Locality: A new Paradigm for Thinking About Normal Behavior and Outsider Threat

John M^{c}Hugh
CERT Research Center and
Center for Computer and Communications Security
Carnegie Mellon University
Pittsburgh, PA, USA
jmchugh@cert.org

Carrie Gates
CERT Analysis Center
Carnegie Mellon University
Pittsburgh, PA, USA and
Department of Computer Science
Dalhousie University
Halifax, NS, Canada
gates@cs.dal.edu

ABSTRACT
Locality as a unifying concept for understanding the normal behavior of benign users of computer systems is suggested as a unifying paradigm that will support the detection of malicious anomalous behaviors. The paper notes that locality appears in many dimensions and applies to such diverse mechanisms as the working set of IP addresses contacted during a web browsing session, the set of email addresses with which one customarily corresponds, the way in which pages are fetched from a web site. In every case intrusive behaviors that violate locality are known to exist and in some cases, the violation is necessary for the intrusive behavior to achieve its goal. If this observation holds up under further investigation, we will have a powerful way of thinking about security and intrusive activity.

Categories and Subject Descriptors
C.2 [**Computer-Communications Networks**]: Local and Wide-Area Networks; C.2.5 [**Local and Wide-Area Networks**]: Internet—*observations of traffic characteristics*

General Terms
Security

Keywords
Locality, Network Observation, System Behavior

1. INTRODUCTION

Big whorls have little whorls
  That feed on their velocity,
And little whorls have lesser whorls
  And so on to viscosity.
    – Lewis F. Richardson as quoted by George Gamow in "Creation of the Universe"[7]

Multiscale locality has proven to be a key to understanding physical a wide variety of physical phenomena. The piece of dogeral above is quoted in a 1950's popular cosmology book[7]. It serves to illustrate the observation that locality, manifested as clustering, appears at many scales in the observable universe. In the cosmological world, clusters form at the scale of planets with their satellites and smaller systems up to clusters of clusters of galaxies and beyond.

Closer to home, locality of program counter and data references turned out to be the key to the design of effective memory paging systems[4]. In this case, the key locality concept is the "working set," i.e a set of memory pages that, if maintained in the physical memory of the computer will allow the program (or programs) in execution to make progress without excessive page faulting. This work was in response to the observation that, on some time sharing computers, page faults occurred so frequently that the CPU was mostly idle, waiting for the page(s) containing the next data or instructions to be referenced to be loaded into memory. This phenomenon, termed thrashing, led to a variety of models of program behavior, the understanding of which allowed efficient implementation of paged memory systems. As a side benefit, this area also led to studies that resulted in efficient data structures and algorithms for dealing with data whose size demanded organization in virtual memory.

The thesis of this paper is that locality principles are a key to distinguishing and understanding "normal" behavior in computer systems that may be subject to attack by outsiders. We feel that an understanding of normal is an important step towards understanding that portion of abnormal behavior that represents the actions of malicious users of the system. Our long term goal is to develop a sufficient understanding of the systems with which we work so that we can identify properties that are necessary parts of certain malicious activities, and, with luck, properties that are sufficient to indicate such activities. As an example, one of the few we have, a necessary aspect of the behavior of rapidly spreading worms such as Code Red or SQL/Slammer is that they attempt to make contact with a large number of potentially infectable hosts in a short period of time.

The individual observations on which the thesis is based are

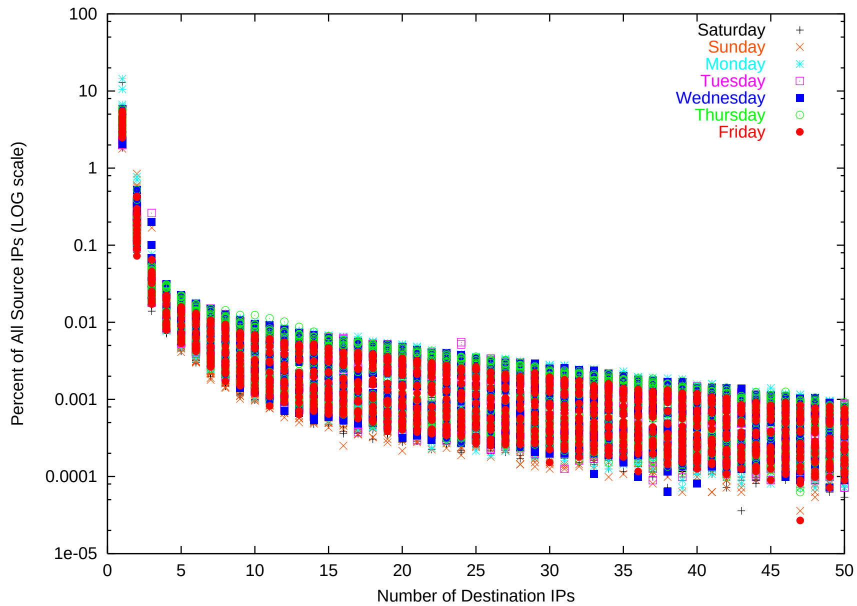

Figure 1: The number of unique source IPs that contacted up to 50 unique destination IPs per hour for the week of 11–17 January 2003. This graph represents all outgoing TCP data.

not unique, but their unification into a guiding principle is. We note that locality is a fairly broad concept. In general, locality is manifest when the behavior of the system can be represented by relatively compact clusters in some dimensions of a multidimensional measurement space. The clustering may appear at various scales, i.e. the time and amount of data necessary to manifest a cluster may vary widely. Time, typically the rate at which events are observed or the intervals between them may be one of the dimensions along which clustering occurs. If normal behavior exhibits clustering and abnormal behavior fails to cluster (or clusters in a different way), we have a mechanism that has the potential for discriminating between normal and abnormal behaviors. Note that we have not used the terms "benign" and "malicious" as surrogates for normal and abnormal. In this context, abnormal means unusual. In some cases, as we attempt to understand why locality appears to characterize normal behavior, we may be able to make a case that certain classes malicious behavior are necessarily abnormal in that it will necessarily fail to meet the "normal" clustering criteria. On the other hand, we may not be able to make the case that all normal activity necessarily satisfies the "normal" clustering criteria so that failure to cluster is evidence of malicious behavior, but does not identify such behavior with absolute certainty.

In the remainder of the paper, we consider a number of observations in which the locality principle is manifest and make the case that, at least observationally, they are strong indicators of normal behavior at an appropriate scale. We then look at a variety of malicious behaviors that appear to violate these notions of locality, examining the question of whether or not the the violation is necessary or fortuitous. We conclude by presenting (limited) evidence from our own data to support the thesis and outline the future research that we hope will lead to a better understanding of the paradigm.

It remains to explain the term "outsider" in the title. As Jim Anderson noted in his 1980 paper on detecting computer misuse[2], in the limit, the malicious behavior of an insider is indistinguishable from that of a normal, non-malicious, user. The phenomena that we are discussing are the result of activities by outsiders who have not taken care to tailor there attacks to mimic the behaviors of the normal user populations of the systems being examined. In some cases, we are examining purely outside or external behaviors such as the characteristics of packets that arrive at the border of an enterprise network. In other cases, the behavior is internal, e.g. a worm propagating from an infected host, but we still prefer to characterize it as outsider since the code that does the propagation originates outside the system and has not been created with any notion of observing and mimicking normal users, i.e. it represents the actions of a visiting outsider who makes no effort to fit in.

2. MOTIVATING EXAMPLES

In this section, we discuss a number of examples in which locality appears to be a key to describing normal behavior. The time scales involved range from weeks or days down to seconds or less.

2.1 Gross Scale Workstation Connectivity

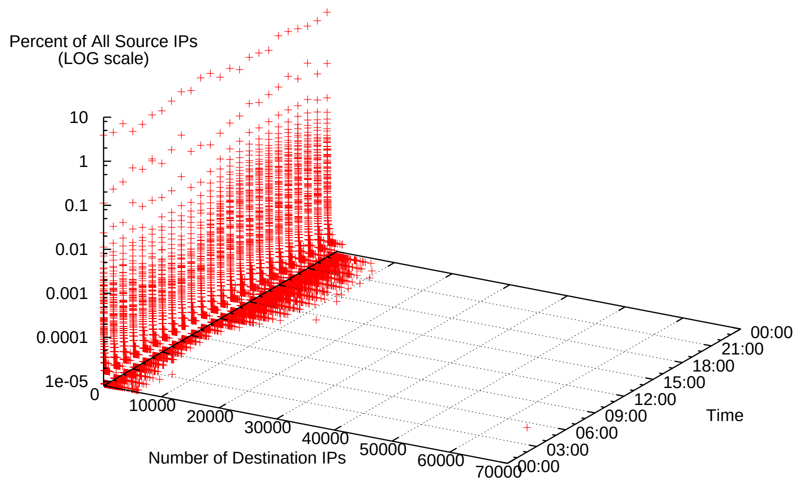

Figure 2: The number of unique source IPs that contacted each number of unique destination IPs per hour for January 14, 2003. This graph represents all outgoing TCP data.

Hofmeyr observed in his dissertation[8], that, at least in the context of his network at the University of New Mexico and with a few exceptions, the set of network addresses with which an individual workstation makes contact stabilizes within a period of several weeks after observation starts. After that point, the addition of new addresses into the set is relatively uncommon and may be taken as an indication of misbehavior on the part of the system initiating the connection.

The general conclusion is that most users operate largely within a closed community of systems with which they make contact. The exceptions are fairly obvious:

- Workstations belonging to a system administrator whose job included making contact with a wide variety of other systems were excluded.
- HTTP browsing behavior was excluded for all users. The nature of the web where material at one site contains links to a variety of other sites is not likely to reach closure in most setting. Fortunately, browsing behavior appears to manifest useful locality on a smaller scale as we will see in section 2.2 below.

2.2 Fine Scale Workstation Connectivity

Williamson [12] presents a hypothesis that browsing behavior exhibits small scale locality. Based on a limited set of observations, he concludes that, for populations similar to his coworkers, a working set containing the 10 most recently visited IP addresses is a good predictor of the next IP address to be visited. Least Recently Used (LRU)replacement is used to maintain the working set with timestamps that are updated whenever an address already in the working set already is accessed. Departures from this locality are relatively frequent, but not persistent. Many of the violations result from extraneous factors such as references to sites containing pop up advertising material.

Figure 1 provides a view of a network, providing support for Williamson's use of 10 for the size of a working set of IP addresses. This graph shows the number of source IP addresses that contacted up to 50 IP addresses per hour, for data spanning one week in January, looking at outgoing data produced by Cisco NetFlow for a large network. (This data consists of flows, not packets, and has no directionality associated with it. That is, we do not know definitively whether the internal or external machine started the flow.) The majority of source IP addresses (nearly 10%) contacted only 1 IP address per hour. And, looking at any one hour, at least 94.9

Williamson uses a "soft limit" to react to locality violations. When an address that are not in the working set are accessed, the access request is placed in a paced delay queue which limits the rate at which such requests can be dispatched to one request per unit delay (1 second in Williamson's case). When a delayed access is dispatched, all queued pending requests destined for the same address are sent immediately in the order in which they were enqueued and the destination is inserted in the working set, replacing its least recently accessed address. In this way, small locality violations are tolerated with minimal delay, but gross violations encounter ever increasing delays. The

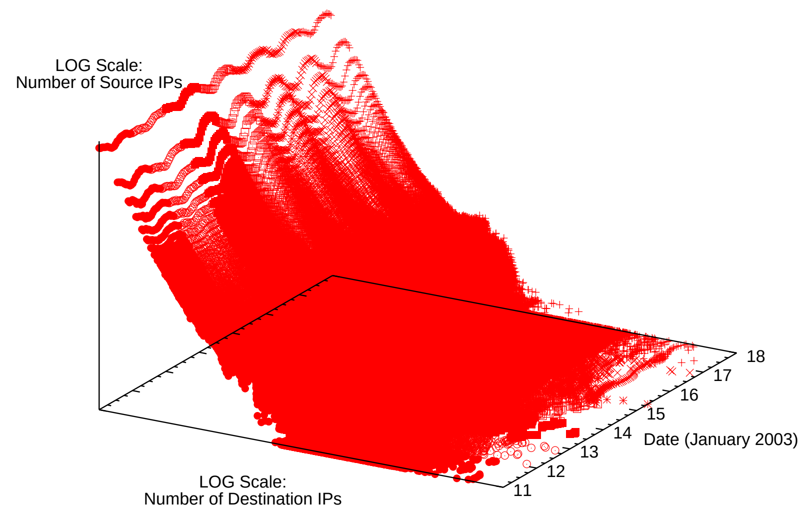

Figure 3: The number of unique source IPs that contacted each number of unique destination IPs per hour for January 14, 2003. This graph represents all outgoing TCP data.

system is intended to counter rapidly spreading malicious codes such as "Code Red"[9]. In this case, queue lengths in excess of some predetermined size might invoke more drastic responsive actions such as disconnecting the affected system.

The notion that unusual and potentially malicious behavior may violate locality properties is demonstrated in Figure 2. In this graph, the number of source IP addresses that contacted some number of destination IP addresses for each hour over a typical workday during the month of January 2003 is shown. There is one source IP address that contacted 65,536 destination IP addresses during one hour. In this case, the source performed a port scan of a class B-sized network. By definition, a port scan must violated notions of locality in terms of the number of destination IP addresses typically contacted by any given source IP, whether server or workstation, and so can be easily recognized in a well-mannered network.

We suggest alternative representations for the working set that obviate the queuing mechanism making for more mathematical representations of locality.

Fixed size working set In this representation, the maximum working set size is fixed. When a new address is seen, the least recently used[1] address is replaced with the newly referenced one. The measure of locality is the frequency with which the contents of the working set changes. In the queuing model, we could look at the effective working set as the union of the actual working set and the set of unique addresses currently in the queue. Based on Williamson's observations where queue lengths are seldom more that two, a working set with a size of 12 (vice Williamson's 10) might be expected to see an update frequency of less than 2 per second. Updates much in excess of this would be considered as gross violations of locality.

Variable size working set with constant removal In this representation, the size of the working set is not fixed. Whenever an address not in the set is seen, it is added. At fixed intervals, say equal to the delay queue time, the least recently used entry is removed from the working set. At any given time, the current size of the working set represents the locality of the monitored system. Based on the behavior of the fixed set plus queue model, we would expect the size of the working set to be relatively small and stable. If the size exceeds a threshold , this would be considered as a gross violation of locality.

Note that, under fixed assumptions on the distributions of addresses with time, the three models could probably be shown to be equivalent. Until we have empirical data on this and some assurance that the actual distributions are tractable, we choose not to undertake this approach.

The notion that unusual and potentially malicious behavior

[1]Other replacement policies than Least Recently Used could be adopted. Investigation of alternate policies is a possible path for future research. Our intuition is that changes in policy may result in slight changes in optimal working set size, but are unlikely to affect the validity of the hypothesis.

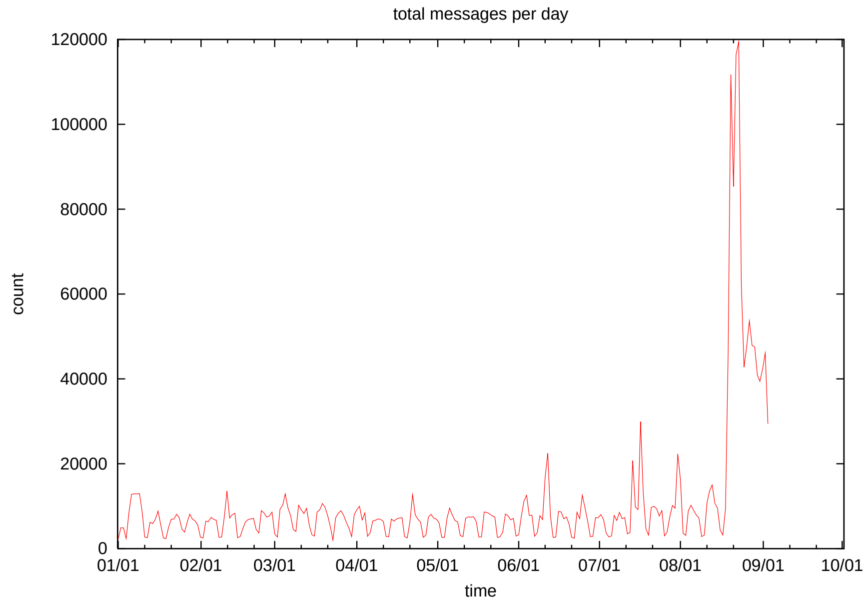

Figure 4: Variations in email message count with a virus

may violate locality properties is demonstrated in Figure 2. In this graph, the number of source IP addresses that contacted some number of destination IP addresses for each hour over a typical workday during the month of January 2003 is shown. There is one source IP address that contacted 65,536 destination IP addresses during one hour. In this case, the source performed a port scan of a class B-sized network. By definition, a port scan must violated notions of locality in terms of the number of destination IP addresses typically contacted by any given source IP, whether server or workstation, and so can be easily recognized in a well-mannered network.

2.3 Incoming Traffic

Both of the examples given in the previous section, along with previous work by Williamson[12], focus on outgoing traffic. The notion of locality should also apply equally to incoming traffic as seen at the border of an enterprise network, most likely with a larger working set. To test this notion, a graph of the number of source IP addresses that contacted some number of destination IP addresses for a week in January for all incoming TCP traffic was generated, and can be seen in Figure 3. The majority of incoming sources contact a single destination IP address on the target network, and may represent activities such as contacting a particular web server, or checking e-mail from a home address. On the opposite extreme, a small number of sources contacted a very large number of internal addresses. It is suspected at this point that these sources may represent events such as port scans. Thus far, attempts to calculate a working set for this data have not proven fruitful and there are no obvious locality violations of the sort seen in Figure 2. Investigation is on-going to better understand the behaviors shown in this graph. The dataset used apparently contains a very large amount of scan and probe data. One of the authors (Gates) is currently working to identify these scans and we plan to repeat the attempts to construct a working set with this data removed. In addition, we do not know precisely which addresses in the monitored network have computers assigned to them at any given time. Removal of attempts to contact non existent machines from the incoming traffic (a first approximation to the removal of scans) may alter the picture substantially.

Work at Boeing [6, 5] indicates that locality (as represented by changes in address entropy) can be observed in network border or core data. In this case, a stream of border data that was known to be free of DDoS attacks was examined and the entropy of the set of the 10,000 most recently seen source addresses was calculated. The stream was augmented with a simulated DDos attack that used spoofed source addresses and the attack was easily recognized by the change in entropy of the addresses. It is noted, that an attack using non-spoofed addresses would also be detected by this mechanism, as well.

2.4 Gross Scale Email Addressing

Beginning with Melissa in 1999, we have seen a number of wide spread email based viruses. While the detailed behavior of of these viruses is discussed in section 2.5 below, email viruses exhibit a kind of locality violation, albeit complex, at an enterprise level. For the purposes of this discussion, we assume that the email origination and reception behavior of the enterprise as a whole is easily observable. This will

Figure 5: First difference of email message counts with a virus

be relatively easy if all email, internal and external passes through a limited number of (perhaps one) mail servers.

In this case, the locality is manifest in the time dimension. Under normal circumstances, email transmission patterns follow a fairly regular pattern that reflects the working hours of the enterprise in question. A typical email pattern under normal circumstances can be seen in the left hand side figure 4. Typical, successful, email viruses spread by sending copies of themselves to a fairly large number of addressees, usually obtained from some address repository available on the account of the user being attacked.

As the enterprise becomes infected, the rate of emission of email increases, sometimes exceeding historical peaks, as shown at the right hand end of figure 4. If we plot the derivative (daily difference) of the email volume, we see that the slope becomes steeper as the attack starts as shown in figure 5. These two figures are based on the CERT email volume for the first 8+ months of 2003. The peak in late August represents the outbreak of a "sobig" email virus. The noise in the peak represents some anomalous behavior caused by the excessive mail load.

In addition we may be able to define secondary indications of locality in email behavior. If we attempt to cluster mail based on properties such as sender, nature and size of attachments, order of addresses, etc., we think that only virus messages and those sent to mailing lists are likely to cluster significantly. Work by Stolfo and his group at Columbia, addressed at the detection of malicious codes in email attachments, holds promise in this area[11, 10]. Their approach, based on data mining, builds classifiers for benign and malicious email content based on the learning of discriminators from labeled data. The result is the establishment of content based locality measures that cluster normal and malicious content.

2.5 Fine Scale Email Addressing

Many users maintain address books that are used in the sending of email. Starting with the Melissa email virus in 1999, a number of email viruses have taken advantage of these address books to guide their propagation. By definition, sending email to an address or addresses found in an address book cannot be considered to be abnormal, per se. On the other hand, we suspect that most email activity follows relatively simple patterns that demonstrate considerable locality over a time frame ranging from hours to days. In observing this locality, it is necessary to distinguish between receiving an email and reading it. When an email is read, the reader may perform one or more responsive actions, including:

- Delete the email.
- Reply to the email.
- Forward the email to one or more individuals.
- Originate an email to another party without including the provocative original.
- etc.

We believe that observing email behaviors for a cross section of users will allow us to build locality models for email

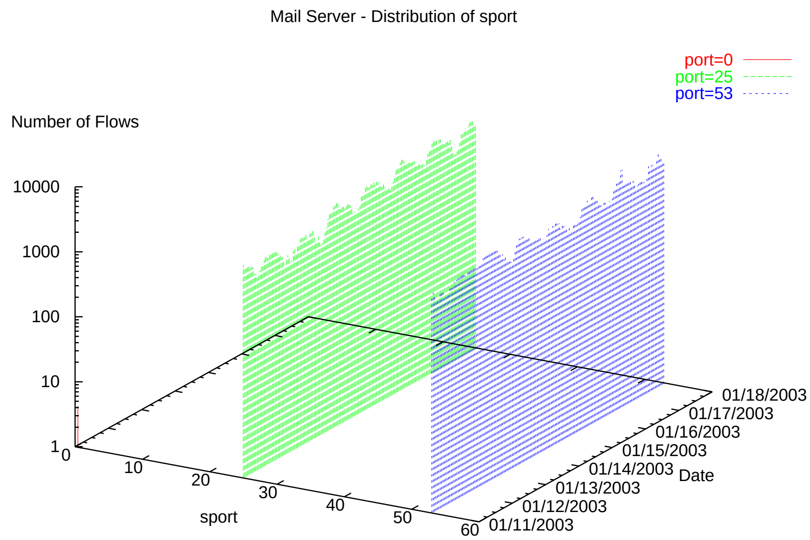

Figure 6: Temporal and port locality for an email server

connectivity that can be used to distinguish normal activity from the automatic spread of malicious code via email. It will be interesting to see what role address books play in characterizing this activity.

Problematic areas include users who maintain large scale mailing lists as part of their job and regularly launch large batches of email with identical content to the lists. We note that only a specifically targeted malicious email worm would be likely to provoke activity from the keeper of such lists without provoking a general flurry of abnormal email activity from other users.

2.6 Server Considerations

Pure servers can be expected to exhibit locality based on their intended function. We note, in passing, that a machine configured as a pure server[2] would have an outgoing locality set with no entries in it and any attempt by such a server to initiate an arbitrary connection is suspect[3]. Hybrid servers do exist. Systems that do price comparisons take requests from users and make outgoing connections to determine the best price for the requested item. We believe that the outgoing connection sets for such machines are relatively small and stable, though perhaps not as small as Williamson suggests for workstations.

[2]by pure server, we mean a machine that only responds to external requests, but never initiates requests to other systems.

[3]Syslog messages being sent to a log server would be the exception, but the point is that the outgoing connection set is both small and stable.

For server machines, it is the incoming connection set that we need to examine for locality behaviors. Fortunately, a substantial amount of work has been done characterize both the temporal and spatial locality of web traffic. Motivated by the need to design appropriate caching mechanisms, Almedia, et. al. observe[1] that a stack distance model based on a LRU stack of object references is a good model for characterizing the temporal locality of web page references. Note that this is quite similar to the workstation locality working set discussed above. In the case of web page temporal locality, a stack of page requests is maintained with the most recently requested page on top. When a page that is already on the stack is requested, it is moved to the top. For each request, r_i, we can compute a distance d_i that is the number of positions up the stack that the requested document is moved. Thus, a request string $r_1, r_2, \ldots r_n \ldots$ Can be converted into a distance string $d_1, d_2, \ldots d_n \ldots$ that preserves the pattern of activity, but does not depend on document names. The statistical distribution of the distance values is a representation of the temporal locality of the request strings. In practice, popular documents stay near the top of the stack with relatively small distances.

In addition to temporal locality, web request traces exhibit spatial locality, defined as the tendency of substrings of requests to be repeated in the overall request trace. This occurs whenever there is a canonical navigation through a series of pages in a particular order. It turns out that spatial locality on web references is a fractal property, i.e. it exhibits detail at all scales[1] showing both short and long range dependencies among request strings. Note that this is characteristic of bursty processes in which regions of intense activity tend to occur at irregular intervals.

Figure 6 shows the temporal behavior of a machine that is acting almost exclusively as an email server. In this figure, we see that the outgoing activity consists of approximately equal amounts of DNS (port 53) and SMPT (port 25) traffic[4]. Since email addresses are symbolic, they must be converted to IP form before the delivery connection can be made. The figure is based on observations made at the border of a large network and reflects only traffic from inside to outside, i.e. deliveries within the network are not shown.

We suspect that patterns involving temporal or spatial locality occur for other services and that further investigation can lead to ways in which their normal activities can be characterized.

2.7 Content Locality and Clustering

We recently became aware of work by Cilibrasi, et. al. on clustering of music[3] based on approximations of its Kolmogorov complexity. This might provide an approach for identifying, for example, members of a polymorphic virus family. We have not yet had time to investigate this further, but hope to do so in the near future.

3. LOCALITY AS A UNIFYING PARADIGM

The examples given above are neither complete nor exhaustive. In many cases, they are based on very limited observational data and a program of observation and experimentation is needed to see whether they hold up on a large scale. Nonetheless, locality appears to be a powerful framework for unifying many aspects of normal behavior. Why this should be is not entirely clear. It is likely that some efficiency of action phenomena are involved. In the biology of motion, small motions involving resonance phenomena lead to efficiency. Trees in a forest tend to have similar resonant frequencies allowing them to bend together in response to wind gusts. People tend to behave in repetitive ways that exhibit various forms of locality. Perhaps it is only reasonable that human artifacts such as systems of computer programs should exhibit similar properties.

Acknowledgments

Numerous people at CERT and elsewhere have contributed to this effort. Matt Williamson of HP Bristol Labs. first started us thinking about locality as discussed here. He continues to work in the area and is currently applying himself to the analysis of email locality for virus detection. Damon Bicknell of the US Army and CMU is responsible for Figure 6. Rudy Maceyko of CERT provided the email data shown in figures 4 and 5. Mike Collins of the CERT Analysis Center made the data on which figures 1, 2, 3, and 6 are based available to us. Sven Dietrich of the CERT Research Center and Mike Reiter and Chenxi Wong of CMU have also contributed to the formulation of our ideas.

4. REFERENCES

[1] Virgílio Almeida, Azer Bestavros, Mark Crovella, and Adriana de Oliveira. Characterizing reference locality in the WWW. In *Proceedings of the IEEE Conference on Parallel and Distributed Information Systems (PDIS)*, Miami Beach, FL, 1996.

[2] James P. Anderson. Computer security threat monitoring and surveillance. Technical report, James P. Anderson Co., Fort Washington, PA, 1980. Available online at `http://seclab.cs.ucdavis.edu/projects/history/CD/ande80.\pdf`.

[3] Rudi Cilibrasi, Paul Vitanyi, and Ronald de Wolf. Algorithmic clustering of music. Found on the web at `http://www.cwi.nl/~paulv/selection.html` apparently points to `http://arxiv.org/archive/cs/0303025`.

[4] E. G. Coffman and P. J. Denning. *Operating Systems Theory*. Prentice-Hall Inc., 1973.

[5] Laura Feinstein, Dan Schnackenberg, Ravindra Balupari, and Darrell Kindred. DDoS tolerant networks. In *Proceedings of the 2003 DARPA Information Survivability Conference and Exposition*, volume II. IEEE Computer Society, April 2003.

[6] Laura Feinstein, Dan Schnackenberg, Ravindra Balupari, and Darrell Kindred. Statistical approaches to DDoS attack detection and response. In *Proceedings of the 2003 DARPA Information Survivability Conference and Exposition*, volume I. IEEE Computer Society, April 2003.

[7] George Gamow. *The Creation of the Universe*. The New American Library for World Literature, Inc., 1952.

[8] Steven Andrew Hofmeyr. *An Immunological Model of Distributed Detection and Its Application to Computer Security*. PhD thesis, University of New Mexico, 1999.

[9] David Moore. An analysis on the spread of the code-red (crv2) worm. Available online at `http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml`.

[10] Matthew G. Schultz, Eleazar Eskin, Erez Zadok, Manasi Bhattacharyya, and Salvatore J. Stolfo. Mef: Malicious email filter a unix mail filter that detects malicious windows executables. In *Freenix '01*, 2001.

[11] Matthew G. Schultz, Eleazar Eskin, Erez Zadok, and Salvatore J. Stolfo. Data mining methods for detection of new malicious executables. In *IEEE Symposium on Security and Privacy*, 2001.

[12] Matthew M Williamson. Throttling viruses: Restricting propagation to defeat malicious mobile code. In *Eighteenth Annual Computer Security Applications Conference*, December 2002.

[4]The port 0 behavior is bogus. It represents a failure to filter out ICMP messages prior to the analysis.