Proceedings of the 2009 IEEE Symposium on Computational Intelligence
in Security and Defense Applications (CISDA 2009)

Analysis of the 1999 DARPA/Lincoln Laboratory IDS Evaluation Data with NetADHICT

Carson Brown, Alex Cowperthwaite, Abdulrahman Hijazi, and Anil Somayaji

Abstract— **The 1999 DARPA/Lincoln Laboratory IDS Evaluation Data has been widely used in the intrusion detection and networking community, even though it is known to have a number of artifacts. Here we show that many of these artifacts, including the lack of damaged or unusual background packets and uniform host distribution, can be easily extracted using NetADHICT, a tool we developed for understanding networks. In addition, using NetADHICT we were able to identify extreme temporal variation in the data, a characteristic that was not identified in past analyses. These results illustrate the utility of NetADHICT in characterizing network traces for experimental purposes.**

I. Introduction

In 1999 DARPA commissioned Lincoln Laboratory to create a synthetic benchmark for evaluating intrusion detection systems [6]. These IDEVAL data sets have become notorious in the intrusion detection community: they have been both widely used and widely criticized. This criticism is due to the fact that an accurate performance with this data set can have little bearing on how an IDS system will perform in "real" environments; however, they remain in use because, to this day, they are some of the only publicly-available data sets for IDS evaluation.

One key criticism of the IDEVAL data sets is that their simulated normal network traffic is unrealistic. While their creators went to considerable effort to simulate a realistic network environment [7], there are some clear deviations from the network traffic patterns one would expect in a real network. For example, critics have noted that the underlying network topology is unusually flat and the traffic is unusually uniform (low in "crud"), leading to artificially low false positive rates in evaluated systems [10], [9]. Such deviations, however, have largely been revealed only after extensive manual analysis guided by expert knowledge.

For the past few years we have been developing NetADHICT [5], a tool for understanding the structure of network traffic. NetADHICT allows network operators to visualize ongoing traffic patterns while using minimal knowledge of standard network protocols. We now have significant experience in analyzing production networks with NetADHICT [2], [3]. In this paper we study whether NetADHICT is able to detect the unusual patterns in the 1999 DARPA IDS Evaluation (IDEVAL) data sets.

The authors are part of the Carleton Computer Security Lab, School of Computer Science, Carleton University, Ottawa, Ontario, Canada (email: {carson, acowpert, ahijazi, soma}@ccsl.carleton.ca}).

This work was supported by Canada's NSERC though the ISSNet Strategic Network and the Discovery Grants Program (AS); additionally, CB and AH were each funded by the Ontario Graduate Scholarships in Science and Technology (OGSST) program.

We have found that these data sets look very different from ones captured from production networks. Many of the characteristics observed by other researchers through painstaking analysis are easy to observe with NetADHICT; in addition, other, even larger-scale artifacts become more obvious with the use of NetADHICT's visualizations. Instead of making use of flow- or bandwidth-based network tools, researchers might select NetADHICT for a high-level view of current usage of a given network—including unpopular or non-standard protocols, such as peer-to-peer services. These results indicate that NetADHICT can be a valuable tool for evaluating network captures for experimental purposes.

The remainder of the paper is structured as follows: Section II describes the IDEVAL and the history of its analysis. Section III provides an introduction to NetADHICT, including a brief description of its operation of the internal algorithm. Description of our testing methodology and organization is contained in Section IV. Our analysis is presented in Section V presents our analysis of the IDEVAL data sets; Section VI discusses what we found. We conclude with final remarks on this research in Section VII.

II. Past Analyses of DARPA/Lincoln Labs Data

In 1998, and again in 1999, the Lincoln Laboratory at MIT, under contract from DARPA, developed a series of data sets in order to test the correctness and robustness of existing Intrusion Detection Systems (IDS) [7]. These data sets were created by using host computers connected together with a traffic generator to model a small US Air Force base of limited personnel, connected to the Internet. Network traffic and host audit information was recorded during the experiments. Three weeks of training data and two weeks of test data were released, as well as a list of all attacks included in these synthetic data sets. Our work examines the network captures from the the 1999 experiments.

Haines *et al.* [1] describe the DARPA/MIT Lincoln Lab evaluation (IDEVAL) data set as a fictitious Air Force base with hundreds of users across thousands of machines. Programmers, secretaries, managers and other users were simulated by user automata. These automata "send and receive mail, browse websites, send and receive files using FTP, use telnet to log into remote computers and perform work, send and receive IRC messages, monitor the router remotely using SNMP, and perform other tasks" [1, p. 18] in order to simulate background traffic for the embedded attacks in the data sets. These *virtual* machines included a heterogeneous mix of Linux, SunOS, Solaris and Windows NT machines, connected by a Cisco router. Attacks are

978-1-4244-3764-1/09/$25.00 ©2009 IEEE

initiated from both inside and outside of the local network. Network traffic is captured on either side of the router using `tcpdump`.

The IDEVAL data sets have been primarily used to evaluate intrusion detection or other network security systems. These data sets are useful because they are entirely synthetic, containing no proprietary nor confidential information for any real users. The IDEVAL data sets have been used in a number of well-cited papers [8], [11], [12], [13]; however, they have also been used in contexts that were not ideal.

Of particular note is the 1999 The Third International Knowledge Discovery and Data Mining Tools Competition (KDD Cup 1999). They chose the 1998 DARPA/Lincoln Labs data sets as their target data set [4]. This competition brought the work of Lincoln Labs to the attention of the machine learning community, leading to a huge number of papers applying various algorithms to them. Unfortunately, as noted by McHugh [10], these data sets have a number of artifacts that make them unsuitable for evaluating learning-based approaches to intrusion detection. Specifically, the normal traffic is too uniform: the machines behave in a too similar manner, and there is a distinct lack of malformed background traffic, or "crud." Mahoney *et al.* [9] also reported several other inconsistencies with real traffic captures, notably regularities regarding TCP SYN packets and severe predictability in source addresses and packet header fields such as the time to live (TTL). Because of these features, attacks in the IDEVAL data sets are much easier to detect than in regular network traffic.

While it is possible to mitigate some of this uniformity by mixing in traffic captured from production networks [9], the underlying problem was that the artifacts of this data set were only apparent to experts in the field, and then only after a significant amount of manual work. What is needed, then, are tools that could reveal such patterns, but in a way that is much clearer and that requires much less domain-specific knowledge. As we explain below, we believe NetADHICT is one such tool.

III. NetADHICT

NetADHICT is a tool for understanding the structure of network traffic. It allows users to visualize traffic as a series of clustering decision trees, typically with one tree for every ten minutes of observed traffic. These trees classify packets depending upon whether they contain a series of (p, n)-grams: fixed-length strings at fixed offsets within a packet. An example tree is shown in Figure 1. A packet is first tested against the (p, n)-gram tree in the root node. If it is present in the packet, clustering proceeds to the left; if not, the right branch is followed. The packet is clustered when it reaches a leaf node.

All leaf nodes (known as *clusters*) are examined by a port-based packet classifier. Each wedge of a cluster represents one class of traffic contained within the node. Wedges are proportional to one another inside of a cluster; cluster size represents a logarithmic measure of packet counts. $N6$, the left-most cluster in our example, only contains DNS traffic.

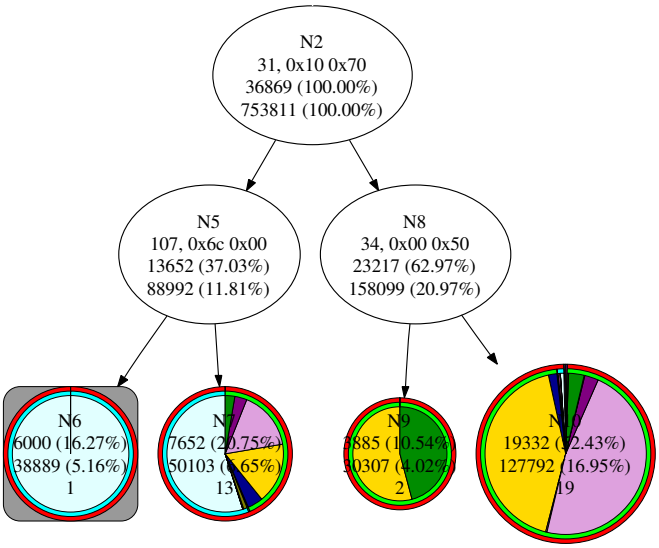


Fig. 1. Example tree created by NetADHICT. The internal nodes are labelled with a node identifier, a 2-byte (p, n)-gram (offset plus two bytes in hexadecimal), and all nodes contain frequency counts for a short and long-term windows (10 minutes and 3 hours). These counts are used to determine when nodes are to be split or deleted. Last, all leaves store a count of all protocols they contain.

We refer to such clusters as being *singular*. The right-most cluster, $N10$, contains 19 types of traffic. Note that these packets did not match any of the (p, n)-grams in the tree. We call this cluster the global *default cluster*.

NetADHICT incrementally learns a set of (p, n)-grams describing network traffic based on the relative frequency of packets. Specifically, if too many packets are clustered in a specific node, it is split by adding a new (p, n)-gram decision node. The (p, n)-gram is chosen such that it matches approximately half of the packets being clustered in the leaf node. Similarly, when too little traffic has been clustered to a given leaf over a sufficient period of time, it and its sibling are deleted; its parent then becomes a new leaf node. The full details of this algorithm, known as approximate hierarchical divisive clustering (ADHIC), is described by Hijazi *et al.* [2].

There are two things to note about NetADHICT. One is that even though its trees are created without reference to standard classifications of network traffic, their structure generally corresponds to standard classifications: IP versus non-IP traffic, TCP versus UDP, HTTP versus SMTP. Another is that NetADHICT makes use of the fact that real network traffic has a (p, n)-gram frequency distribution that is remarkably consistent over time. We discuss these patterns further in Section V.

IV. Test Method

To test NetADHICT's performance on IDEVAL, we used three weeks of the training data captured from the sniffer on the inside of the network. Each week is comprised of data for Monday through Friday from approximately 8am to 6am the following day. Some created traces were cut short due to system crashes during the data capture [6].

We performed two rounds of testing with NetADHICT. The first involved running the NetADHICT backend on each of the $\leq$ 22 hour traces. Each trace was run from a single file (rather than the multiple files used in [2]). NetADHICT was run with its standard parameters: 10 minute ticks and an 18 tick maturation period (180 minutes). A tick is the interval at which the tree is evaluated for merges and splits. A node may not split or be pruned from the tree until its maturation period had expired. This resulted in trees with a maximum depth of 5 or 6, and 18 to 26 terminal clusters.

For the second test, we merged the data sets together to form three week-long traces. The timestamps in the traces were shifted to remove the two hour empty period between each trace (from approximately 6am to 8am) and then merged into a single large trace file. NetADHICT was again run with standard parameters. This second run provided a longer term view of the evolution of the tree, providing more time for the tree's structure to stabilize. The trees for the week long traces were much larger: they had a maximum depth of 10 or 11 and contained 60 to 75 terminal clusters.

Many of the results are compared to a previous capture of normal traffic taken at our lab. This previous capture has been published and thoroughly described in [2]. The behavior of our lab network appears to be representative, based upon our observations of a university network and that of a small company [3].

Protocol	IDEVAL week 2	CCSL
IPv4	7199540	6132185
TCP	6524425	3801738
TCP Unknown	3317	43678
MS WBT/MS RDP	0	9811
IPP	0	486936
IMAPS	0	166291
HTTPS	0	123979
SSH	344044	238124
MS Streaming/RTSP	0	97813
MSNMS	0	3767
XMPP	0	1221
TCP Sophos	0	5879
TCP No Payload	3071845	1888325
RTSP	0	2319
NBSS	2474	105
IRC	3119	0
TELNET	1930763	0
FTP	107781	5276
SMTP	250670	25923
CVS	0	11644
POP	2367	4881
HTTP	808037	684335
UDP	651324	2047182
UDP Unknown	4468	660
DNS	527710	66911
CUPS	0	128278
WHO	0	6650
RTP	0	248642
NBDGM	1427	62493
DCE_RPC	790	15873
NBNS	12512	176379
RIPv1	39534	41538
HSRP	0	1293451
DHCP	0	1161
NTP	60804	5081
ICMP	23791	24475
EIGRP	0	258756
ARP	30715	869547
ETHER (old)	6419	74107
Total no. of Packets	**7275137**	**7075868**
Total Size in MB	**1,539**	**1,819**

TABLE I

PROTOCOL CLASSIFICATION AND CONTENT STATISTICS FOR THE SECOND WEEK OF THE IDEVAL DATA SET AND OUR LAB (CCSL) NETWORK TRACE. ONLY PROTOCOLS WITH PERCENTAGE $\geq 0.01\%$ ARE SHOWN (BEST VIEWED IN COLOR).

V. ANALYSIS

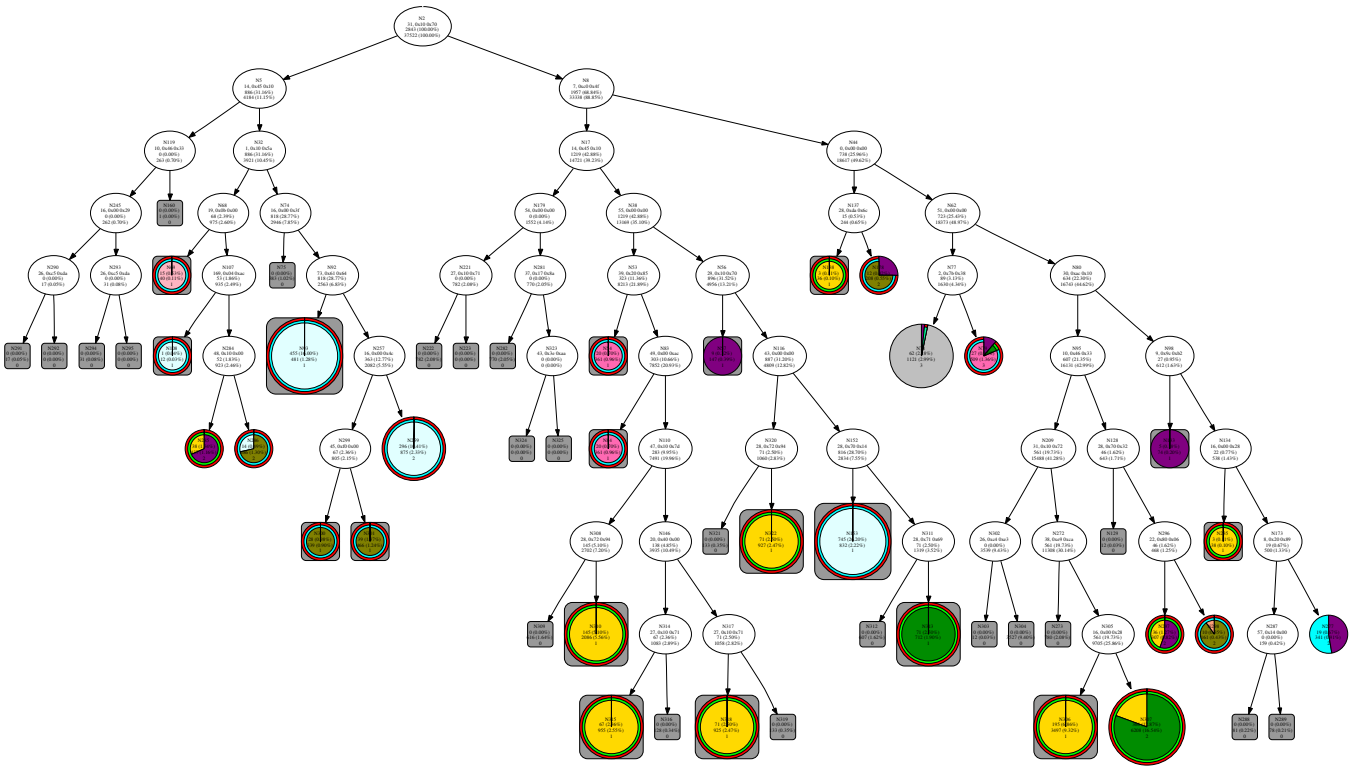
We divide our analysis of IDEVAL into four parts. First, we examine the temporal distribution of traffic as shown by the evolution of NetADHICT's trees (Section V-A). We then examine artifacts in the frequency distribution of (p, n)-grams in Section V-B. We examine the surprising lack of unclassified traffic in IDEVAL in SectionV-C. We then explore the significance of how traffic is classified, or how easy is it for NetADHICT to characterize the IDEVAL traffic, in Section V-D.

It is important to note here that our analysis of the IDEVAL data set is primarily concerned with the quality of the network captures, not potential attacks or other anomalies. The objective is to show, through the use of NetADHICT, that this data set lacks particular qualities that are found in real network captures, and contains artifacts of an artificial nature.

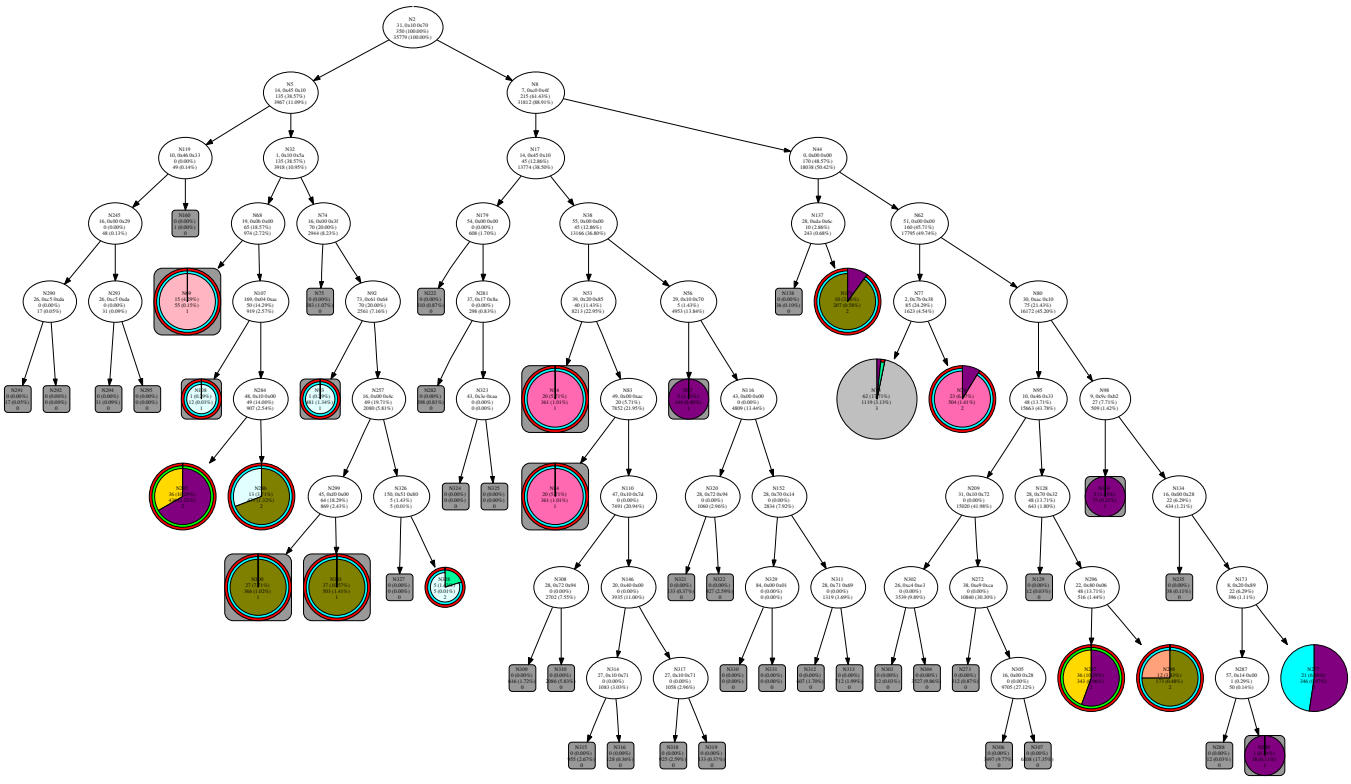
A. Temporal Distribution of Traffic

While network traffic is very heterogeneous, there are many consistent structural patterns in network traffic: the same hosts communicate using the same protocols, sending back and forth the same kinds of information, again and again. NetADHICT works because it can extract much of this structural similarity. The IDEVAL datasets, however, appears to be missing some of this consistency. We make this observation because we see a "strobe"-like effect in regards to the trees NetADHICT creates. Consider Figures 2(a) and 2(b), which show a two consecutive "snapshots" of the NetADHICT tree from the second week of the IDEVAL data set, ten minutes (one tick) apart. NetADHICT does not commonly show completely different looking trees this close in time. Many clusters are created from a particular burst of traffic, then left empty when the burst ceases. New bursts cause new nodes to be created, but they later quickly disappear. Such large changes in tree structure over a short period of time is something we never see in regular network traffic. Some clusters may grow or shrink; overall, however, the structure of the trees remain consistent.

We further characterize the bursty nature of the IDEVAL data sets in Figure 3. Here we have compared them to captures of our lab's network. The Carleton Computer Security Lab (CCSL) network is comprised of a half-dozen servers, and more than a dozen desktop machines, composed of Linux and Macintosh operating systems. We note that our traffic capture also contains attacks, just as the second week of the IDEVAL data set contains labelled attacks. Though our network is significantly smaller than the IDEVAL virtual machines, its network topology is similar, and it generates traffic at a similar scale.



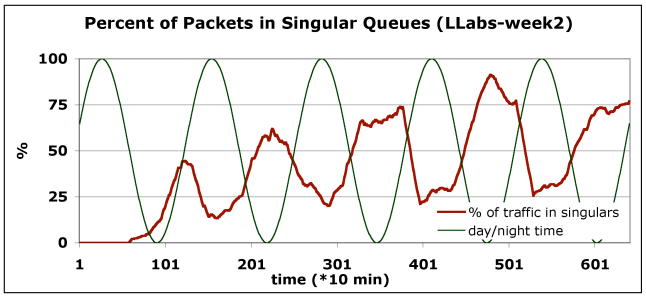
(a) IDEVAL, week 2 data set at 240^{th} tick

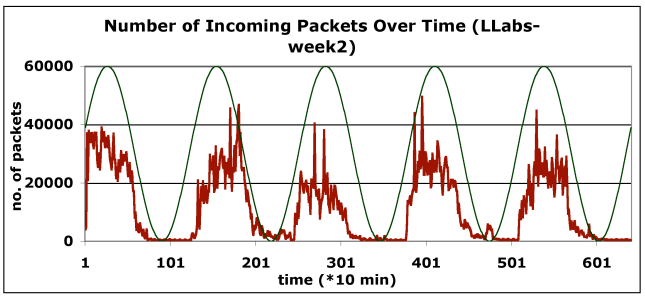


(b) IDEVAL, week 2 data set at 241^{st} tick

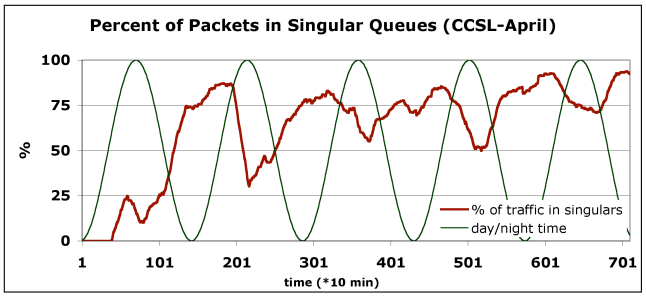
Fig. 2. The IDEVAL data lacks consistency, which causes erratic, strobing trees, with clusters appearing and disappearing. Note the large amount of empty clusters (small gray squares) in Figure 2(b).

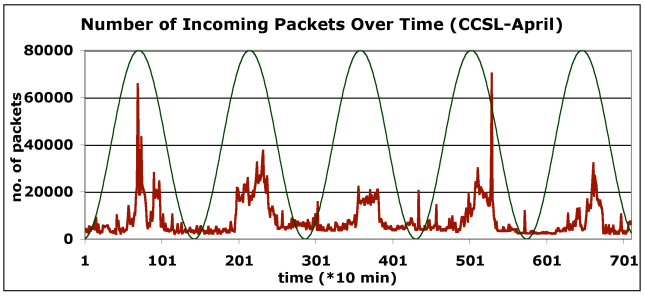
Surges of traffic are more pronounced in the IDEVAL data set, fitting much more closely to the passage of daytime (see Figure 3(a)). The nighttime hours contain far less traffic than our lab captures, providing at times only a few hundred packets over ten-minute intervals—something remarkably quiet for a network with thousands of machines. In contrast, our lab (Figure 3(b)), with many fewer (but real) machines has a steady baseline of thousands of packets in the same sized intervals.





(a) The IDEVAL Data set





(b) CCSL data set

Fig. 3. Temporal analysis of packet distribution over one week periods in the IDEVAL data set and our lab (CCSL). Note the IDEVAL graphs contain data which has been modified to close the two-hour gap between traces; the overlaid sine wave has been adjusted to account for this. The top of the crest of the wave in both figures denotes noon, and the bottom of the valley denotes midnight.

B. Frequency Distribution of (p, n)-grams

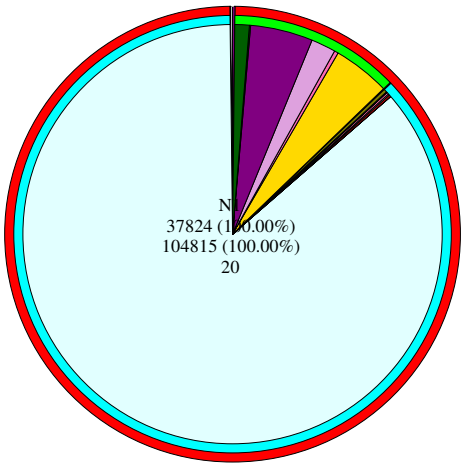


Fig. 4. Example of high volumes of DNS traffic. DNS is illustrated as the large, light blue wedge.

The breakdown of traffic in the IDEVAL data set compared to our lab's capture in Table I does not show any significant irregularities. However, if we look at the traffic at shorter time periods (10 minutes), we can see that some protocols are over-populating the traffic. This is exemplified by Figure 4 with the large amounts of DNS traffic over a 10 minute time period. The graph shows a single cluster—before any splits have occurred—dominated by approximately 85% DNS traffic. This is the first 40 minutes of the second week of the data set. Little explanation is available for such a large amount of DNS traffic effectively flooding the network.

Moreover, Figure 5 looks at offsets of the 1000 most frequent (p, n)-grams in three periods of our lab (CCSL) and the IDEVAL data sets. While the percentages of (p, n)-grams throughout the three different periods of CCSL show consistency between day and night, the percentages of the IDEVAL data sets do not. Moreover, the consistency difference is also visible when examining the 10-minute period against the 3-hour period it is part of. The discrepancy with the IDEVAL data set can be clearly seen among the day (3-hour and 10-minute) and night (3-hour) time periods with payload (i.e. $p > 53$) and TCP header (i.e. $37 > p > 54$) (p, n)-grams. Note the contrast with the very consistent real traffic also in the figure.

C. Unclassified Traffic

NetADHICT normally leaves a portion of the analyzed traffic unclassified. These packets can be found in the furthest right leaf of the tree in the global default cluster. In our analysis of the IDEVAL data sets, we have noticed a lower amount of this unclassified traffic compared to the traces from our lab. While the lack of unclassified traffic does potentially point to a lack of "crud" in the dataset [10], it is also potentially due to the greater variety of network protocols currently in use today compared to those used in 1999. Note that the majority (85.3%) of unclassified traffic

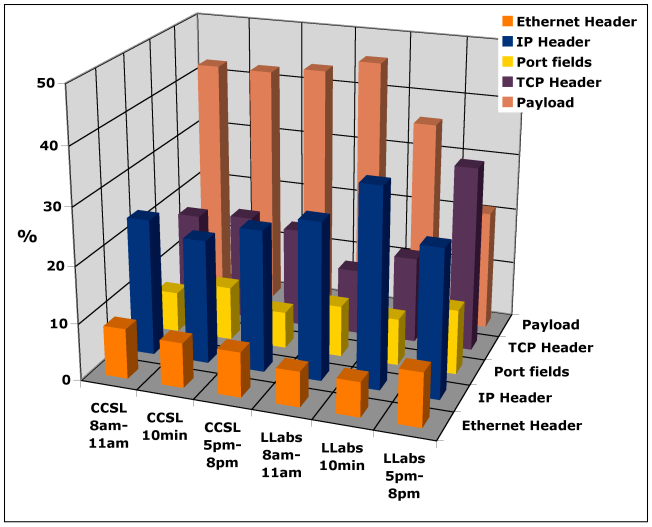


Fig. 5. Comparison of our lab (CCSL) and the IDEVAL (LL) traffic captures over three hour and ten minute periods, separated by packet type. The x-axis represents the two 3-hour data sets (morning and evening) of both data sets along with the last 10 minutes of the 3-hour morning time period of each. The y-axis describes at what packet offset p those (p, n)-grams are found (Ethernet header, IP header, port fields, TCP header and payload). The z-axis (height) of the graph denotes the percentage of (p, n)-grams contained within each packet offset range. Note the relative consistency of the CCSL traffic at different times of the day and at shorter time intervals.

from the IDEVAL data set is the `loopback` protocol used for network equipment management, such as Cisco routers.

D. Classified Traffic

Figures 3(a) and 3(b) also show the rate of isolation of traffic types. A singular cluster is a node of the ADHIC tree which contains traffic of one classification by our port-based classifier. Comparing our lab capture with the IDEVAL data set, it is clear that the IDEVAL data is not as quickly separated by protocol. This is likely due to the burstiness of the traffic: traffic comes and goes frequently enough that NetADHICT doesn't have time to cluster it properly.

When looking at the classification performance, we noticed further artifacts in the relative distribution of DNS traffic. During a 10 minute tick there is a high number of DNS requests relative to other traffic. During the tick from approximately March 17, 1999, 00:40-00:50, 60% of the traffic is DNS. Traditionally, we have seen spikes of DNS traffic due to other protocols requiring DNS information; however, the other 40% is primarily RIPv1 and NTP—protocols that normally do not generate large amounts of DNS traffic. This seems to imply that at least 20% of the traffic is arbitrary name look-ups, without any further communication. Consider another case as captured in Figure 4, where the majority of traffic thus far was DNS. Thus, not only is the DNS traffic bursty; it also does not coincide with the traffic that would normally generate DNS requests.

VI. Discussion

To summarize, NetADHICT quickly revealed a number of unusual traffic patterns in the IDEVAL data set, illustrating shortcomings in its simulation of normal network traffic. Some of these patterns, such as the unusually uniform distribution of packets [9], have been previously noted. Other observations, in particular, the extreme temporal variation, we believe is novel.

The contribution of this paper, however, does not lie in our observations of the IDEVAL data set, per se. Instead, what is notable is the ease with which we could identify the unusual properties of the data sets. The temporal variation manifests as remarkably dynamic tree that "strobes" in a way that virtually never happens with traces gathered from production networks. A modest amount of subsequent analysis then revealed the other characteristics, such as the lack of "crud," identified by past researchers.

NetADHICT was designed to provide a high-level view of network data, one that reveals large-scale patterns that may or may not follow the bounds of IP addresses and ports. While such functionality is potentially valuable when monitoring production networks, here we have shown that it is also a potentially valuable tool for the researcher, one that complements standard packet aggregate counts and manual packet and flow-level inspection. While there are many patterns that it does not readily capture (such as flow counts), we believe NetADHICT's ability to unify high-level and low-level network traffic views make it a powerful addition to the network researcher's toolbox.

The problem of creating network data sets for research purposes is a difficult one. Synthetic and anonymized data sets are essential resources; however, artifacts in them can lead to conclusions that do not hold on production networks. We believe lightweight clustering strategies such as that employed by NetADHICT hold the potential for proactively identifying data artifacts in network data—captured, synthetic, and anonymized—so they may be factored into experimental design. Such work should increase the quality of research results and reduce the need for later critiques.

VII. Conclusion

NetADHICT provides a novel way of visualizing network traffic, both as snapshots and changes over time. This paper describes how a given synthetic data set, the 1999 DARPA/Lincoln Laboratory IDS Evaluation Data, can be examined, and describes ways in which we have confirmed previous shortcomings of this data set, as well as provided a novel way to locate and analyze discrepancies within a given data set. By uncovering the varying distribution of (p, n)-grams in network traffic over time, NetADHICT allows administrators and researchers to observe structural patterns not readily observable using standard network visualizations. This work illustrates the utility of NetADHICT in identifying unusual patterns in network traces, something that is necessary for understanding the results of any experiments utilizing network captures. It is in this way that we believe NetADHICT is a useful addition to the network researcher's toolbox.

REFERENCES

[1] J. W. Haines, R. P. Lippmann, D. J. Fried, M. A. Zissman, and E. Tran. 1999 DARPA intrusion detection evaluation: Design and procedures. Technical Report TR-1062, MIT Lincoln Laboratory Technical Report, 2001.

[2] A. Hijazi, H. Inoue, A. Matrawy, P. van Oorschot, and A. Somayaji. Discovering packet structure through lightweight hierarchical clustering. *Communications, 2008. ICC '08. IEEE International Conference on*, pages 33–39, May 2008.

[3] A. Hijazi, H. Inoue, A. Matrawy, P. van Oorschot, and A. Somayaji. Lightweight hierarchical clustering of network packets using (p,n)-grams. Technical Report TR-09-03, School of Computer Science, Carleton University, February 2009.

[4] Information and Computer Science, University of California, Irvine. KDD Cup 1999 data, 1999. `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.

[5] H. Inoue, D. Jansens, A. Hijazi, and A. Somayaji. NetADHICT: A tool for understanding network traffic. In *Proceedings of the 21st Large Installation System Administration Conference (LISA'07)*, Nov 2007.

[6] Lincoln Laboratory, MIT. DARPA intrusion detection data sets, 2008. `http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html`.

[7] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. The 1999 darpa off-line intrusion detection evaluation. *Computer Networks*, 34(4):579 – 595, 2000. Recent Advances in Intrusion Detection Systems.

[8] M. Mahoney and P. Chan. Learning rules for anomaly detection of hostile network traffic. pages 601–604, Nov. 2003.

[9] M. V. Mahoney and P. K. Chan. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In *Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection*, pages 220–237. Springer-Verlag, 2003.

[10] J. McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Trans. Inf. Syst. Secur.*, 3(4):262–294, 2000.

[11] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, pages 5–8, 2001.

[12] K. Wang and S. J. Stolfo. *Recent Advances in Intrusion Detection*, chapter Anomalous Payload-Based Network Intrusion Detection, pages 203–222. 2004.

[13] K. Xinidis, I. Charitakis, S. Antonatos, K. Anagnostakis, and E. Markatos. An active splitter architecture for intrusion detection and prevention. *Dependable and Secure Computing, IEEE Transactions on*, 3(1):31–44, Jan.-March 2006.