Entropy Based Analysis of Worm Attacks in a Local Network

P. Velarde-Alvarado[1,3], C. Vargas-Rosales[2], D. Torres-Román[1], D. Muñoz-Rodríguez[2]

[1] CINVESTAV-IPN, Guadalajara, Jal., México
Department of Electrical Engineering and Computer Sciences, Telecommunications Section
{pvelarde, dtorres}@gdl.cinvestav.mx

[2] Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey, N.L., México
Center of Electronics and Telecommunications
{cvargas, dmunoz}@itesm.mx

[3] Universidad Autónoma de Nayarit, Tepic, Nay., México
Department of Electronics

Abstract. Network worms are a threat to the functionality and security of large networks such as the Internet. This threat is increased when Local Area Networks are used as platforms to propagate massively new infections. In this work, a LAN is subjected to real attacks of the worms W32.Blaster and W32.Sasser. A forensic analysis based on the estimation of the entropy for traffic traces indicate that the ensembles show anomalous behavior during the infection. The analysis suggests that the statistical tools, e.g. mean interquartile (IQR), and the correlation coefficient capture these anomalies. The analysis introduced is robust to the size of the measurement periods used in the traces.

1. Introduction

Internet has become a vital part of the activities of many individuals, and organizations. The benefits of its infrastructure have revolutionized the way in which communications are carried out. However, associated with this success, it has had to face a diversity of network-based attacks, such as viruses, worms, DDoS, etc. Worm spreading is an important issue, since they cost billions of dollars each year. Further, the risks of new outbreaks are always present.

Traditionally, worm research has been focused on large-scale networks, the contributions of these investigations are clear; however it is important to note that worms using local scanning techniques have been very effective in their attacks. Localized scanning [1], gives preference to targets in the address space that is close to the victim. An infected host scanning for targets in the local space, has a higher probability to infect than if the scan is made in a random mode within the whole address space. Accordingly, a localized-scanning worm could infect a local network very fast, and as a consequence, could use the resources of the infected network to launch a massive attack to external networks. Therefore, it is important to design mitigation systems that could early diminish worm spreading in LANs.

© G. Sidorov, B. Cruz, M. Martínez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 34, 2008, pp. 225-235

Received 21/03/08
Accepted 26/04/08
Final version 03/05/08

A work related to worm detection in LANs is that in [2], which describes a method to minimize the damage caused by worms, employing fuzzy analysis of three parameters that are affected: openness, homogeneity and trust. In [3], authors suggest a two-step algorithm for detecting victims of worms based on the behavior in terms of the pattern of infection and scanning. With respect to the use of entropy for the anomaly detection in network traffic, [4] proposes a generalization of the Shannon entropy, called Nonextensive Entropy. Also, [5] develops an entropy-based approach that determines and reports entropy contents of some traffic parameters like IP addresses and ports. They use several compressors, to obtain entropy. In our case, we determine the entropy contents of traffic parameters using a maximum likelihood entropy estimator with no compression and we analyze the correlation of such parameters. In [6], a behavior-based anomaly detection method is developed by comparing the current network traffic to a baseline distribution.

In this paper, the analysis is based on forensic evidence obtained in experiments conducted in a LAN, with an attack at a real academic network under a controlled environment. This forensic analysis helps us to determine anomalies to declare an infection status of the network. We captured the traffic for six days of normal operations, and then we released separately two TCP worms, *W32.Blaster.Worm* and *W32.Sasser.Worm*. The traffic was processed, and analyzed for the identification of anomalies based on the maximum likelihood entropy estimator of the variables studied. The objective is to generate standard and anomalous behavioral profiles of the network under normal conditions and under attacks, to be incorporated into a method for detection. Estimating entropy, we show that there is correlation of source IP addresses and those of the destinations, with high variability between results for a normal operation, and those during the worm attacks.

The paper is organized as follows. In Section 2, we give some basic concepts about worms. Section 3 gives a mathematical abstraction for the captured traces needed for our analysis. A discussion of the relation of entropy and worm detection is given en section 4. Sections 5 and 6 describe our test environment and results. Section 7 gives concluding remarks.

2. Worms

A worm is a *malware* or malicious code designed to self-propagate through the network, infiltrating vulnerable systems to exploit their security holes. The fundamental difference between a conventional virus and a worm is that the latter is autonomous, i.e., it does not require of the human interaction to carry out its dispersion. A worm uses network connectivity to transfer copies of itself from a host infected to a vulnerable host, and to search for new victims through a process called target discovery. Vulnerability is the necessary condition for a healthy host to become a victim. Based on how worms are transmitted, there are TCP worms and UDP worms. The major difference between these two types of worms is that TCP worms are latency-limited and UDP worms are bandwidth-limited [7].

2.1. Worm Components

Worms have a modular structure to aid propagation which consists of: *Entry mechanism*, which exploits known or unknown vulnerabilities in services to gain access into the target system. Some techniques that worms could use are: buffer overflow exploits, file–sharing attack, and e-mail attachments. *Propagation mechanism*, after gaining access to the target via the entry mechanism, the worm must transfer the rest of its body to the target. File transfer mechanisms (e. g. FTP, TFTP, HTTP) are most popular propagation methods utilized by this mechanism. *Target discovery*, once the worm is running on the victim machine, the target discovery starts looking for new victims to infect. There are many different methods to infect the next victim. For instance, sequential, permutation, and random scanning belongs to the scheme of blind scanning. An improved version of this scheme is the localized scanning, which uses information from the victim to conduct the scanning to addresses near to the victim, improving the hit rate of the scanning. In [7, 8] are explained more details about Target discovery techniques. *Payload*, is the code responsible for carrying out a specific task on behalf of the attacker on a target system (e. g. opening up a backdoor, planting a DDoS, etc.).

2.2. W.32.Blaster.Worm

W32.Blaster.Worm [9], propagates by exploiting the DCOM RPC Interface Buffer Overrun Vulnerability. It uses a routine that optimizes spread infection in the networks closer to the infected host. Blaster attempts to infect sequential IP addresses endlessly. Each time a host is infected, there is a 40% chance that it will begin at the first address of its "Class C"-size subnet (x.x.x.0), and a 60% chance that it will start at a completely random IP address with the last octet set to 0. Once a starting address is determined, the worm attempts to probe blocks of 20 sequential IP addresses at a time for hosts with TCP port 135 open, by sending connection attempts to each one simultaneously. Subsequently, Blaster tries to send a payload exploiting the RCP vulnerability to the hosts, where a TCP connection successfully could be established. If its RPC service is vulnerable to the DCOM buffer overflow, the payload causes a command shell to be bound to port 4444 on the infected target. The shell only stays open for one connection, and will therefore be closed once the worm has finished issuing commands. After sending the payload, it assumes that a command shell is listening on the remote port 4444 and attempts to connect. If successful, it starts a TFTP server thread on the local machine and sends a command to instruct the remote machine to download a copy of the "msblast.exe" worm via TFTP. Once the executable has been transferred, or after 20 seconds have elapsed, the TFTP server is shut down and the worm then issues further commands to the victim to execute msblast.exe. Assuming the executable was downloaded successfully, the propagation cycle begins again from the recently infected host, while the infecting instance continues iterating through IP addresses, [20].

2.3. W32.Sasser.Worm

W32.Sasser.Worm [10], propagates by exploiting the LSASS. It is exploited by sending a specially crafted RPC request to the LSASS named pipe on machines listening on 445 TCP port. Upon successful exploitation, shell code is injected into the lsass.exe process, which executes a shell (cmd.exe) and binds it to 9996/TCP port. The attacking instance of the worm then connects to this port and sends commands to the shell to download and run the main worm executable on the recent infected system. The download is through FTP, using the default Windows ftp.exe on the client side (victim). On the server side (attacker), Sasser implements its own crude FTP server, which listens on 5554/TCP. The infection scheme of Sasser is very similar to that of W32/Blaster with the exception of using FTP instead of TFTP. Once it is running on a new machine, Sasser installs itself in the Windows directory under the name 'avserve.exe'. Then it simply tries to infect new systems. Sasser generates target IP addresses using three different methods: completely random IPs are used 52% of the time; random IPs located in the same /16 network as the host are used 27% of the time; and random IPs located in the same /8 network as the host are used 21% of the time.

3. Abstraction of the Captured Traffic

Consider the analyzed traffic trace χ of a duration of t_D seconds with a total of N packets, the set of packet time stamps is denoted by $T_a = \{t_0, t_1, \mathrm{L}\ , t_{N-1}\}$. χ is divided into M non-overlapping blocks of $t_d = \frac{t_D}{M}$ seconds each. The i-th block has W_i packets, $i = 1, 2, \mathrm{L}\ , M$, namely $N = \sum_{i=1}^{M} W_i$. Figure 1 shows this partition.

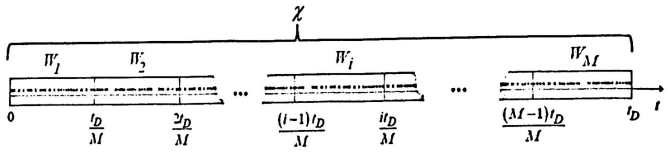

Fig. 1. Block representation for the analyzed traffic trace χ

In each i-block, we analyze four parameter ensembles denoted X_i^r, $r = 1, 2, 3, 4$. Each ensemble is related to a traffic parameter as follows: $X_i^1 \sim$ Source IP, $X_i^2 \sim$ Destination IP, $X_i^3 \sim$ Source port, and $X_i^4 \sim$ Destination port. An ensemble X_i^r is a triplet $(x, \mathrm{A}_{X_i^r}, \mathrm{P}_{X_i^r})$, where the outcome x is the random variable, which takes on one of a finite set of possible values in an alphabet,

$A_{X_i^r} = \{a_1, a_2, \mathrm{L}\ , a_J, \mathrm{L}\ , a_J\}$, having probabilities $P_{X_i^r} = \{p_1, p_2, \mathrm{L}\ , p_J\}$, with $P_{X_i^r}(a_J) = P(X_i^r = a_J) = P_J \geq 0$ and $\sum_{a_J \in A_{X_i^r}} P_{X_i^r}(a_J) = 1$. The Shannon information content of an outcome x is defined to be $h_{X_i^r}(x) = \log_2 \frac{1}{P_{X_i^r}(x)}$. The entropy of an ensemble X_i^r is defined to be the average Shannon information content of an outcome, i.e.

$$H_i^r(X_i^r) \equiv \sum_{x \in A_{X_i^r}} P_{X_i^r}(x) \log_2 \frac{1}{P_{X_i^r}(x)}, \tag{1}$$

where we consider that for $P_{X_i^r}(x) = 0$, $0 \log_2 \frac{1}{0} \equiv 0$, since $\lim_{\theta \to 0^+} \theta \log \frac{1}{\theta} = 0$. A property of the entropy function is $0 \leq H_i^r(X_i^r) \leq \log_2 \left(\left| A_{X_i^r} \right| \right)$. In order to obtain a likelihood estimator of H_i^r, the discrete probabilities $P_{X_i^r}$ in (1) are replaced by maximum likelihood estimates $\hat{P}_{X_i^r}$. Specifically, if we consider i-blocks of W_i packets or observations, and let n_j be the frequency of the value $a_j \in A_{X_i^r}$ in the ensemble X_i^r, then, with the choice $\hat{P}_{X_i^r} = \frac{n_j}{W_i}$ we have an estimate

$$\hat{H}_i^r(X_i^r) \equiv \sum_{x \in A_{X_i^r}} \hat{P}_{X_i^r}(x) \log_2 \frac{1}{\hat{P}_{X_i^r}(x)}. \tag{2}$$

(2) is the maximum likelihood entropy estimator. Now, during the time of the i-th block, $\left(\frac{(i-1)t_D}{M}, \frac{it_D}{M} \right)$, we have four measures of traffic in terms of entropy that helps us to construct a representative matrix with dimension $4 \times M$ for the trace χ.

$$\hat{\mathbf{H}}(\chi) = \begin{pmatrix} \hat{H}_i^1 \\ \hat{H}_i^2 \\ \hat{H}_i^3 \\ \hat{H}_i^4 \end{pmatrix} = \begin{pmatrix} \hat{H}_1^1 & \hat{H}_2^1 & \mathrm{L} & \hat{H}_M^1 \\ \hat{H}_1^2 & \hat{H}_2^2 & \mathrm{L} & \hat{H}_M^2 \\ \hat{H}_1^3 & \hat{H}_2^3 & \mathrm{L} & \hat{H}_M^3 \\ \hat{H}_1^4 & \hat{H}_2^4 & \mathrm{L} & \hat{H}_M^4 \end{pmatrix}. \tag{3}$$

With (3), we have four traffic vectors corresponding to the entropy of a given feature (i. e. source and destination IP address and source and destination port).

4. Entropy and Worm Propagation

Recently, the concept of entropy has been applied to the development of a new generation of network security systems. These systems must rapidly detect (in the order of seconds) a wide variety of intrusions of worms and other attacks, with a low rate of

false positives and negatives [11]. This contrasts with conventional systems, that focus primarily on changes in the volume of traffic at specific points of the network and whose system's response might be hours or even days. The application of the entropy on these new systems is based on the feature that worm's activity changes the characteristics of the traffic in measurable ways, these changes alter the entropy of the network. The entropy of the network is a measure fluctuating between different levels of predictability and randomness caused by changes in the nature and composition of traffic. By building profiles of the network's normal behavior based on the entropy of traffic features such as the source and destination IP addresses, the source and the destination port numbers, or even the type of protocol, the number of bytes, and the inter-packet times, it is possible to define a baseline to verify anomalies correlating it with current traffic features.

5. Empirical Analysis

This empirical analysis is based on collected data in a LAN operating under normal conditions and under the attack of two worms: W32.Blaster and W32.Sasser. Network Security and privacy policies applied for some projects who share traces for research purposes, limit the availability of traces suitable for accurate characterization of anomalous traffic caused by worms. This poor availability of suitable traces led us to carry out a real worm attack on a LAN to observe the behavior patterns of traffic under normal conditions for several days and later under worm attacks.

5.1. Network Environment

The worm propagation has been tested in a class C IP network subdivided into four subnets. There are 100 hosts running Windows XP SP2 mainly. Two routers connect the subnets with 10 Ethernet switches and 18 IEEE 802.11b/g wireless access points. The data rate of the core network is 100Mbps. A sector of the network is left vulnerable on purpose, with ten not patched Windows XP stations. In the experiments Blaster and Sasser worms where released in the vulnerable sector.

5.2. Data Collection and Tools

The data-set was collected by a network sniffer tool based on *libpcap* library used by *tcpdump*, [12]. This data-set contains traces corresponding to a six day period of standard-traffic in user's typical work hours, and one standard-traffic trace combined with anomalous traffic of the day of the attacks. All traces were sanited to remove spurious data using *plab* a platform for packet capture and analysis, [13]. Traces were split in segments using *tracesplit* which is a tool that belongs to *Libtrace*, [14]. The traffic-files in ASCII format suitable for MATLAB processing were created with *ipsumdump* [15].

6. Test Results

Our methodology is as follows. We first capture and process the traces for the analysis using tools mentioned in section 5.2. These traces were classified into two groups: the first group consists of standard traffic during normal working hours. The second group consists of three sub-traces of the day of the attacks: P1, corresponding to approximately 70 min. of benign traffic previous to the first attack, P2, which is the traffic during the Blaster attack and, P4, which is during the Sasser attack.

Secondly, we analyze separately the two sets of traces to obtain the following: (a) the average value of the entropies of the four ensembles, (b) the correlation coefficients between ensembles of addresses and between ports, and (c) the average values of IQR (interquartile range) for entropies of ensembles obtained with different window sizes. The analysis of the first group of traces helped us to have a baseline of the network under normal conditions (i.e. free of worms), which is summarized in Tables 1 and 2. The second group of traces allowed us to observe the behavior of the traffic after the release of worms and is summarized in Tables 3 and 4.

In the next phase, we analyzed three behavioral symptoms in traffic: changes in the value of the entropy of the ensembles, the correlation coefficient of the entropy of the ensembles and the changes of the size of the boxes in the boxplots used for the estimation of entropy with different window sizes (this through IQR).

For the first symptom, Figure 2 can be seen as a timeline which shows the behavior of traffic under standard conditions (sub-trace P1) and under attack by the Blaster and Sasser worms (sub-trace P2 and P4 respectively), the most evident change between normal and anomalous is the abrupt change in the value of entropies subsequent to the release of worms. These changes in entropy values can be quantified using the results of Tables 1 – 4. We note that the average value of the ensemble of source directions in benign traffic (column 2 of Table 1) is 3.8 bits and during the Blaster worm attack declined to 2.12 (Table 3). For the Sasser worm rose to 5.78 bits (Table 3). In the case of ensembles of destination addresses under normal conditions the average entropy was 3.5 bits (Table 1), but during the attacks rose to 13.8 under the Blaster worm and 10.36 under Sasser (Table 3).

Table 1. Entropy values and correlation coefficient during standard traffic.

Trace of day	Mean of entropy (bits)				Correlation coefficient	
	$\hat{H}_i^1$	$\hat{H}_i^2$	$\hat{H}_i^3$	$\hat{H}_i^4$	$\hat{H}_i^1$ vs. $\hat{H}_i^2$	$\hat{H}_i^3$ vs. $\hat{H}_i^4$
1	3.78	3.46	2.96	4.52	0.80	0.41
2	3.82	3.35	2.66	4.57	0.90	0.40
3	3.62	3.20	2.66	4.29	0.94	0.56
4	3.71	3.42	2.95	4.36	0.90	0.45
5	3.64	3.45	2.94	4.39	0.78	0.46
6	4.17	4.14	4.13	4.41	0.98	0.93

The second symptom studied is associated with correlation coefficient of entropy assemblages of source and destination addresses and ports. The statistical analysis

using bootstrap, for the correlation coefficient calculation, of the network behavior of the six days of benign traffic during normal operation, allows us to identify a linear relation between the entropies of the ensembles, i.e. $\hat{H}_i^1$ vs. $\hat{H}_i^2$ and $\hat{H}_i^3$ vs $\hat{H}_i^4$, as it is shown in Table 1. We can see that all the coefficients are positive, and some close to one. However, when we apply the same analysis for segments P2 and P4, we can see in the last two columns of Table 3 that these coefficients become negative and some close to minus one. This can also be used as an indication of the anomalous behavior present during an attack.

Table 2. Mean values for IQR calculated in 10, 20,30, 60, 120, 180, 240, and 300 sec for standard traffic days

Trace of day	$IQR(\hat{H}_i^1)$	$IQR(\hat{H}_i^2)$	$IQR(\hat{H}_i^3)$	$IQR(\hat{H}_i^4)$
1	1.15	0.90	0.93	1.47
2	1.48	1.05	0.72	1.79
3	1.48	1.01	1.01	1.71
4	1.64	1.15	1.19	1.49
5	1.20	1.02	1.20	1.38
6	1.00	1.04	0.9	1.13

Table 3. Entropy values and correlation coefficient during the day of the attack for the segments P1, P2, and P4

Sub-trace	Mean of entropy (bits)				Correlation coefficient	
	$\hat{H}_i^1$	$\hat{H}_i^2$	$\hat{H}_i^3$	$\hat{H}_i^4$	$\hat{H}_i^1$ vs. $\hat{H}_i^2$	$\hat{H}_i^3$ vs. $\hat{H}_i^4$
P1	3.24	3.24	3.62	3.80	0.96	0.97
P2	2.12	13.18	4.27	7.67	-0.95	-0.3
P4	5.78	10.36	9.1	3.82	-0.45	-0.9

For the third symptom, we analyzed the traces using boxplots which help us to observe other characteristic behavior. In Figures 3 - 5 and Table 2 and 4, we could see the mean values of IQR of the ensemble entropies of the four parameters. In Figure 3 and Table 2, we have the mean IQR for standard traffic for different values of t_d, we take these as the baseline behavior to compare to those mean IQR for P2 and P4 traces. For the Blaster attack, shown in Figure 4 and the second row of Table 4, we could see that there is a significant decrease in the mean value of IQR with respect to that of segment P1. It can also be seen in Figure 5 and the third row of Table 4 that for trace P4, there is a significant change in the mean value of IQR for $\hat{H}_i^3$, indicating that this parameter is adequate to capture the effect. Another observation is that entropy as t_d changes is robust, since regardless of this, the effects of the attacks are evident with an important decrease in the size of the boxes for the traces in P2 and P4.

Table 4. Mean values for IQR calculated in 10, 20,30, 60, 120, 180, 240, and 300 sec for day of the attaks

Sub-trace	$IQR(\hat{H}_i^1)$	$IQR(\hat{H}_i^2)$	$IQR(\hat{H}_i^3)$	$IQR(\hat{H}_i^4)$
P1	1.17	1.10	1.67	1.59
P2	0.25	0.73	0.21	0.11
P4	.92	0.78	0.90	1.25

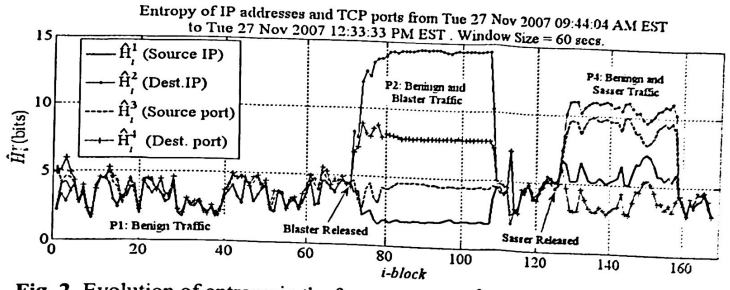

Fig. 2. Evolution of entropy in the four ensembles $\hat{H}_i^r$ of traffic during the day of worm attacks.

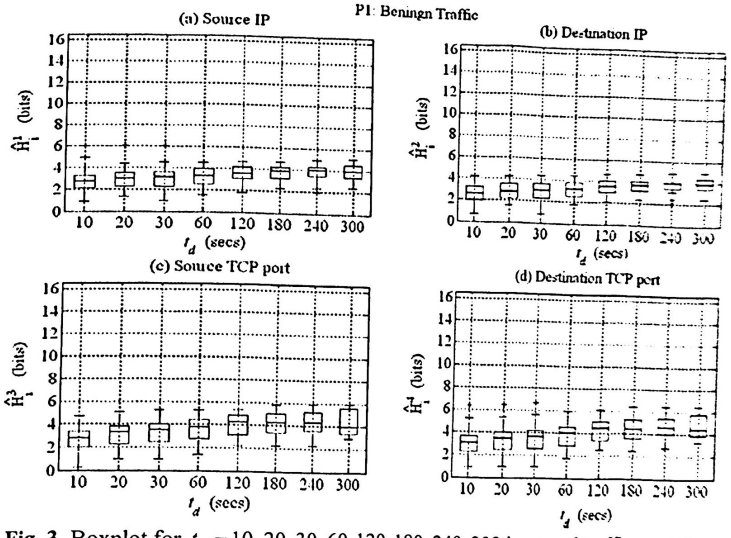

Fig. 3. Boxplot for $t_d = 10, 20, 30, 60, 120, 180, 240, 300$ in normal traffic conditions

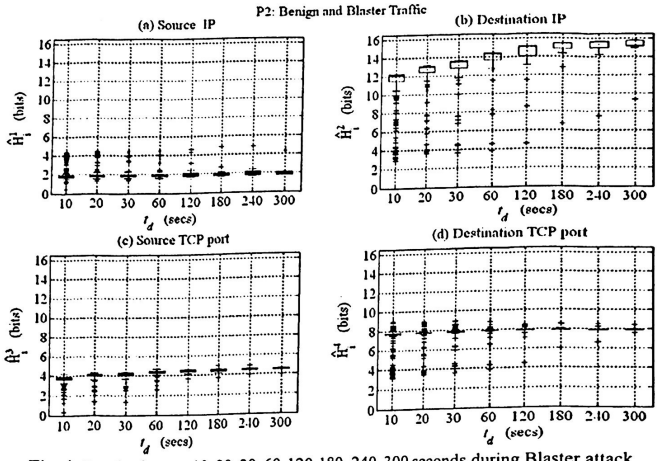

Fig. 4. Boxplot for $t_d = 10, 20, 30, 60, 120, 180, 240, 300$ seconds during Blaster attack

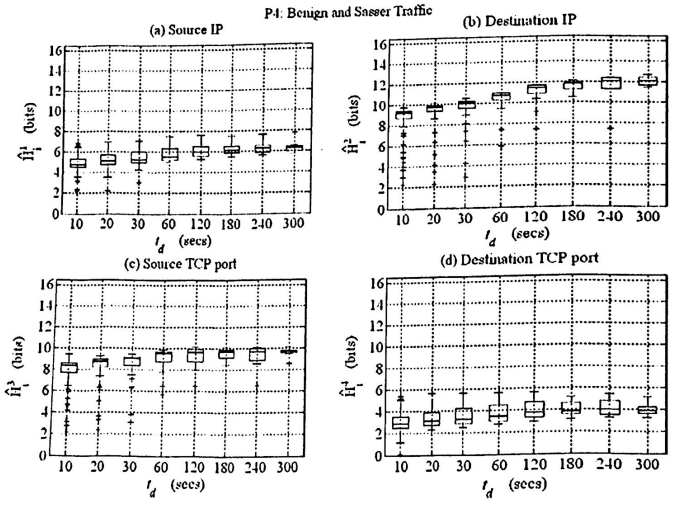

Fig. 5. Boxplot for $t_d = 10, 20, 30, 60, 120, 180, 240, 300$ seconds during Sasser attack

7. Conclusions and Final Remarks

We introduced a statistical analysis tool for traffic traces in a local network where ensemble entropy is calculated for four parameters of the traffic. We found that a significant variation in entropy is an effective way to identify the presence of an anomaly in the data set. The four feature entropy based detection also helps to identify worm outbreaks. Although what we introduced were conducted as a forensic analysis, we can conclude that the methodology could be used with some changes in an online fashion, due to the robustness of the observation time slot t_d. Analysis of a number of traffic traces suggested a relationship between the correlation coefficients and the mean values of IQR for the entropies, and these can be incorporated as a method to infer anomalous behavior that could represent a network worm attack.

Within the future work we are conducting is the mathematical modeling of the entropy and the traffic traces as the attacks are carried out.

References

1. Chen, Z., Chen, Z., Ji, C.: Understanding Localized-Scanning Worms. IEEE IPCCC, (2007) 186-193
2. Sanguanpong, S., Kanlayasiri, U.: Worm Damage Minimization in Enterprise Networks. International Journal of Human-Computer Studies. Vol. 65, No. 1. (2007) 3-16
3. Gu, G., Sharif, M., Qin, X., Dagon, D., Lee, W., Riley, G.: Worm Detection, Early Warning and Response Based on Local Victim Information. Proceedings of the 20th Annual Computer Security Applications Conference. (2004) 136-145
4. Ziviani, A., Gomes, A.T.A., Monsores, M.L., Rodrigues, P.S.S.: Network Anomaly Detection Using Nonextensive Entropy. Communications Letters, IEEE. Vol. 11, No. 12. (2007) 1034-1036
5. Wagner, A., Plattner, B.: Entropy Based Worm and Anomaly Detection in Fast IP Networks. 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise. (2005) 172-177
6. Gu, Y., McCallum, A., Towsley, D.: Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation. Internet Measurement Conference (2005) 345-350
7. Li, P., Salour, M., Su, X.: A Survey of Internet Worm Detection and Containment. Communications Surveys & Tutorials. Vol.10 No.1 (2008) 20-35
8. Weaver, N., Paxson, V., Staniford, S., Cunningham, R.: A Taxonomy of Computer Worms. ACM Workshop on Rapid Malcode. (2003) 11-18
9. Copley, D., Hassell, R., Jack, B., Lynn, K., Permeh, R., Soeder, D.: ANALYSIS: Blaster Worm. eEye Digital Security Research. http://research.eeye.com/html/advisories/published/AL20030811.html
10. Ukai, Y., Soeder, D.: ANALYSIS: Sasser. eEye Digital Security Research. http://research.eeye.com/html/advisories/published/AD20040501.html
11. Nucci, A., Bannerman, S.: Controlled Chaos. IEEE Spectrum. Vol.44. No.12. (2007) 42-48
12. Jacobson, V., Leres, C., McCanne, S.: Tcpdump/libpcap. http://www.tcpdump.org/
13. Peppo, A. plab. Tool for traffic traces. http://www.grid.unina.it/software/Plab/
14. Trac Project. Libtrace. http://www.wand.net.nz/trac/libtrace
15. Kohler, E. ipsumdump. Traffic tool. http://www.cs.ucla.edu/~kohler/ipsumdump/