet against the rule
Average Ticks Per Match	AVG time that snort takes to check each packet that matches all rule options
Average Ticks Per No Match	AVG time that snort takes to check each packet that did not generate an event
Total Ticks	Rule responsible for the most processing time.

The base snort system is a bit lacking in archived reporting and analysis capabilities compared to commercial products. To meet these needs the open source community has built additional components on top of the traditional snort engine that will allow security teams to leverage graphics, trending and reporting.

6.1. SGUIL

SGUIL (http://SGUIL.sourceforge.net/), created by Bamm Visscher has been referred to as "The" analyst console for network security monitoring. It is the analyst's right hand, providing visibility into the event data being collected and the context to validate the detection (Security Onion Wiki 2010). SGUIL started off as the "Snort GUI for Lamers" and expands beyond a simple snort GUI in that it also integrates other technologies into the recording of data for use by the analyst. One example of this would include fulltime, full packet capture. Written in Tool Command Language (TCL), SQUIL is a very well designed engine for analyzing packets.

Tim Proffitt, tim@timproffitt.com

The SGUIL tool can be built on both a Linux and Windows platform. The product is designed to record packets for analysis and will need a database framework installed. The database configuration necessary to run SGUIL will depend on the amount of traffic to monitor and the length it will be retained. The Tcltls package is leveraged for securing communications between the server and client consoles. SGUIL contains some very useful analysis packages such as the Security Analyst Network Connection Profiler (SANCP) which can collect statistical data, utilizes scripts to log all the packets in PCAP format, and uses tcpflow and p0f to get session data. The Passive Asset Detection System (PADS) is used to collect banners on host services and uses Wireshark for GUI based packet analysis.

The SGUIL tool is a system consisting of three components: a sensor, server, and client. A deployment can be built for all three components on a single platform or each utilizing separate platforms. The SGUIL sensor, running with Snort, will be placed at the proper location for traffic visibility while the server and console components can be placed on any internal segments. The Security Onion (http://securityonion.blogspot.com/) is a prebuilt Linux distribution for network security monitoring. It's based on Ubuntu and contains Snort, Suricata, Bro, SGUIL, Squert, Snorby, Xplico, Network Miner, and many other security tools. The Onion setup wizard allows for quick building of the SGUIL components without dealing with collecting and compiling of dependent packages needed for SGUIL.

The SGUIL dashboard allows an analyst to “pivot” directly from an alert into a packet capture from Wireshark or a transcript of the full session that triggered the alert. The analyst can view all of the associated traffic and answer important questions about what transpired during a conversation. Additionally, SGUIL allows the analyst to query all packets captured, not just those involved with an alert. Traffic can be correlated that may not have triggered any alerts but still could be associated with malicious activity. Lastly, SGUIL allows the analyst to conduct reverse DNS and whois lookups of IP addresses associated with alerts.

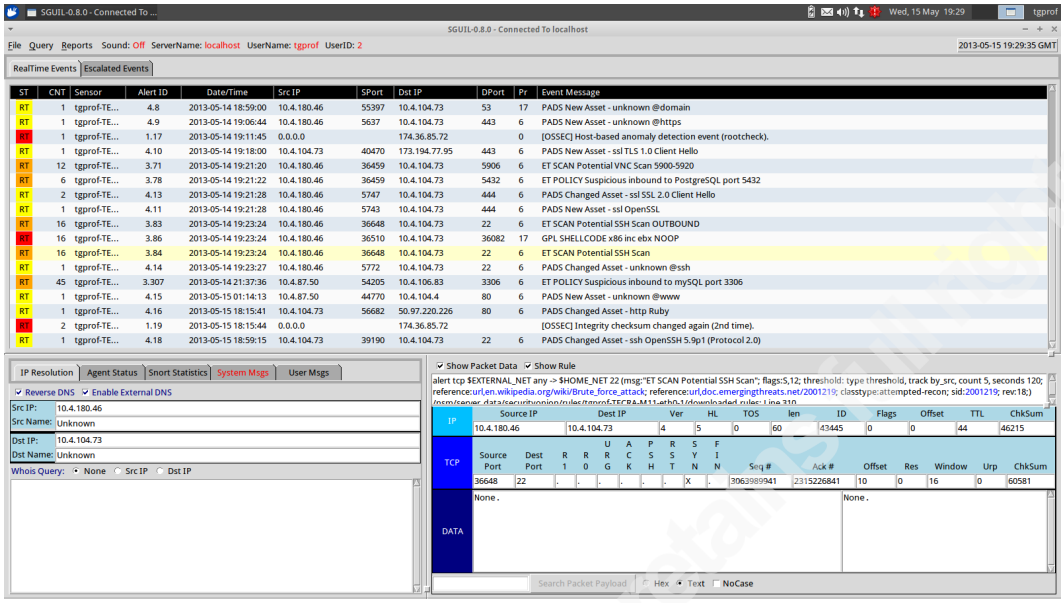


Figure 8: SGUIL dashboard

Collecting data for metrics can be accomplished down several paths. The reporting menu inside SGUIL allows for exporting of crafted events into text files. This requires upfront configuring of reporting elements. SQueRT, packaged with the Security Onion, is tailored to provide metrics data from SGUIL events. SQueRT efficiently reads the data collected by SGUIL and provides daily metrics on source IP, destination IP, source ports, signatures and categories of severity. Additionally there is an archive feature that will allow an analyst to review previous days. This reporting tool will automatically generate bar and line graphs for top sources and destinations.

6.2. B.A.S.E.

Basic Analysis and Security Engine (BASE) is a browser based dashboard for the analysis of the Snort IDS and precedes the older Analysis Console for Intrusion Databases (ACID) codebase. The ACID interface was first started as Carnegie Mellon college project written by a student. BASE, a descendant of the ACID code, picked up where the original author left off. BASE added a bunch of new features, made the interface easy to use and provides for a highly functional GUI. This addition to snort remains the most popular Snort GUI interface with over 215,000 downloads (Communication Network and Information Security Research Group 2010).

Tim Proffitt, tim@timproffitt.com

The BASE application, originally designed for RedHat / CentOS, can be compiled and used on most flavors of Linux. Alternatively, BASE can run on Windows with IIS and MSSQL. The BASE on Linux depends on several installed packages such as PCRE, MySQL, PHP, JPGraph and ADODB. The last release for this package is BASE 1.4.4 (dawn) from 2009 and can be found at SourceForge.net. A build document for this installation of BASE can be found at http://www.internetsecurityguru.com/documents.

Alternatively, the EasyIDS (www.skynet-solutions.net) provides an IDS distribution ISO based upon CentOS, Snort, Barnyard Oinkmaster and BASE to allow organizations to quickly deploy a monitoring system with only minor modifications.

The BASE frontend is accessed via a remote browser. The analysis features of the tool allow for an analyst to quickly view alerts based on daily, hourly, most recent, last destination, frequency, and recent unique.

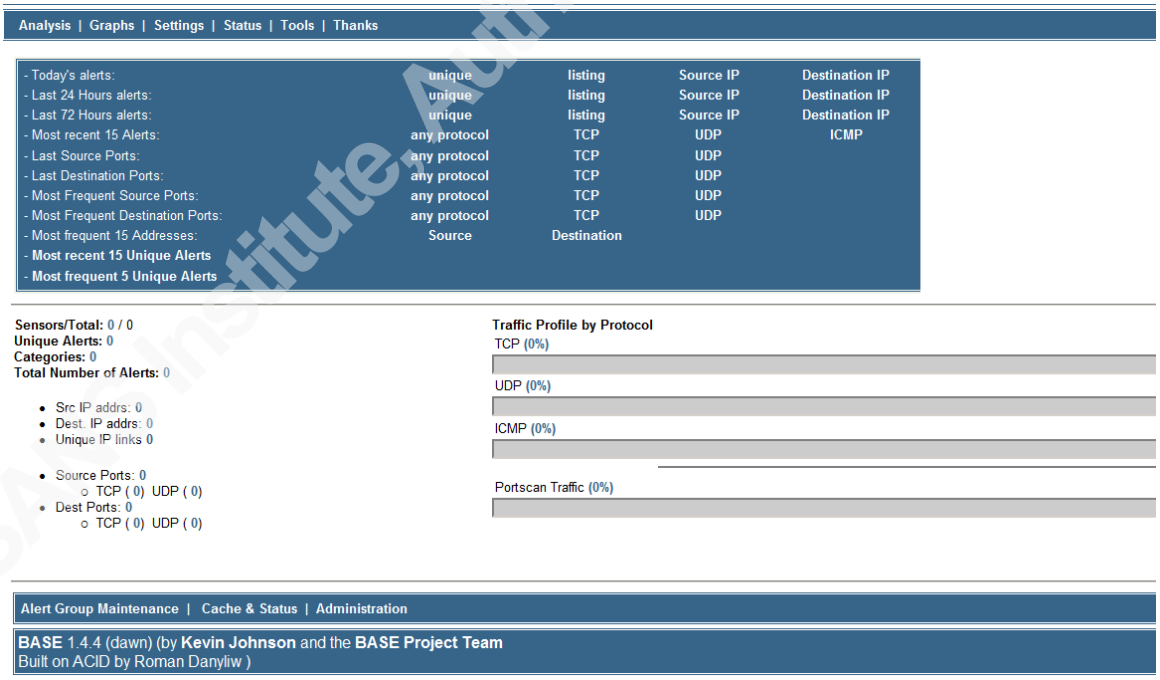


Figure 9: BASE dashboard

Metrics can be generated in BASE by using the Graphs menu. Graphs allows for the charting of network, Snort and system data. The Snort performance graphs allows for

charting dropped packets, alerts per second, session events, open sessions, frag events, average bytes per second and more.

6.3. OSSIM

Open Source Security Information Management (OSSIM) is maintained by AlienVault (http://communities.alienvault.com/). OSSIM, designed to be a security event manager, can take the logs from Snort, among other things, and display them in a great looking interface. Additionally the tool integrates with many other tools such as p0f, arpwatch, nessus, ntop, nagios and more. The tool, released in 2003 has already surpassed 200,000 downloads. OSSIM provides features that an analyst may need from a SIEM offering event collection, normalization, and correlation. Unique among the products in this space, AlienVault continues to provide ongoing development for OSSIM.

The OSSIM tool is a Linux based tool. Originally designed for RedHat / CentOS, the product can be used by most flavors of Linux. The OSSIM v4.2 ISO can be used to deploy the management and sensors quickly resulting in the complete experience of OSSIM in a simple deployment. Alternatively, Alien Vault has provided a flavor of OSSIM running in the Amazon EC3 cloud and in this configuration provides for a 64-bit Amazon Machine Image.

The intrusion detection engine that currently deploys with OSSIM is the SNORT open source intrusion detection engine. Signature 'hits' from the Snort IDS system in OSSIM generate SIEM events just like log events from system logs are normalized into source and destination, protocol, and more. These can be searched and pivoted on just as with any regular SIEM event. The IDS information is consumed into the SIEM via device plugins and message SID for each event type.

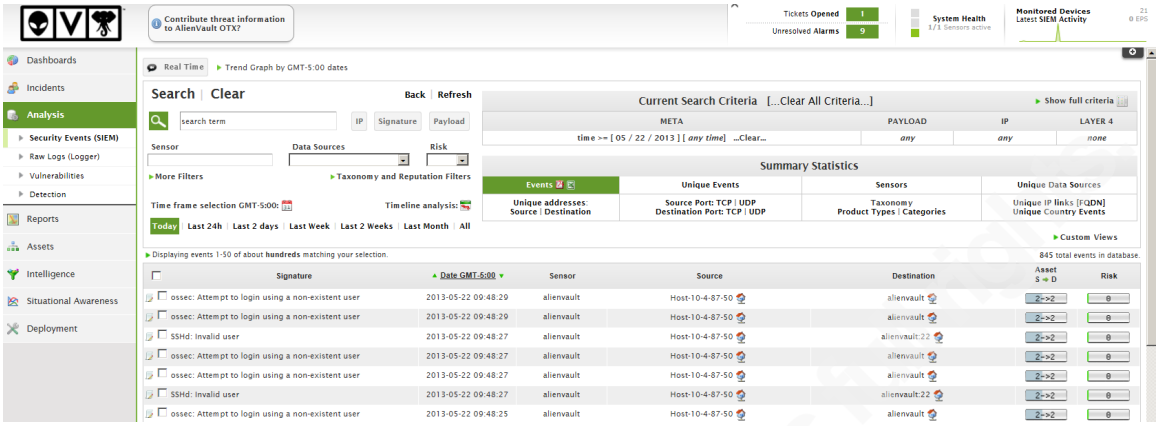


Figure 10: OSSIM dashboard

The OSSIM engine provides a robust reporting section. Being that OSSIM is primarily a SEM, the feature set is greater than what is seen from most Snort graphical front ends. By default OSSIM provides reporting for top 10 alarms, assets, availability, compliance such as PCI or ISO27001, geographical locations, metrics by time frame, user activity, and tickets from the built in ticketing system. The reporting engine allows for the dynamic creation of a PDF or sending of an email containing the relevant data in PDF format.

SIEM Events - Top 15 Events from: 2013-04-22 to: 2013-05-22

Event	Occurrences
ossec: Integrity checksum changed.	335
ossec: Attempt to login using a non-existent user	175
SSHd: Invalid user	92
ossec: Integrity checksum changed again (2nd time).	59
ossec: SSH insecure connection attempt (scan).	48
SSHd: Did not receive identification string	48
Host operating system change	25
ossec: Host-based anomaly detection event (rootcheck).	21
ossec: SSHD brute force trying to get access to the system.	9
ossec: Login session opened.	5
ossec: Login session closed.	4
Host service change	3
ossec: Ossec server started.	3
ossec: Integrity checksum changed again (3rd time).	2
ossec: Physical root login.	2

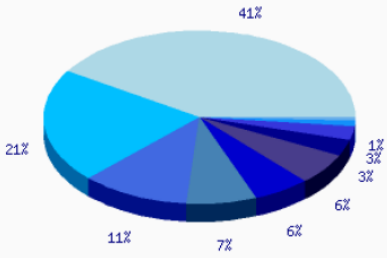


Figure 11: OSSIM Top N Report

Tim Proffitt, tim@timproffitt.com

6.4. Snorby

Snorby is a front end web application scripted in Ruby on Rails. It is designed to use Web 2.0 effects and rendering to provide an analyst with a very sharp and beautifully functioning tool. While it has many of the features of BASE and similar to SGUIL, the tool is extremely easy to deploy, looks fantastic and functions as an alert browser. Additionally, Snorby can be utilized for any application logs that are output in the unified2 format. An advantage of this tool is that it can integrate with the OpenFPC project (www.openfpc.org) to collect full packet captures. The tool can provide an analyst the ability to not only view the IDS alert, but also to view the alerts in context with the rest of the packet flow on the network.

Based on Linux the tool has a handful of dependent packages such as GIT, Ruby, RAILs, ImageMagick, MySQL, Barnyard and Wkhtmltopdf. Additionally, Snorby allows the analysts to choose which IDS solution will be utilized (Snort or Suricata). The tool can be deployed as a unified sensor and server console or can be installed as a server with multiple sensors. Adding a sensor is as simple as standing up a new snort instance to log to Snorby's barnyard database. Similar to the OSSIM offering, Snorby has a cloud offering that can be found at www.threatstack.com. This would allow an organization to easily point their snort logs to a Snorby instance in the cloud.

The Snorby interface provides several mechanisms to quickly see IDS metrics based on timeframe. The analyst can see data on events by sensor, severity over time, protocol counts, signatures matched, sources and destinations.

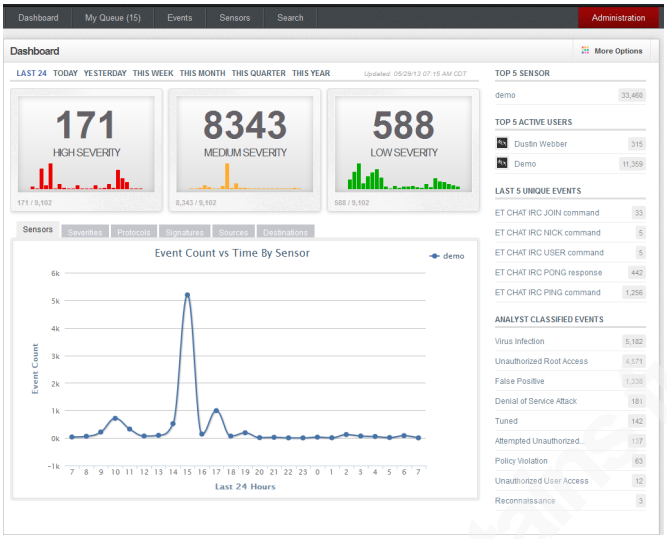


Figure 12: Snorby Dashboard

The ability to generate metrics from Snorby is a rich feature. The interface allows for a PDF to be generated for the context the analyst is searching. Daily, Weekly, and Monthly reports can be configured to be summarized, generate a PDF and delivered over SMTP and in XML format.

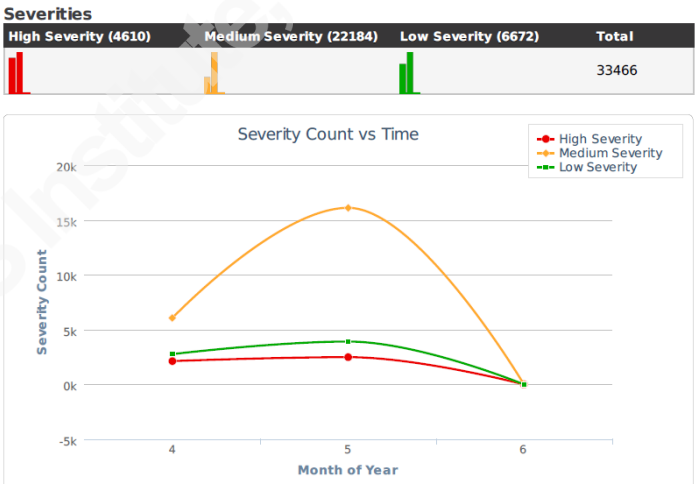


Figure 13: Snorby PDF extract

6.5. redBorder

redBorder is a new Ruby on Rails based Open Source project around Snort (http://redborder.net/products/ips/). The redBorder IPS is a self-contained Linux distribution. The tool has several capabilities such as event view and storage, hierarchical management of multiple sensor configurations, web front end based of Snorby, powerful

rule management system, updating of feeds and configurations, SNMP monitoring for generic system capabilities, and advanced user management with roles and auditing.

The redBorder IPS manager allows the administrator to centrally manage the rules of an enterprise deployment. The manager allows for hierarchical rule management, nesting of rules, action configuration, versioning with rollback capability, rule searching, multiple rule feed updates and sensor upgrades based on a schedule. The SNMP agent provided by RedBorder has the ability to monitor sensor state (CPU, RAM, throughput, etc.) and trap to the redBorder manager as well as the organization's other monitoring consoles. Each sensor is able to analyze its status and load.

redBorder is unique in the number of users that are utilized while operating the tool. In addition to the traditional root account, an administrator account is used to access the redBorder configuration. The "redBorder" account is used for tasks inside manager console, while "manager" and "sensor" are used for registering sensors and configuring communication with the sensors. Custom users created for use in the manager console can be granted read, manage and none.

Since the redBorder engine is based off of the Snorby interface, to generate metrics the operator has several options available in the menu. Top alerts, sessions, sources, destinations, signatures, and packets dropped are available out of the default configuration. Each of these views can be exported for the creation of metrics.

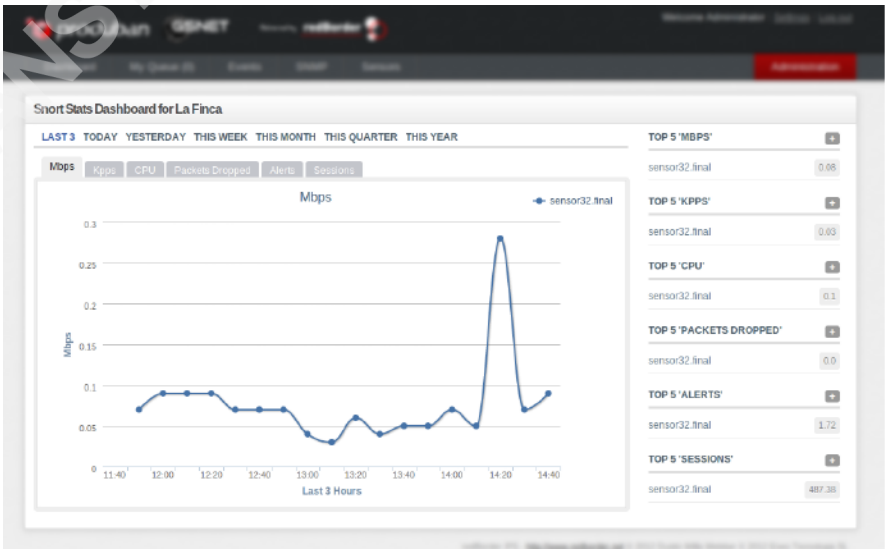
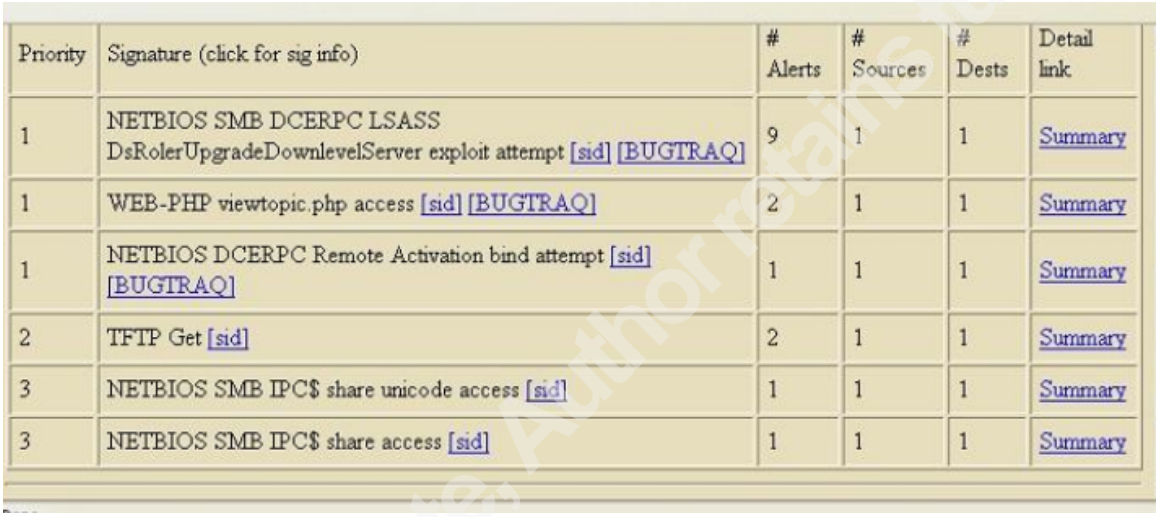


Figure 14: RedBorder Dashboard

6.6. SnortSnarf

SnortSnarf (http://sourceforge.net/projects/snortsnarf/) is a pearl script to take alert log files created by the Snort engine and produces sanitized reporting data. This script can be run via a cron job or scheduler at regular intervals to produce a convenient HTML breakout of alert data that can be used for metrics. Since SnortSnarf is written in pearl it will run on Windows or Unix flavors.



Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests	Detail link
1	NETBIOS SMB DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt [sid] [BUGTRAQ]	9	1	1	Summary
1	WEB-PHP viewtopic.php access [sid] [BUGTRAQ]	2	1	1	Summary
1	NETBIOS DCERPC Remote Activation bind attempt [sid] [BUGTRAQ]	1	1	1	Summary
2	TFTP Get [sid]	2	1	1	Summary
3	NETBIOS SMB IPC$ share unicode access [sid]	1	1	1	Summary
3	NETBIOS SMB IPC$ share access [sid]	1	1	1	Summary

Figure 15: SnortSnarf Dashboard

The interface is such that potential problems can be easily analyzed. The Snort alert logs and system log files will provide data of what was possibly compromised. When a security incident occurs the research link points will allow the analyst to start looking for ways to prevent further incursions. This further research and analysis SnortSnarf data will help provide enough information to make a recovery plan. The analysis should help identify where a defense in depth plan failed. With this knowledge of what failed, where it failed and how it failed, plans can be made on how to prevent unauthorized access in the future (Greene, 2001). Creating metrics data with the SnortSnarf engine is in the HTML files themselves.