Generating Realistic Network Traffic for Security Experiments

Conference Paper · April 2004

DOI: 10.1109/SECON.2004.1287918 · Source: IEEE Xplore

Gerald Marin
Florida Institute of Technology

Generating Realistic Network Traffic for Security Experiments*

Song Luo
University of Central Florida
Orlando, Florida
sluo@cs.ucf.edu

Gerald A. Marin
Florida Institute of Technology
Melbourne, Florida
gmarin@cs.fit.edu

Abstract

This paper reports results of an effort to develop a test environment in which "live" attack-free background traffic reflects the characteristics of the network to be defended. The expectation is that new intrusion detection techniques can be better evaluated (and tuned), in such a background, against inserted attacks and no others. Based on analysis of traffic captured from an example network in 2003, we determine models appropriate for the major Internet protocols present and compare these with previously obtained results. We describe the traffic modeling, and we describe an approach for generating realistic attack-free traffic (that is statistically similar to the captured traffic) in a test environment.

1. Introduction

Considerable research is being done on detecting network attacks using signature analysis techniques to find known attacks and a variety of approaches to discover previously unseen attacks [9], [10], [11]. A common requirement in such work is an isolated test bed with realistic network traffic that can be used to test and evaluate proposed network intrusion detection techniques. Although some recent commercial results have been promising, current IDS products are still addressing performance problems, high false-alarm rates, configuration and tuning problems, and even stability problems [12], [13]. Clearly more remains to be done and certainly there will remain a strong demand for good test environments.

Ideally, to evaluate an Intrusion Detection System (IDS) one needs realistic traffic that contains only known attacks. (The nature and location of the attacks should be known by testers; attacks should not all be known by IDS developers.) Furthermore, the traffic needs to be "live" so that one can better judge the system's ability to protect targeted test-bed systems. In addition, the non-attack traffic ideally *should reflect the characteristics of typical traffic of the defended network* because today's tools generally need to be tuned to a particular environment in order to mitigate false alarms.

In pursuit of such an environment we have been analyzing the characteristics of the CS LAN traffic at UCF to identify the major contributors to that traffic, to model the corresponding statistical distributions, and to measure key distribution parameters. The intent is to model a majority of the traffic threads present in the original LAN traffic and emulate the modeled sources of traffic while targeting them towards servers that are present in the lab environment. In related work, we are also developing state-based methods for generating attacks that can be inserted as the modeled traffic is replayed in the emulated LAN environment. This paper, however, will focus on the background traffic analysis and modeling.

2. Related Work

Under a grant received from the Defense Advanced Research Projects Agency (DARPA), researchers from MIT's Lincoln Laboratory used an approach quite similar to the one mentioned above to create an environment to test a number of IDS systems in 1998 and 1999 [2], [3]. The resulting traffic set has been valuable to a number of researchers [1]. It includes periods that are certified to be attack free and periods where attacks are included and documented. During the tests that were part of the original research, developers also used a dataset that contained unidentified attacks; thus, the results of various IDS approaches were compared against both known and previously unseen attacks. The datasets are available to other interested researchers and include several days of emulated attack-free traffic plus several days where the traffic is a mixture of background traffic, attack data, and the responses of target machines.

The background traffic in the Lincoln Lab experiments was based on the analysis of samples of 4 months of network traffic from 50 Air Force bases. This was a substantial, well documented, and widely leveraged effort; but, of course, it had its limitations. These have been

*This work was sponsored in part by a grant from the Office of Naval Research.

0-7803-8367-2/04/$20.00 ©2004 IEEE

analyzed in [3] and [4]. We mention here, briefly, only those that pertain to the generation of the background traffic. First, Lincoln Lab researchers noted that the generated traffic was similar to that of measured Air Force traffic, but they did not sufficiently describe how that similarity was demonstrated. Second, they reported, for example, that Poisson distributions were used to simulate session arrival time, but many other statistical variables (such as connection size, and idle times between connections) were not given. Third, data rates used in these experiments seemed to be low. The analysis in [4], for instance, shows data rates ranging from 8.8 kbps to 51.4 kbps.

In spite of these limitations, the Lincoln Lab effort demonstrated a set of techniques that could be used to generate traffic plus attacks that (1) represented a particular networking environment, (2) provided for "live" testing, and (3) ensured that known attacks *and only known attacks* were present in the traffic. In this paper we leverage a similar approach, base our analysis on "live" traffic headers collected from the CS LAN at UCF and carefully document the resulting statistical distributions. As will be seen, we also leverage some of the other classical work in traffic modeling.

3. Classic TCP Traffic Distribution Models

Perhaps the most frequently referenced models of TCP connection distributions are those from Paxson [5], [6]. Paxson and his colleagues examined 3 million TCP connections from a number of wide-area traffic traces and a variety of sources. Some analytical models were derived to describe the random variables associated with TELNET, NNTP, SMTP and FTP connections [5]. In [6], it was found that, except for user-initiated TCP session arrivals, other TCP connection arrivals were not Poisson. Table 1 summarizes the models used in [5] and [6] to describe the traffic characteristics of different services

Table 1: Summary of previous models

Protocol	Variables	Model
TELNET	Session arrivals	Poisson
	Originator bytes	Log2-extrem
	Responder bytes	Log2-normal, 80-100%
	Conn size (in packets)	Log2-normal
	Packet interarrivals	Empirical, from Tcplib
SMTP	Session arrivals	Poisson
	Originator bytes	Log2-normal
FTP	Session arrivals	Poisson
	Connection bytes	Log2-normal
	FTP-data spacing	Log-normal/log-logistic
	Session bytes	Log2-normal

Notably missing from Table 1 (and from the referenced work) is any reference to HTTP. The worldwide web (WWW) application has become a major force in the Internet but only in years after the work done in [5] and [6]. A complete WWW traffic simulation for wireless system was reported in [7] and [8], where a page-oriented model is presented. This research led to the results presented in Table 2.

Table 2. Classic distribution model for HTTP

Parameter	Distribution
Session interarrival time	Poisson/Exponential
Pages per session	Lognormal
Time between pages	Gamma
Page size	Pareto
Packet size	Multimodal
Packet interarrival time	Exponential

4. LAN Traffic Analysis

As stated previously, our intent is to develop methods for modeling a given network's non-attack background traffic sufficiently so that we can emulate the environment for network security testing. In this paper we are using the CS LAN traffic from UCF for illustration. To this end we captured protocol headers from millions of Ethernet frames during a two-day period in February 2003. For our initial effort we restricted our attention to TCP traffic (which represented the vast majority). The CS department uses NFS (Network File System) and an auxiliary server to reduce the load on its main email and file server. Thus, all email, FTP, and web traffic that is addressed to the main server is automatically redirected to port 2049 of the NFS server. Once we understood this, we eliminated the duplication traffic on port 2049 from further consideration. The CS LAN also supports significant printer sharing with the network printer service at port 9100. We also eliminated this traffic prior to modeling. We also use SNORT [11] to test for the presence of known attacks whose intensities might affect the statistics. Table 3 gives an overview of the remaining traffic statistics from the two days.

In the remainder of this paper we report the results obtained by building empirical distributions for FTP, SSH, SMTP, POP3 and HTTP and comparing them with the classical results. Results are mixed (based on chi-square values). We close by using the classical models to emulate the FTP background traffic streams and comparing these simulated results again with the measured results. It may well be that less than perfect statistical results are nevertheless useful for background traffic generation.

Table 3. IP Traffic on Day 1 and Day 2

Day 1 - Feb. 05			Day 2 - Feb. 07		
PKTS	32,070,078		PKTS	20,162,983	
	IP	32,070,078		IP	20,162,983
	ICMP	29,197		ICMP	26,694
	UDP	1,740,446		UDP	1,631,138
	TCP	30,300,214		TCP	18,504,809
BYTES	19,857,363,053		BYTES	13,295,981,557	
	ICMP	2,994,825		ICMP	2,658,906
	UDP	772,239,405		UDP	657,411,561
	TCP	19,082,115,557		TCP	12,635,890,568
TCPCONN	212,716		TCPCONN	170,550	
	HTTP	92,034		HTTP	78,581
	SMTP	5,520		SMTP	5,214
	FTPDATA	2,236		FTPDATA	3,817
	TELNET	39		TELNET	37
	FINGER	8		FINGER	10
	FTP	435		FTP	466
	POP3	13,586		POP3	12,018
	TIME	2		TIME	2
	SSH	833		SSH	690
	IRC	0		IRC	0
	IDENT	315		IDENT	246

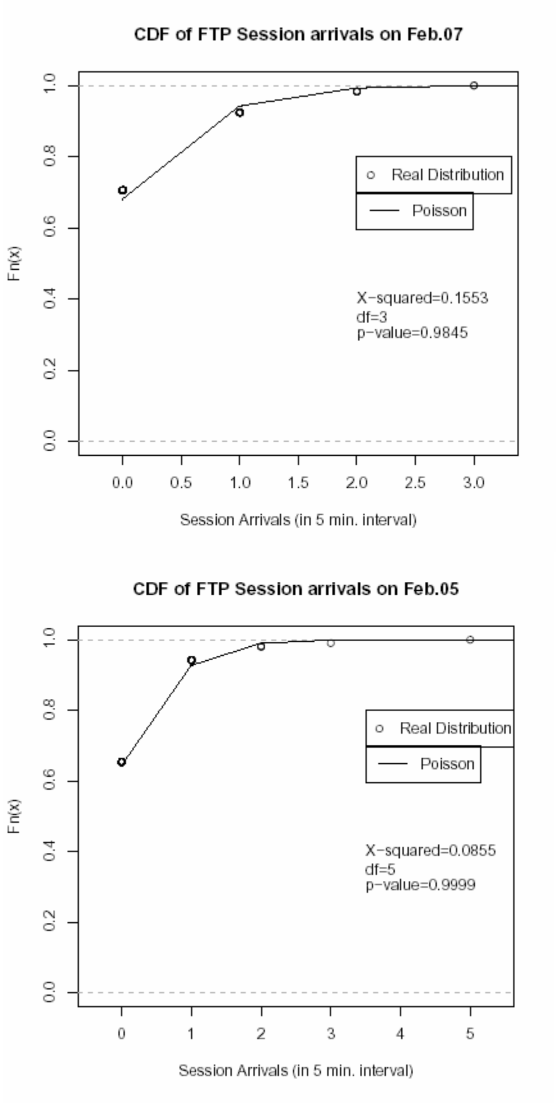
4.1 FTP Traffic

Each FTP session includes an FTP control connection and either zero, one, or multiple FTP-DATA connections in “active” or “passive” mode, as described in [5]. In this case we are interested in the appropriate distributions for FTP session arrivals, idle-time between adjacent FTP-DATA connections, number of bytes transferred during a single FTP-DATA connection, and number of bytes transferred during a whole FTP session. We refer to the former bytes as FTP-DATA bytes, and the latter bytes as FTP session bytes. Figure 1 depicts both the empirical cumulative distribution functions (CDF) for FTP session arrivals and the predicted Poisson arrivals first for Day 1 and then for Day 2. These confirm the classical results. Figure 2 depicts corresponding results for the total bytes per FTP session (comparing, in this case, with the predicted $\log_2 - \mathrm{normal}$ distribution). According to the chi-square tests, Day 1 compares favorably and Day 2 does not.

To examine the idle time between data connections within a single FTP session one must recognize that the transfers may be driven manually or by software. In the first case, the user just types the FTP command in a FTP client shell, and each subsequent command is typed by hand. In this case, the idle times between two continuous data transfers are usually more than one second. In the second case, FTP client software tools, such WS-FTP and SmartFTP, transfer multiple files in response to drag-and-drop selections by a user. In this case the idle times are, generally, much less than one second.

Based on this observation, we divide our collected FTP-DATA idle time in two groups. One group contains idle times greater than or equal to one second. The second contains idle times less than one second. Figure 3 shows that Day 1 results from the first group match the Gamma distribution (not the anticipated log-normal) and results from the second group do not.

Figure 1: CDFs of FTP session arrivals from Day 1 and Day 2 of captured traffic compared with predicted Poisson arrivals.



4.2 HTTP Traffic

Figures 4 and 5 depict the results of the HTTP session arrival comparisons. HTTP session arrivals are predicted to be Poisson; our results show that this depends on the time intervals used for counting arrivals. For example, Figure 4 shows the cumulative distribution functions of HTTP session arrivals from the two days' traffic over 5-minute intervals, while Figure 5 depicts the same data on 1-minute intervals. While neither passes chi-square, it is visually apparent that the distribution with 1-minute intervals compares to Poisson more favorably. In the interest of brevity we do not show page or session statistics, which were visually appealing, but did not pass the chi-square tests.

Figure 2: CDFs of FTP session bytes from Day 1 and Day 2 of captured traffic compared with the predicted Log2-Normal distribution.

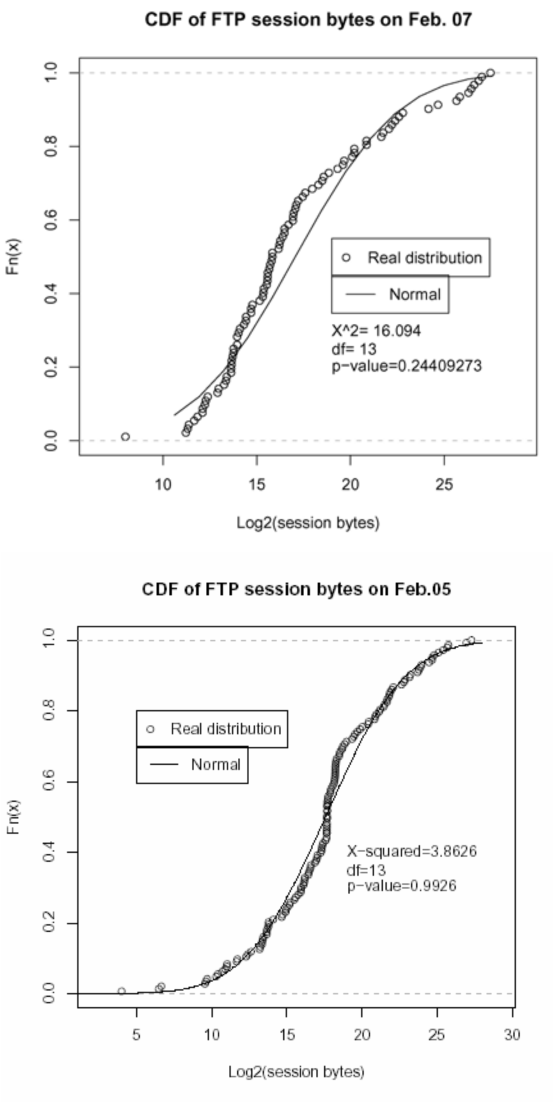
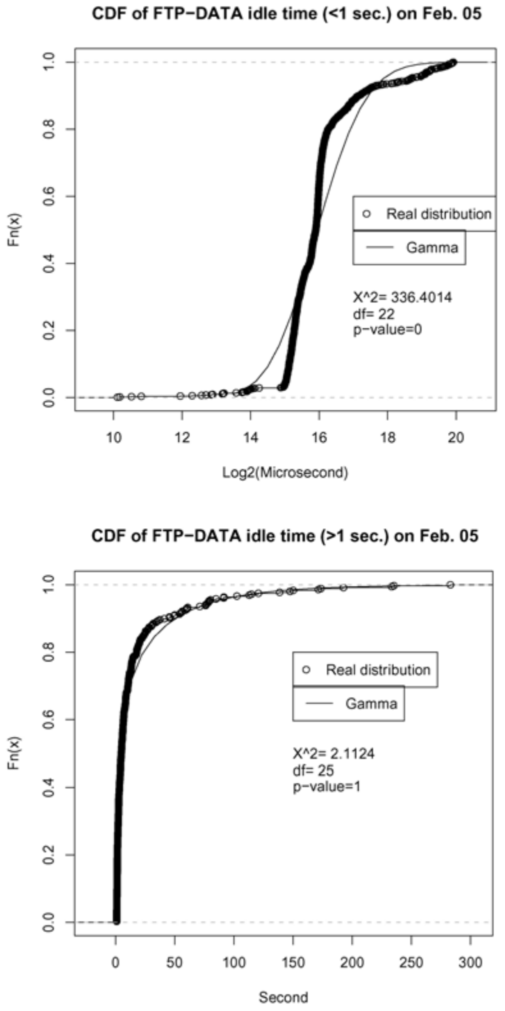


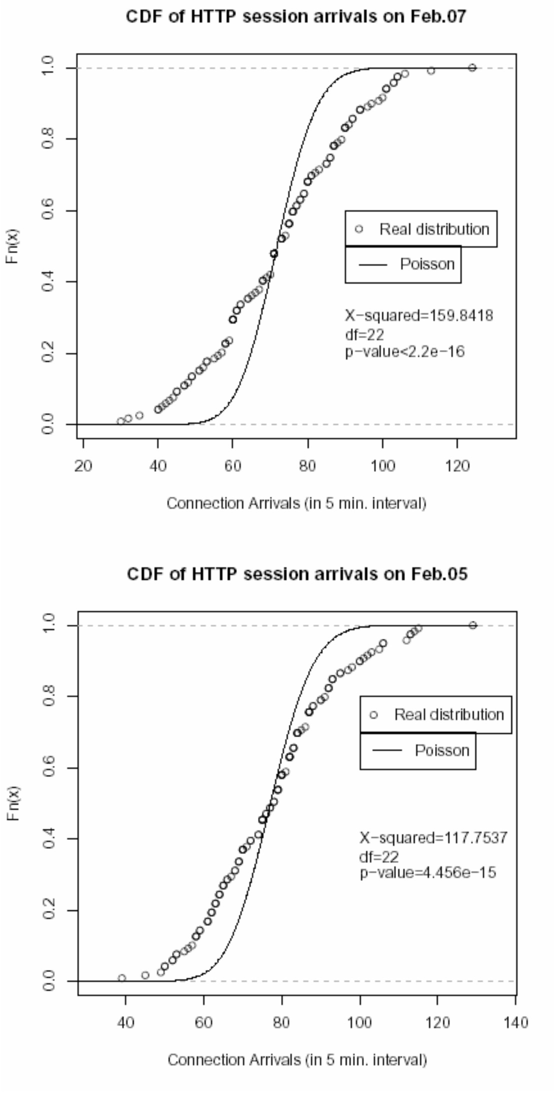
Figure 3: CDFs of FTP-DATA idle time from Day 1 of captured traffic compared with expected Gamma distribution.



4.3 Other Results

We used the same methodology to compare empirical versus "classical" distributions for SMTP connection arrivals and connection bytes, for POP3 connection arrivals and sizes, and for SSH statistics. Although some of the detailed results are interesting (and will be made available to interested readers upon request), it is not possible to present the details here. Further experimentation caused us to modify the classical results somewhat (and this may be needed for any particular environment being modeled). Table 4 summarizes the results that we conclude to be appropriate for the CS LAN modeling after further experimentation.

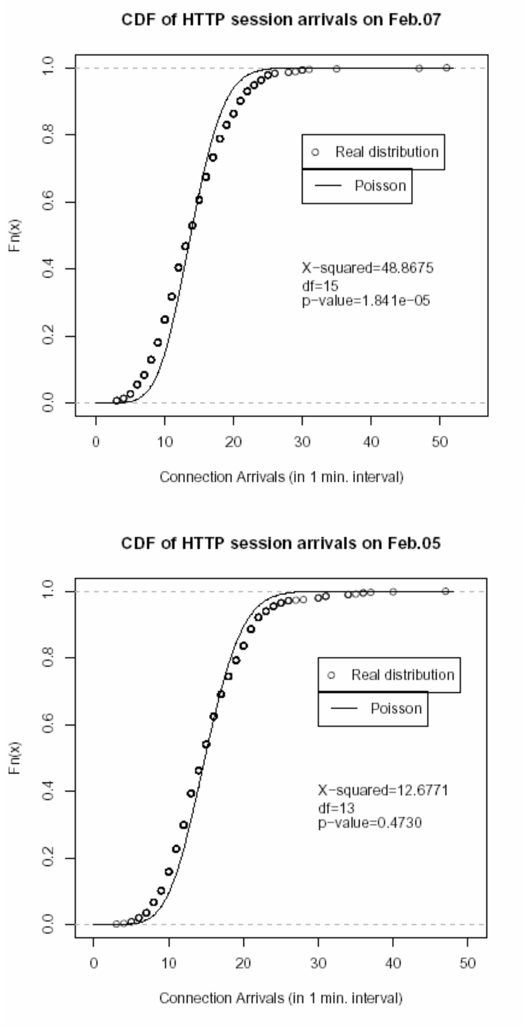
Figure 4: CDFs of HTTP session arrivals from Day 1 and Day 2 traffic compared with Poisson distribution.



5. Traffic Generation

Having determined the models that are appropriate to simulate this particular traffic, we need to estimate key distribution parameters from the captured traffic to drive the simulations over time. To accomplish this we divide the captured traffic into intervals of appropriate length (determined in the distributional analysis). Then we run programs developed to estimate parameters of all random variables for each time interval, using the predefined models.

Figure 5: CDFs of HTTP session arrivals from Day 1 and Day 2 traffic compared with Poisson.



Using the appropriate parameter estimates in each case, we launched five different traffic generators for FTP, SSH, SMTP, POP3 and HTTP changing the parameter estimates at appropriate intervals. To model SMTP, for example, the SMTP generator reads the adjusted mean arrival rate each minute and generates actual arrivals randomly. Upon session arrival, a Log-normal process (with parameters adjusted on 15-minute intervals) determines the number of bytes for the next connection. This number is input to another program that generates actual SMTP conversations with a remote server and delivers an email of the correct length to that server.

Figure 6 shows the comparison between the FTP traffic simulated using these techniques and the actual traffic captured on the CS LAN on Feb 5 of 2003. We note that actual traffic is sent in the simulations (as described above) so that server responses, ICMP control traffic and other network events can be included as realistically as possible. The expectation is that this traffic (added to traffic of the other dominant protocols) will ultimately form a background suitable for a number of testing purposes. Thus, even though not all of the distributions pass rigorous chi-square tests, it appears that the combined background will support a test environment useful, for example, for developing new IDS techniques.

Table 4: Revised Models for Network Traffic

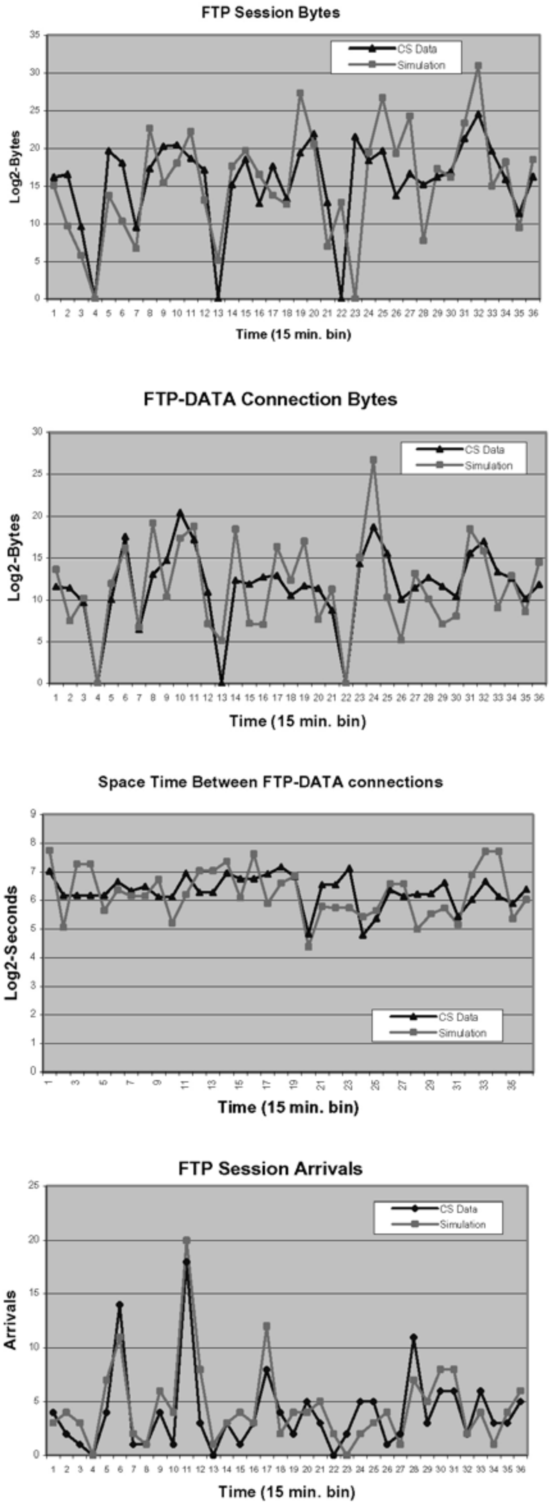
Protocol	Variables	Model
SSH	Session arrivals	Poisson (1-min. bin)
	Orig pkt interarrivals	<100us, normal >100us, log10-normal
	Response size (in bytes)	Log2-normal
	Conn size (in packets)	Log2-normal
SMTP	Session arrivals	Poisson (1-min. bin)
	Conn size (in bytes)	>=500 bytes, Log2-no <500 bytes, invalid
POP3	Session Arrivals	Poisson
	Connection Bytes	"Loaded", Log2-norm "Unloaded", Gamma
FTP	Session arrivals	Poisson
	FTP-data size (in bytes)	Log2-normal/Empiric
	FTP-data idle time	Gamma
	Session size (in bytes)	Log2-normal
HTTP	Session arrivals	Poisson (1-min. bin)
	Pages per session	Log-Gamma
	Page size (in bytes)	Log2-normal
	Page spacing	Exponential

6. Summary

The problem of generating realistic attack-free Internet traffic with characteristics similar to that of a given network is challenging. It requires careful modeling and parameter estimation at least for the dominant protocols. In pursuit of this goal the work described in this paper reviewed classical models for traffic of major Internet protocols, compared them with traffic recently captured from a local area network at UCF, and obtained new models to replace ones which obviously did not match the CS traffic.

Our analysis found that (with proper choice of time interval) arrival times for user-initiated sessions are still typical Poisson processes, including session arrivals of all protocols discussed in this paper. Smaller size (1-min.) bins are desired for protocols with high frequency, such as HTTP and SSH; while larger size (5-min.) bins are better for lower frequency protocols (like FTP).

Figure 6: Simulation of FTP Processes Compared with Actual Data.



FTP Session Bytes

FTP-DATA Connection Bytes

Space Time Between FTP-DATA connections

FTP Session Arrivals

The bytes-per-session or per-page usually match a Log-Normal distribution. However, it may be important to divide the overall range of sizes into sub-intervals. For example, we found that the bytes transferred during a POP3 connection were modeled differently in different ranges. Random variables related to time spacing exhibit the greatest diversity. Time interarrivals between SSH client packets seems to be Normal/Log-Normal; the idle time of FTP-DATA connections is Gamma; time spacing between HTTP pages is Exponential.

We illustrate the use of these methods to generate realistic traffic using the FTP protocol. Results are generally encouraging to this point and the work continues.

7. References

[1] William H. Allen and Gerald A. Marin, “On the Self-Similarity of Synthetic Traffic for the Evaluation of Intrusion Detection Systems. In Proceedings, IEEE/IPSJ International Symposium on Applications and the Internet (SAINT), 2003.

[2] R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkle, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogrod, M. A. Zissman, "Evaluating Intrusion Detection Systems without Attacking your Friends: The 1998 DARPA Intrusion Detection Evaluation", SANS 1999.

[3] Joshua W. Haines, Richard P. Lippmann, David J. Fried, Eushiuan Tran, Steve Boswell, Marc A. Zissman, "1999 DARPA Intrusion Detection System Evaluation: Design and Procedures,” MIT Lincoln Laboratory Technical Report.

[4] John McHugh, “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 (TISSEC) November 2000, Volume 3 Issue 4.

[5] V. Paxson, “Empirically derived analytic models of wide-area TCP connections,” Networking, IEEE/ACM Transactions on, Volume: 2 Issue: 4, Aug. 1994, Page(s): 316 –336.

[6] V. Paxson and S. Floyd, “Wide Area Traffic: the Failure of Poisson Modeling,” Networking, IEEE/ACM Transactions on, Volume: 3 Issue: 3, June 1995, Page(s): 226 –244.

[7] A. Reyes-Lecuona, et al, ``A Page-oriented WWW Traffic Model for Wireless System Simulations,” 16th International Telegraphic Congress, Vol 2, pp 1271-1280, Edinburgh, June, 1999.

[8] Eduardo Casiliari; Francisco J. González, and Francisco Sandoval, “Modeling of HTTP Traffic,” IEEE COMMUNICATIONS LETTERS, VOL. 5, NO. 6, JUNE 2001.

[9] L. Heberlein, Gihan Dias, Karl Levitt, et al., "A Network Security Monitor," In Proceedings, IEEE Symposium on Research Workshop, 1999.

[10] Steven Snapp, James Brentano, Gihan Dias, et al., "DIDS (Distributed Intrusion Detection System) – Motivation, Architecture and an Early Prototype," In Proceedings, 14th National Computer Secruity Conference, Oct. 1991.

[11] Martin Roesch, "Snort – Lightweight Intrusion Detection for Networks," In Proceedings, 13th USENIX System Administration Conference, Nov. 1999.

[12] Robert Erbacher, Kenneth Walker, and Deborah Frincke, "Intrusion and Misuse Detection in Large-scale Systems. IEEE Computer Graphics and Applications, Jan/Feb 2002.

[13] David Newman, Joel Snyder, and Rodney Thayer, "Crying Wolf: False Alarms hide attacks." www.nwfusion.com, 2002.