# Bad Neighborhoods Near Hostile Addresses

Scott Campbell

**Abstract**— Production security analysis often develop intuition about hostile network activity based on previous experience with traffic from related networks. This paper investigates the relationship between suspected hostile IP addresses, and the behavior of address spaces immediately surrounding them. By using network scan detection as a judge of intent, the behavioral characteristics of subnets local to identified scanners can be seen to exhibit activity which can not be explained away by random characteristics or secondary effects. When intrusion detection systems take into consideration the previous behavior of hosts within a near-neighborhood address, a meaningful increase in hostile behavior detection is achieved.

**Keywords-** Internet Background Radiation, Intrusion Detection, Network Security.

- *Scott Campbell is a Computer Scientist at the National Energy Research Scientific Computing Center (NERSC), Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley CA 94720. E-mail: scampbell@lbl.gov.*

## 1 Introduction

Recently advances in the way that network connection information is analyzed by intrusion detection algorithms have received considerable attention from the production computer security and research communities. Historically this analysis has been based on the interpretation of a single IP address' behavior over some length of time. In this paper we will look at extending the usefulness of information derived from a single hostile host by inferring that the hosts local subnet can also be considered more hostile.

To examine this we need to address a number of issues, including the how hostile hosts are identified. While there are a number of characteristics that can be looked at such as connection volume, timing, or payload, we are asserting that a host is hostile if it is determined to be scanning the local address space.

While the notion of a network scan is implicitly defined within it's context of use, on closer examination there is some degree of ambiguity attached to what makes it up. Differentiating between active scanning, radiation, worm traffic, interactive and non-interactive traffic may seem somewhat arbitrary, but it is possible for a 'network scan' to be composed of one or more of these elements. This becomes important when it is noted that each of these network traffic types has it's own model and connection characteristics. This will be described in more detail in § 2.

The problem of defining what exactly is meant by a scan, as well as how one goes about identifying and parameterizing such an event has been gone over in some detail in Jung [8] and Staniford *et. al.* [23]. In order to provide the best coverage for scan based events, both the native scan detection algorithm found in Bro [18] as well as the TRW (Threshold Random Walk) [7] are used to gather data. Because the native algorithm has advantages in terms of simplicity as well as general intuitive understanding, it will be used to illustrate our point.

The number that describes the connection threshold defining when a host is considered a scanner, can be modified based on the historical behavior of other hosts within the scanners local subnet. Changes made to this threshold are directly proportional to the number of hosts identified as scanners within the time window used for learning. For example, if within a given subnet three hosts have each been identified as scanners two times during the learning period, the threshold for scan determination might be lowered by six for address' contained within the subnet. In the case of the prototype software described in § 5, the number of *failed* connections determine how a scanner is identified, so the number of failed connections required would be reduced by six. The significance of this is that typical benign activity from non-hostile hosts is unlikely to be interfered with since for such activities the number of failed connections is typically quite small.

For this paper, connection data was generated by converting Bro connection logs from their original ascii format to a binary representation. From there they were then converted into native root format and indicies were generated. This was then fed into a Root [22] instance which has been modified to use FastBit [11] data indicies. This setup allows for simple programs to be written which can run dynamic 'what if' scenarios against a large set of data, while at the same time running faster than traditional pacp based file applications. For similar analysis tools, we have worked with data sets on the order of a billion records. Additionally, privacy issues are touched on since there is no additional data beyond what is clearly defined in the schema. This will be further described in the 'Dataset and Method' section.

Since large numbers of hosts on the Internet sit behind some sort of NAT infrastructure [15][14] there are several issues that need to be addressed. These include the possibility that a single host can exhibit undue influence over other hosts who share it's translation address, and that implicit assumption about 1:1 mapping between the source of a connection and the IP describing it built into the scan detection mechanism. We expect to see this behavior as Casado et all in [4] suggested that addresses sitting behind NAT mechanisms constituted on the order of 50% of observed Code Red II traffic from the private 192.168/16 address space. A similar problem to NAT is the 'DHCP Effect described by Moore in [26]. Like NAT, DHCP abstracts the real source IP from the one that is permanently associated with an observed connection. In this case the drift tends to be temporal – there is no real guarantee that a host will have the same IP day after day. When combined with NAT, this problem is simply compounded. This problem will be picked up and addressed in § 4.4 .

Besides NAT effects, false negatives can be created by arbitrarily lowering the threshold which identifies scanners for a given subnet. Given that our analysis is based on lowering scanning thresholds, we need to show that the newly located addresses are not purely an artifact of lowering the threshold. This problem is addressed in § 4.1 and 4.2 where initial results are presented which show that discovered hosts and the networks around them exhibit characteristics that are not described by random behavior.

The rest of this paper is structured as follows. In § 2, work relating to our research is discussed. In § 3, the initial source of network data is described, as well as a brief overview of the Root toolkits use of Fastbit data indexing in the data analysis process. In § 4 an initial analysis of data is presented. This looks at the distribution of scanning hosts vs. TRW results, the variation in radiation from identified networks, a discussion of NAT as a possible explanation for the observed results, and the calculation of the most effective granularity for the subnet attached to the hostile address. In § 5 operational experience is described, and in § 6, the conclusion and future work is presented.

## 2 Related Work

Although a general description of the work falls under the arena of network intrusion detection, the background and empirical components of the data analysis are more diverse. A greater emphasis on traffic characterization and modeling is being made here than would otherwise be expected, since there seems to be little data on characterizing *active* scanning – that is scanning purposefully directed into an address range in a non-random or accidental manner.

Scan detection provides our foundation for determining malicious intent. For the native Bro [18] scan algorithm (BSA), a host is allowed to have a set number of *failed* connection attempts before they are identified as a scanner. In addition, to the connection failure threshold, there are a number of filters that can be applied to this determination such as destination port or service, or source host/network. These filters are used to ignore services that are not interesting from a production security perspective, and to define lists of addresses that are immune to being identified as scanners. The default number of failed connections from a source address is set to 100, and the list of ignored services is defined to be 'ident'. Changes to the default configuration are described in the Dataset section.

The Threshold Random Walk (TRW) [7] algorithm works by using an 'oracle' to determine if a connection will succeed or fail. A successfully completed connection drives a random walk upwards, a failure to connect drives it downwards. By modeling the benign traffic as having a different (higher) probability of success than attack traffic, TRW can then make a decision regarding the likelihood that a particular series of connection attempts from a given host reflect benign or attack activity, based on how far the random walk deviates above or below the origin. [25]

A principal tool for the investigation of traffic destined to routeable, but unused portions of address space is called a Network Telescope. Examples of such traffic include undirected scanning and worm traffic. These tend to be driven by linear address traversal, or random/pseudo-random target address schema's [5] [15]. As described in [17], observation of traffic is based on looking at address or packet distribution – when an IP address is chosen at random, the probability of observation is expressed as a geometric distribution, while the odds of multiple packet observation are binomial in nature. Traffic modeling for active scanning is not independent per packet observed, so while binomial expressions for packet observations work well for *unused* address ranges, there may be issues using the same models for *used* address space.

Traffic characterization is broken up into several classes, Real (non-radiation/worm) traffic modeling was classically described by Paxon in [20]. Of significance is that type of traffic described was completely legitimate (in a TCP sense) and directed. Recently the idea that connection arrivals for interactive, user driven activity are the only successful candidates for Poisson description was looked at by Karagiannias et al. [9] in terms of appropriate time scales, aggregate traffic flows and packet inter arrival time distributions.

Radiation traffic is described as traffic that is the byproduct of systems that are infected with malicious code, or those that are somehow mis configured such that network traffic is sent in a non-directed manner to the local IP space. Barford et al [2] describes distributions of source addresses for radiation as being tightly clustered. These measurements include not just unused address space, but also firewall and intrusion detection data from the DSHIELD log aggregate in used address space. Hostile address space are shown to be both stable and localized. The result that these hosts are not smoothly distributed across address space is not only consistent with our observed data, but may lend further help in the analysis process here. Pang et al. in [17] look at issues in active responders and filtering while providing significantly more detail on content and specific exploit examples.

## 3 Dataset and Method

Data analysis looking for portability and repeatability will normally use pcap as a means to store and transport historical network traffic. The dataset and method used in this analysis differs from most in that raw connection data was gathered using Bro, then post-processed using Fast Bit indexing for the Root data analysis toolkit. This allows for the ability to interactively query the data set in 'what if' scenarios, even when the record count of the data set becomes enormous. Each component of this will be described in detail.

### 3.1 Initial Dataset

The data set consists of one week of complete network connection records originally generated by the Bro intrusion detection system located at a mid size high performance computing facility. The facility has a high speed (10 gigabit) Internet connection, and no firewalls or port blocking relative to the position of the data collection point. The address distribution of the class-B space is quite sparse, with approximately 4000 live hosts populating the space. The week selected for analysis was chosen at random from a set of 42 weeks which constituted a data set being used for other performance measurements described in [3].

The Bro instance used for data collection had a default configuration except that the base scan threshold was set at 50 rather than 100. In addition, the instance was checkpointed, or restarted every 24 hours. Scanners were detected via the native Bro algorithm, a modified native with an artificially low threshold (30) and TRW w/ default values. In addition, the TRW algorithm was modified to show when it identified a host as being a scanner, as well as when it determined that a connection was non-hostile.

The raw connection storage data structure provides access to the following attributes for each of the connections: connection time, duration; source IP, port, data transfered; destination IP, port and data transfered as well as a description of the overall direction and state characteristics.

In order to use the connection records with the indexing schema, each attribute had to be translated into binary form – each field is converted into its natural unit (for example, ports are saved as integers, Unix timestamps as doubles etc) and saved in their own individual file per attribute. These files are then indexed and used as data for the Root interface.

For further discussion, a TCP connection is defined to be 'good' (suspected non-hostile) if the three way handshake completes and neither side of the connection sends a RST. This rather liberal notion of a non-hostile connection is used since we are limited by several factors including missing connection startup and ending for long lived connections, and the possibility that packets may get dropped somewhere in the analysis pipeline. We recognize that a connection may be successful on the TCP layer, only to provide hostile data to an application, but given the lack of higher layer protocol data, there is nothing that we can do at this time to address application layer attacks using this tool.

## 3.2 Method

This section provides a brief description of the data indexing mechanism as well as the procedure required to use these tools for data analysis.

### 3.2.1 Root and Fastbit Data Indexing

The main tool used for interacting with the converted raw data is based on ROOT [22], an object-oriented data analysis system originally developed for processing large volumes of high-energy physics data. The ROOT system has a comprehensive set of analysis capability and rudimentary visualization capabilities. ROOT is straightforward to extend through loadable modules. We take advantage of this extensibility to extend ROOT so that it can answer multidimensional range queries using FastBit [11].

FastBit is a research code that implements a number of different forms of bitmap index compression including WAH - Word Aligned Hybrid coding [10]. In a basic bitmap index, one bitmap is allocated for each distinct value of the indexed attribute, where each bitmap has as many bits as the number of records in the indexed dataset. The size of the index grows linearly with the attribute cardinality and is small only for low cardinality attributes. A number of strategies have been proposed to reduce the size of a bitmap index, but WAH compression was shown to keep the index sizes compact, as well as to significantly reduce the query processing time compared to other indexing schemes [10].

Interaction with the data sets via this interface requires the use of an API that can be accessed via C++ code, or a shell like interface built into the ROOT package. For this work, short C++ programs were used to interact with Root, and shell scripts were used to drive the programs as well as doing simple data analysis.

### 3.2.2 Traffic Extraction and Analysis

In traditional data analysis, there are issues not only with processing the data, but also in storing and keeping track of it as well [19]. To combat the processing problem, a two step method is used. Initially the indexing is used for the extraction of a small, high value data set from the large general connection pool. This is normally possible at human interactive speeds, even for extremely large data sets [3]. A second program can then process the smaller extracted set in a slower (possibly linear) way without creating as significant a CPU or memory burden.

An example of this is the generation of scan detection data – the week of test data is segmented up into 24 hour blocks, which are broken out into /16 networks. The 24 hour blocks mimic the restarting of the IDS that normally takes place every 24 hours (when network connection data is globally reset), while the /16 network chunks allow for a small enough address block to run analysis against. As addresses are broken out into their individual octet components during initial conversion, it is natural to develop and process queries along a 8-bit boundary to maximize efficiency. In the second round, this subset of connection data can be run through known scan algorithms in parallel. This allows for a high degree of repeatability on tests and for the tuning of user applications / algorithms.

# 4 Data Analysis

The data Analysis section is broken out into several pieces. First the general distribution of known hostile addresses and their respective subnets are examined and compared to the TRW list. The connection characteristics from the augmented list of addresses is then compared to randomly selected addresses to see if they can be differentiated. The question of NAT and DHCP effects is then examined. Finally, the optimal granularity for the definition of 'localized subnet' is examined.

## 4.1 Overview of Data Set

In analyzing connection data, we used four different algorithms to identify scanners. As previously described, the native BSA was used in two modes: with thresholds set to 50 connections for a base set of scanners, and 30 connections to test what an unusually low value would produce relative to the proposed method. The TRW algorithm was used as an 'oracle' value to check suspected scanners against, as well as a large pool of high quality suspect addresses to qualify results against. Finally the modified BSA with adjustable scan threshold (MBA) is used as well.

To answer the natural question of just using TRW as a trigger mechanism, it is worth noting that the TRW algorithm suffers from a problem of being *too good* for some production infrastructures - the volume of addresses that need to be processed can exceed the capacity of some CAM/state tables. In addition, it is less familiar to many analysts and designers so there is resistance to it's implementation. With that in mind, it will be used only in an

Oracle capacity.

The scan results are summarized in table 1:

| ***Algorithm*** | ***Number of Identified Scanners*** |
|---|---|
| Bro Scan Algorithm (BSA) | 11624 |
| Bro Scan Algorithm – low threshold (BSAL) | 14972 |
| Threshold Random Walk (TRW) | 56888 |
| Modified Bro Algorithm (MBA) | 12695 |

Table 1: The number of identified scanning hosts per tested algorithm.

One of the things that stands out the most is that the lowered threshold BSA identified more scanners than MBA. This is expected behavior. Since MBA starts with the default scan threshold, it will always create fewer identified scanners than BSAL if the lowered threshold for BSAL is aggressive enough.

The BSAL entry in this case can be thought of as a straw man in that if the value used is low enough to catch scanners, but high enough to avoid false positives than it can be used in the base value for MBA as well. In this test case, the unusually low value (30 failed connections) would likely introduce false positives.

For the MBA algorithm, the scan threshold is lowered for subnet by a value that is directly proportional to the number of identified scanners during the time period used for learning. Looking at the distribution of these values in table 2, the vast majority of entries can be explained by background radiation since there is only one or two identified scanners per subnet during any point in the week of data. With a modified threshold of 48 or 49, the behavior of MBA very closely resembles that of BSA. Therefore the majority of the high value identifications are done with a fairly small fraction of the total number.

| ***Count*** | ***Number*** | ***% of Total*** |
|---|---|---|
| 1 – 2 | 7836 | 92.9 |
| 3 – 5 | 486 | 5.8 |
| 6-10 | 94 | 1.1 |
| 11-55 | 15 | 0.2 |

Table 2: Distribution of MBA metric counts

For the largest values (those greater than 20, or 5 instances), several instances of false positives are introduced. For the single value greater than 50, this produces automatic false positives for the entire subnet associated with it. In the production version of the MBA algorithm, a limit is placed on the maximum change possible to the threshold to avoid this scenario.

## 4.2 Distribution of Scanning Hosts and Networks

Given that a larger number of scanning hosts are seen with MBA than with the traditional BSA, it would be useful to get some indication as to the value of these addresses. If the additional hosts are better at describing the TRW list (which is treated as an Oracle) than BSA augmented by random selections from known connections, then the additional hosts are not the random byproduct of arbitrarily lowering the threshold. BSA is augmented by known connections in order to minimize the effects of data set abnormalities.

Judging similarity between the various address lists is a similar problem to selecting which of two collections of text are most similar to a control set. A tool kit has been developed which simplifies these calculations called the Ngram Statistics Package (NSP) [1].

In comparing the various lists (or vectors), the use of Dice Coefficients can be used to determine similarity between elements in each vector. This coefficient yields a number between 0 and 1. A coefficient of zero implies two documents have no terms in common, while a coefficient of 1 implies that the sets of terms occurring in each document are identical. It is typically derived by:

$$D(x,y) = 2|\vec{X} \cap \vec{Y}|/|\vec{X}|+|\vec{Y}|$$

Where the vectors are broken up into a sequence of tokens that occurs within a window of at least n tokens within the vector. If the first vector is shorter than the second, the longer is truncated.

S1: 10.0.0.1 10.0.0.2 10.0.0.3
S2: 10.0.0.1 10.0.0.2 10.0.0.5
C: 10.0.0.1 10.0.0.2 10.0.0.3 10.0.0.4

Here the average value of D(S1,C) = 1.000 and D(S2,C) = 0.8335 indicating (not unexpectedly) that S1 is closer to C than S2.

For this assessment, the network blocks are broken out into class A (/8) networks in order to provide a global comparison. There is no reason not to run this on smaller address lists, except for brevity. The comparison was run five times in order to reduce the chance that an unusual selection of random connections would throw the data. When the values for MBA and BSA are compared to TRW as the control group, there are several things that are noticed. This is illustrated in figure 1.

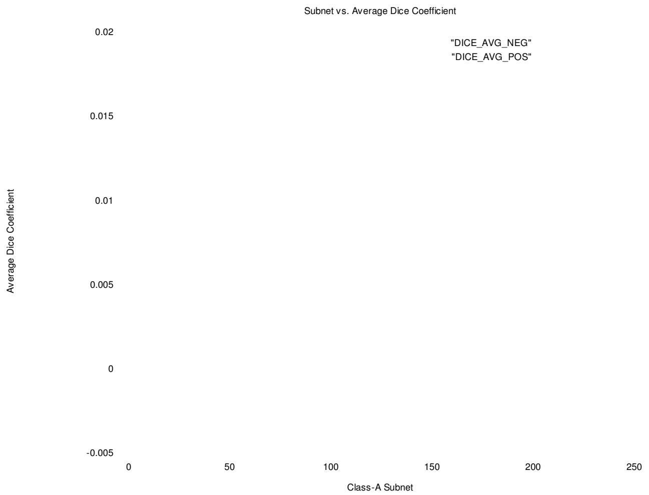

Figure 1: Average differences between the Dice coefficient for BSA subtracted from the value for MBA. Values above zero indicate that MBA is a closer match to TRW than BSA.

First is that a significant number of comparisons that favor the

MBA match over the BSA plus random addresses. This indicates that the additional addresses in MBA represent a better match to the TRW list than random;y selected addresses,

Another interesting point is that the negative results tended to be quite stable over *multiple* runs of random numbers. This is something that suggests further investigation as some variation was expected with the selection of different addresses.

### 4.3 Data Set vs. Random Distribution of Hosts

If there is an association between a scanning IP address and it's local network, then you would expect to see a difference between the number of connections seen from the scanner subnet and with a subnet derived from a randomly selected IP address. Here the number of connections needs to be corrected for the original scanning IP address' activity since we are interested in the overall behavior of the local subnet.

Using the list of scanners described by the BSA, a series of /24 networks is created. A subnet can appear multiple times if a scanning source is listed on multiple days, or multiple scanning sources are seen within the subnet. Hostile connections (as described in 3.1) from these /24 networks are extracted minus any attributed to known scanning addresses. The same process is done with an equal number of IP addresses randomly selected from the set of connections that contacted the destination address space during the sample week. The set is drawn from known addresses to avoid the possibility that the destination address range has characteristics that would invalidate a 'normal' random selection. The set is also checked against known BSA scanners to avoid duplication. When completed, we have a complete set of hostile connections from random and BSA identified networks, as well as values for the number of times we have repeatedly seen each network.

If a network is seen multiple times in the data set (such as when a scanner spends several days attacking the local address space), it will be known as a *revisit*.

If there is a correlation between hostile hosts and the networks that they are identified with, there should be a greater number of connections seen from the 'hostile networks' than from those randomly selected. In addition, as the number of revisits increases, there should be a growing divergence between the hostile and the random lines. This is expected since randomly selected subnets should have on the whole traffic that is not directed, and should therefore fulfill the binomial distribution described for Network Telescopes in [16].

As seen in Figure 2, the initial raw data provides a rough view of the relationship between random and scanner networks since the numbers tend to be dominated by a small number of outlier values.

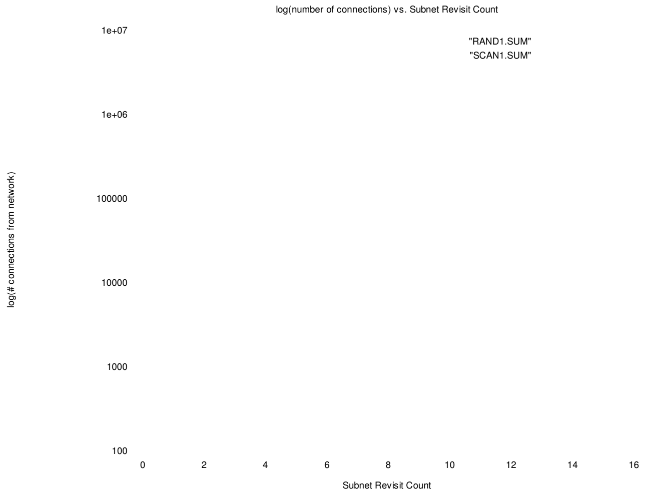

Figure2: Number of hostile connections per subnet vs. the revisit number. For the random values, there was no revisit value for 5.

In the random set of networks, we see significantly fewer failed connections than exhibited by the scanner networks. As well, the number of revisits seen by random networks is significantly smaller as well. This points to the conclusion that the makeup of traffic surrounding the given address sets can be viewed as more hostile than arbitrary traffic in terms of network scanning behavior.

Since the total number of connections for revisits is almost the same for random as well as scanner networks, it is worth explaining. By plotting the distribution of the number of connection per subnet vs. how many times that value was seen over the week, the similarities and differences for each of the revisit counts can be seen.

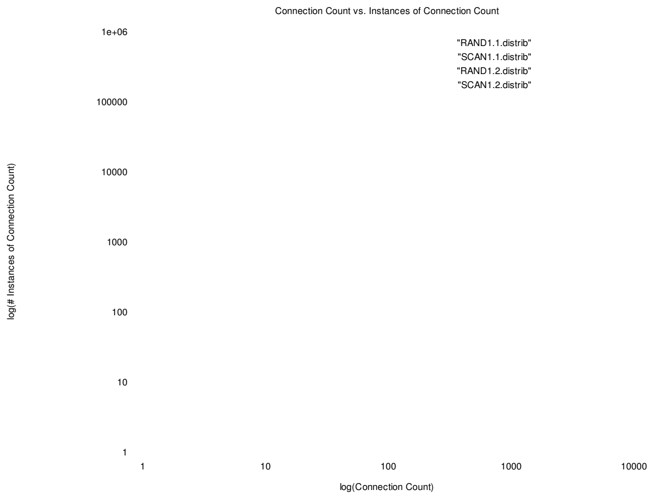

Figure 3: Number of connections per subnet vs. the count for each of those values.

There is a (not unexpected) divergence between the values for one and two revisits. Plotting these values in figure 3, we see that there is reasonably tight matching when the revisit count is one. As the revisit count increases from one, the two values diverge as expected. This should indicate that the characteristics of the two data sets (scanning vs. random) are similar in the instance where the scanning data most resembles the randomly selected addresses

### 4.4 Addressing NAT and DHCP

As already mentioned, there are questions about problems with NAT shielding one or more subnets behind a set of address in a subnet as an explanation for the neighborhood behavior described above. Additionally, the extensive use of DHCP raises issues of associating malicious behavior with a subnet whose population may be transient. These simple questions do not, unfortunately, have a simple answers.

Detecting the existence of a translated address space from empirical data has been looked at in several studies, and is considered a known hard problem. As discussed in Casado et all in [4], it was possible to infer the existence of RFC 1918 address space (192.168.0.0/16) by examining the distribution of source addresses in Code Red II HTTP GET requests, and adjusting for the worms preferential scanning of local networks. This is based on Code Red II selecting a random destination address from the local /8 half of the time, so disproportional numbers of these addresses will be seen by sensors in the legitimate 192.0.0.0/8 networks. Deraison and Gula in [14] discuss monitoring the IP ID field to detect and identify areas of NAT traffic by tracking reuse and sequences. Moore [26] most recently discusses looking at browser identification strings in order to associate multiple web browsers to a single IP address – an indicator of possible NAT activity. Since neither application layer protocol information, nor detailed IP header data is available via the current data source, these tools and methods are not useful here.

A variation on the general method used by Moore in [26] is to look for inconsistencies in the data set of identified scanners. As previously mentioned, the TRW algorithm was used to identify both hostile and non-hostile connections. An argument for identifying NAT networks could be developed if there exists an address in both hostile and non-hostile TRW lists. Since there was no crossover between the two lists, it can be argued that during the time period represented by the data set there were no gross examples of actively hostile and non-hostile actions going on where there is a *single* external NAT translation address. This does not suggest that there were no NAT instances, just none that were in front of address spaces that would routinely visit the destination address space.

If the scope is increased to looking at overlap between /24 networks in the good and bad TRW lists, there is significantly more crossover observed. On the other hand, we move from a clear case of inconsistent with a single address, to a loose inference from a set of addresses. The results in table 3 are calculated by describing the most constraining subnet mask that can hold all addresses identified by TRW as hostile and non-hostile.

| *Subnet Mask* | *Number Hosts Allowed by Mask* | *Count Observed* |
|---|---|---|
| 32 | 1 | 0 |
| 31 | 2 | 0 |
| 30 | 4 | 1 |
| 29 | 8 | 1 |
| 28 | 16 | 2 |
| 27 | 32 | 3 |
| 26 | 64 | 11 |
| 25 | 128 | 10 |
| 24 | 256 | 18 |

Table 3:

Again, it should be noted that looking at more than one address provides only a loose inference (at best) that there might be some sort of NAT activity. It has been included here for completeness.

### 4.5 Optimal Subnet Size

In looking at the appropriate size for subnet granularity, a balance must be struck between making the neighborhood large enough to capture traffic from the actual local sources, and not so large that false positives are generated from artifacts in unrelated systems. Also, if this algorithm is used to modify the weights for scan detection, the effects on benign traffic must also be minimized. Up till this point, subnet size has been described in terms of a class-C or /24 network. The selection for this was initially based on the smallest typical network described in general works such as Barford et al. in [2].

To determine the optimal subnet size, traffic from the /16, /24 and /25 networks surrounding addresses identified in the original BSA run was extracted from the week long sample period. Connections were classified as 'hostile' and 'not hostile' using the rules defined in section 3.2. Table 4 shows the percent changes in hostile and non- hostile connection that result when the definition of a local subnet is changed from a /16 to a /24, and from a / 24 to a /25.

| *Connection Type* | *Change in Value for/16 -> /24* | *Change in Value /24 -> /25* |
|---|---:|---:|
| Not Hostile | -63.49% | -0.13% |
| Hostile | -58.17% | -6.68% |

Table 4: Changes in TCP Number of Connections for Changes in Subnet Size

Based on Table 4, the benefit from moving from a /16 subnet mask to a /24 subnet mask is considerable – particularly in terms of reducing the effect on successful connections. For moving from the /24 to a /25, there is a trivial change in the number of non-hostile connections seen (which is not good), and a non trivial number of reduced hostile connections. The gain (not touching possibly legitimate connections) is far outpaced by the loss in hostile traffic.

From this quick analysis, it seems as though a /24 subnet provides the most benefit at the lowest cost.

## 5 Operational Experience

Using a modified scanner detection policy, the 'Bad-Neighborhood' algorithm was put into operational use for a period of seven months on the same link that the test data was extracted from. The hostile address list was derived from the previous six month periods detected scanning hosts and during that time it was used for modifying scan thresholds, the list was never modified. To avoid impacting legitimate traffic, the limit to how far the scan threshold could be adjusted down was set to 15. For this, the subnet mask associated with each of the scanning addresses was set to a /25, as the data presented in 4.3 was not yet available.

Additional factors in this data set include the existence of other scan detection algorithms that will, in some cases, flag scanners before the traditional BSA.

Based on the proportion and frequency of how hosts are effected by this, the use and re-use of addresses for attackers can be illustrated. Looking at Fig 4 we see:

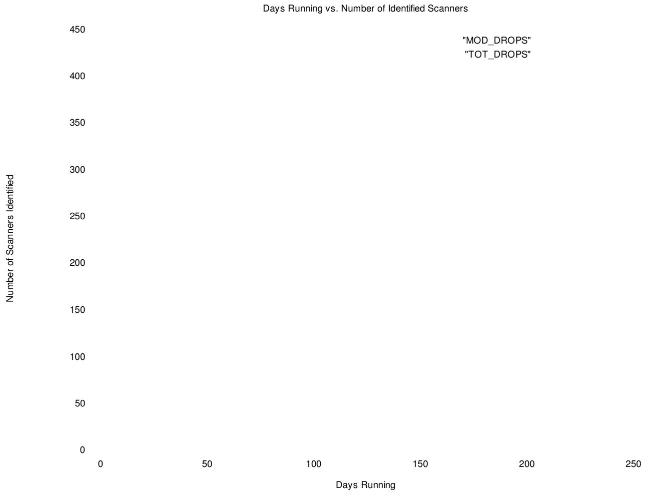

Figure 4: Number of scanners identified per day, and the number of scanners that were effected by the MBA algorithm.

From which can be seen several interesting points. First is that the proportion of traffic that is effected by MBA decreases with time. The effect is not as pronounced as might be expected if there were significant churn in the location of hostile address space. This is consistent with the findings of Barford in [2].

Second, there are several large attacks that take place from address space that seems unrelated to the previously learned ranges. This is particularly evident around day 75. This indicates the need to be able to dynamically update the list of hostile addresses/networks.

Finally there seemed to be no identified ill effect from the use of this algorithm as far as the mis identification of legitimate connections as hostile. Since the drop threshold was the only thing modified, this is not particularly surprising.

## 6 Conclusion and Future Work

In this paper we looked at the characteristics of traffic originating from the local subnets of identified scanners. We concluded that lists of hostile addresses created with the MBA algorithm more closely resemble the assumed good list of attackers provided by TRW. In addition, the volume of traffic exhibiting hostile characteristics is significantly different both in terms of volume and the number of subnet revisits when subnets are compared between random and known scanner sources. Finally the optimal size of the subnet local to the scanner address was calculated.

There are a number of directions that this could be taken. Initially it would be quite beneficial to show that this behavior is not specific to the data set used in this analysis. If the thesis holds, then looking at trying to apply it to newer trends in the 'scanning landscape' [21] would be beneficial.

Another example of this might be lowering the number of failed attempts for ssh logins from address space that it identified as hostile. In addition providing a feedback mechanism to adjust metrics associated with given networks should be simple. Also, expanding the idea of what is being measured by adjusting network threshold based on failed logins or http attacks as well as scanning. This should help draw in active attack spaces as well as the scanners that are using them

## Acknowledgment

The author would also like to thank the National Energy Research Scientific Computing Center for access to computing facilities, as well as Vern Paxon for providing significant feedback in the development part of this project.

## References

[1] S. Banerjee, T Pedersen. The Design, Implementation, and Use of the Ngram Statistic Package. Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Feb. 2003.

[2] Paul Barford, Rob Nowak, Rebecca Willett, Vinod Yegneswaran, Toward a Model For Source Addresses of Internet Radiation. PAM Adelaide, Australia 2006.

[3] E. Wes Bethel, Scott Campbell, et al. “Interactive Analysis of Large Network Data Collections Using Query Driven Visualization”, LBNL Report Number 59166 .

[4] M. Casado, T. Garfinkel, W. Cui, V. Paxson and S. Savage, Opportunistic Measurement: Extracting Insight from Spurious Traffic, Proc. HOTNETS 2005.

[5] Renaud Deraison, Ron Gula, “Using Nessus to Detect Wireless Access Points”, Tenable Network Security, May 5, 2003, http://www.tenablesecurity.com/images/pdfs/wap-id-nessus.pdf .

[6] Mike Gleason, The Ephemeral Port Range, http://www.ncftpd.com/ncftpd/doc/misc/ephemeral_ports.html .

[7] Tony Hain, A Pragmatic Report on IPv4 Address Space Consumption, The Internet Protocol Journal - Volume 8, Number 3 http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-3/ipv4.html .

[8] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan, Fast Portscan Detection Using Sequential Hypothesis Testing, Proc. IEEE Symposium on Security and Privacy, May 2004.

[9] T. Karagiannis, M. Molle, M. Faloutsos, A. Broido, A Nonstationary Poisson View of Internet Traffic, INFOCOM, Hong Kong, 2004.

[10] Kesheng Wu, Ekow Otoo, and Arie Shoshani. On the rerfor-

mance of bitmap indicies for high cardinality attributes. In Proceedings of the International Conference of Very Large Data Bases, pages 24-35, 2004 .

[11] Lawrence Berkeley National Laboratory Scientific Data Management Group. Fastbit, http://sdm.lbl.gov/fastbit, 2005

[12] Lippman, R.P., Haines, J.W., Fried, D.J., Korba, J., Das, K. , "The 1999 DARPA Off-Line Intrusion Detection Evaluation", Proceedings of the Third International Workshop on the Recent Advances in Intrusion Detection: Lecture Notes in Computer Science series, Springer Verlag, Toulouse, France, Oct. 2000.

[13] McHugh, J. ,"Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory", ACM Transactions on Information and System Security, 3 (4), Association for Computing Machinery, New York, NY, Nov. 2000, pp. 262-294.

[14] avid Moore, Colleen Shannon, and Jeffery Brown, "Code-Red: a case study on the spread and victims of an Internet Worm," in ACM Internet Measurment Workshop 2002, Marseille, France, Nov 2002, http://www.caida.org/outreach/papers/2002/codered/.

[15] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver, Inside the Slammer Worm, Security and Privacy, July/August 2003.

[16] David Moore, Colleen Shannon, Geoffrey M. Voelker, Stefan Savage, Network Telescopes: Technical Report http://www.-caida.org/publications/papers/2004/tr-2004-04/

[17] R. Pang, V. Yegneswaran, P. Barford, V. Paxson and L. Peterson, Characteristics of Internet Background Radiation, Proc. ACM IMC, October 2004.

[18] V. Paxson, Bro: A System for Detecting Network Intruders in Real-Time, Computer Networks, 31(23-24), pp. 2435-2463, 14 Dec. 1999.

[19] V. Paxson, Strategies for Sound Internet Measurement, Proc. ACM IMC, October 2004.

[20] V. Paxson and S. Floyd, Wide-Area Traffic: The Failure of Poisson Modeling. IEEE/ACM Transactions on Networking, Vol. 3 No. 3, pp. 226-244, June 1995.

[21] F. Pouget, M. Dacier, V.H. Pham. Collection and Analysis of Data from Honeypots. LEURRE.COM Honeypot Project.

[22] ROOT – An Object Oriented Data Analysis Framework, http://root.cern.ch, Nov 2005 .

[23] Stuart Staniford, James A. Hoagland, and Joseph M. , McAlerney. Practical automated detection of stealthy portscans. In Proceedings of the IDS Workshop of the 7th Computer and Communications Security Conference, Athens, 2000.

[24] Paul A. Watson, Slipping in the Window: TCP Reset attacks, Can Sec West, 2004, Vancouver, Canada.

[25] Nicholas Weaver, Stuart Staniford, Vern Paxson. Very Fast Containment of Scanning Worms, Proc. USENIX Security Symposium, August 2004.