Network Web Traffic Generator for Cyber Range Exercises

Chitra Javali
Institute for Infocomm Research (I2R), A*STAR
Singapore
Email: chitra_javali@i2r.a-star.edu.sg

Girish Revadigar
Shield Lab, Huawei International Pte Ltd.
Singapore
Email: girish.revadigar@huawei.com

***Abstract*—Cyber range exercises are important to validate security tools and train personnel to enhance their testing skills. The background web traffic for cyber range exercises must emulate a real network. The currently available traffic generator tools can only replay the captured network traces, or generate simple packet streams and do not consider the representative end-users behaviour. Generating realistic and useful web traffic incorporating a varsity of user behaviour models is a challenging research task.**

In this work, we present a web traffic generator based on Markov model, Dirichlet distribution, and Hybrid distribution. We evaluate our models using a large dataset containing one million web sessions, and our results show that the entropy of output data and cross entropy is approximately same as the entropy of training data. Our tool has interactive GUI and supports various OS environments, and hence, can be used for various web traffic generation applications including CR.

***Index Terms*—Network traffic generator, User behaviour modeling, Cyber Range, Markov Model**

I. Introduction

In recent years, several cyberattacks to the government agencies [1], [2], have raised the concern for developing robust and secure infrastructure systems. Many cyber defence software packages and tools are being developed by the researchers that help to protect the critical infrastructure by preventing and detecting possible cyber threats. Cyber security personnel of government agencies and organizations must also practice skills such as penetration testing, defending networks, hardening the security of critical infrastructure and responding to the adversarial activities.

A cyber range (CR) provides an environment to practice the personnel testing skills and validation of security tools. It is essential to verify such tools and also develop cyber technology skills for the personnel in a realistic environment that comprises of a high-fidelity network. The environment, in addition to the malicious traffic, must contain real legitimate traffic as well, so that the cyber defence tools are robustly tested and do not falsely detect the real traffic as malicious. Thus, background traffic is essential to simulate the complex network traffic. In CR exercises, three different teams are formed based on the expertise of the participants viz., white team, blue team and red team as shown in Figure 1. The white team organises the cyber range exercises, designs the rules, framework and assigns resources to different functions/teams. The white team is also responsible to control the background

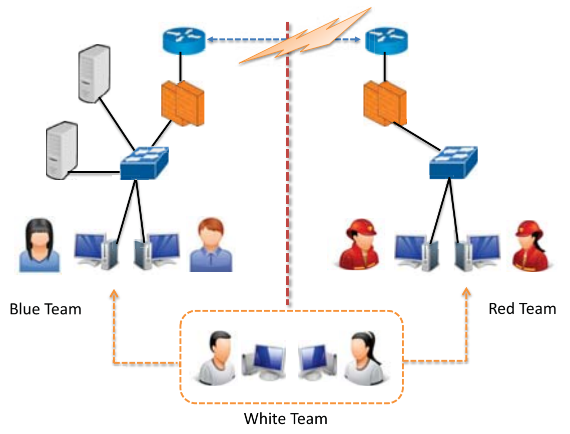


Fig. 1: Architecture of a cyber range environment.

traffic at different stages of the exercises. The blue team defends and maintains the network, servers, applications and services such as mail server, database server, and web server etc., as defined by the framework of the white team, and responds to the attacks by red team. The red team attacks the network of the blue team by following the pre-configured rules designed by white team.

A. Motivation

The background traffic used in CR must be similar to the random traffic generated as a result of every day web activities [3]. Several tools [4]–[6] are available that can capture and replay the traffic. However, the tools are limited to generate simple packet streams that are useful for throughput testing, or are designed only for specific applications. They do not consider the representative end-user's behaviour who is primarily responsible for generating the network traffic. Prior work [7], [8] have considered these problems, however, there is still lack of adaptable user-behaviour models. Generating realistic and useful web traffic incorporating a varsity of user behaviour models is a challenging research task.

Generally a user browsing session consists of a sequence of webpages visited in a specified interval of time [9], [10]. These sessions can be distinguished into two main types - *goal-oriented* and *general* browsing [11]–[14]. In the former, the user searches the web for specific information to satisfy his/her task need and the latter is defined as browsing the web casually for some random information. For instance, consider

Alice who is interested to know the details of the football world cup matches, e.g., the venue, ticket price, teams involved etc., then she clicks on the sports related webpages and delves to several webpages until she finds the required information. Such a session can be described as a sports-based persona and can be categorised as goal-oriented browsing. Overall, a user browsing behaviour depends on his/her different abilities, background, interests and knowledge. Hence, the modeling of user's browsing behaviour is important to generate various persona-based realistic web traffic used in many applications, especially in CR.

Motivated by the above research challenge, in this work, we model our web traffic generation based on the following requirements. The web traffic generator:

- must be able to generate realistic network traffic to be employed in cyber range exercises.
- must be able to emulate different user behaviours.
- must be easy to handle by people with non-technical background, and require minimal human intervention.
- must not replay the captured traffic, as such traffic can be easily distinguished, since it repeats several times.

B. Our Contributions

1. We propose an algorithm to categorise the user-web browsing behaviour into two major classifications - *general* and *goal-oriented*, and further sub-categorise the goal-oriented sessions into different persona-based web browsing.
2. We propose a Markov model-based human web browsing behaviour model. We use a large dataset consisting of 1 million user web browsing sessions for modeling and evaluate the performance by generating different persona-based web traffic. We calculate the entropy, cross-entropy and perplexity of the generated traffic and the transitions of different users between various states/categories.
3. Our results show that the cross-entropy of generated HTTP/HTTPS traffic for various categories is approximately similar to input entropy (i.e., entropy of same category in the dataset) which implies that our model has strong associativity with the input data. The evaluated perplexity for all the categories shows a lower value, implying that the model is a good predictor and generates network traffic as close as the input data used for training the model.
4. We present another 'smart' traffic generator based on dirichlet and hybrid distribution models that generate different persona-based web traffic without the need of training, i.e., generates solely based on the control settings, which can be used when no prior dataset is available for training.
5. We evaluate the smart generator by calculating the entropy of different persona-based traffic generated by setting the parameters (e.g., probability of selected category only) matching to the reference dataset. The results show that, the smart generator produces traffic with nearly matching entropy, similar to the original dataset.
6. Finally, we design a GUI for the web traffic generator having the options to select a specific model. Our tool is built using python packages and is platform independent and can be run on any OS (supporting python) without any modifications.

To the best of our knowledge, we are the first to propose a web traffic generator tool based on markov-model, dirichlet and hybrid distribution-based models, that is useful for cyber range exercises supporting multiple OS environments.

The rest of the paper is organised as follows: Section II presents the related work. In Section III, we explain the methodology. In Section IV and V, we present the markov model and dirichlet, hybrid distribution modeling of user web browsing. The evaluation of the models are explained in Section VI. Section VII presents the GUI design of the traffic generator tool, and Section VIII concludes the paper.

II. Related Work

Several studies exist for internet traffic characterisation and modeling, and there are several tools that generate the network traffic [15], [16]. Flow-level network traffic models have been proposed in [17]–[19], where the researchers have presented mathematical techniques to analyse the response times, fairness, and queue lengths. In [5], the authors have presented another flow-level tool that follows an empirical distribution of TCP and UDP characteristics and uses source-level descriptions to generate the traffic. Another tool described in [6] presents a structural model to capture application level packet interactions. The main aim of this tool is to analyse the distribution of the packet arrival time at different time scales and simulate the traffic for various individual applications.

There also exist many HTTP load testing tools such as SPECweb99 [20], Web Polygraph [21], httperf [22] that are based on synthetic models of Web traffic. The tools run considering only local area network characteristics and fail to scale for wide area network. Another tool SURGE [23] creates a web workload generation tool by observing distributional properties of the Web server. The authors in [24], [25] have studied stress testing and evaluated the performance of World Wide Web Servers.

Few tools generate traffic based on raw packet traces. Tcpreplay [4] is a tool used to replay the captured traffic for both passive sniffer devices and also routers, firewalls and IPS. The tool allows one to modify the IP address, MAC address, and selectively choose the packets in the captured traffic. The main aim of Tcpreplay is for stress testing of various network devices which do not have the capability to connect to services. Hence, for this purpose, Flowreplay [26] was developed that can connect via a TCP or UDP to server that sends or receives data based on a pcap capture file. However, it is only capable of replaying the client side of a pcap against a real service on the target host. Another tool tcplib [27] simulates the traffic based on applications particularly FTP and telnet. The tool models application-level characteristics and user workloads to generate the traffic. Monkey-See Monkey-Do [28] tool replays an emulated workload similar to a site's operating conditions.

Ixias BreakingPoint [29] generates packet-level network traffic on specific protocols and replays the traffic with high-bit rate. EXata [30] is another product for cyber security testing and training. EXata focuses on wireless communications like how the network would respond/behave in a wireless network for scenarios of eavesdropping, jamming etc.

Our work is entirely different from all the above existing tools. Rather than building a traffic emulator, we build a different type of smart traffic generator that satisfies user application scenarios, i.e., models persona-based user browsing behaviours by (i) generating web traffic that resembles a reference data, and (ii) generating traffic solely based on administrator's setting when no prior data is available. Thus, each virtual user in the cyber range environment can generate network traffic based on his/her interest and hence produce realistic traffic.

III. Methodology

In this section, we present our methodology of classification of user web sessions and propose two types of persona-based web traffic generators, namely, (i) Markov model-based generator, and (ii) Smart traffic generator. The Markov model-based design is useful when the web traffic need to be generated according to (or as close as) network traces/data captured in historic web sessions. In this method, the web session logs are initially classified into *goal-oriented* and *general* browsing. The logs consist of the URL/webpages visited by each user in a sequential order. The goal-oriented logs are further clustered into different categories based on the user's interests. Once the user sessions are categorised, a first order Markov model is built for each category. Additionally, a Markov model is also built for the general category. Finally, the browsing traffic is generated based on the transition matrices constructed for each category of goal-oriented and general sessions.

The Smart web traffic generator generates the persona-based traffic without requiring training data/captured web session logs. It can be employed when there is no prior log/data depicting specific human browsing behaviour. This method is useful particularly when the traffic needs to be generated for a newly constructed website e.g., a customised website for CR exercises which is not accessible to outside world, to generate different persona-based browsing traffic purely based on the control settings which facilitates an administrator to set the parameters like - the total number of categories available in the given website, select a particular category/behaviour of browsing, the probability for selected category in the overall traffic, etc. In the following sections, we describe the steps of each model in detail.

IV. Markov Model-Based Web Traffic Generator

A. Classification of Web Sessions

The logs captured in web browsing sessions consist of series of web pages visited by multiple users. We represent the web pages by numbers. For instance, if the URL sequence is *session* = {*3, 3, 4, 2, 3, 5*}, this implies that a user has visited the web-page associated with category 3 thrice, and to categories 4, 2 and 5 only once.

TABLE I: Classification of web sessions as general and goal-oriented with t = 0.4.

Sl.No.	Web session	Classification
1	9, 7, 9, 7, 7, 9, 12, 6, 10, 13, 14, 12, 3	$S_{general}$
2	1, 1, 16, 16, 6, 6, 6, 16 6, 6, 1	$S_{goal-oriented}$ as $ep_i > t$
3	11, 15, 7, 15, 11, 11, 2, 2, 8, 8, 8, 8, 2	$S_{general}$
4	12, 12, 12, 16, 16, 14, 16, 16, 16, 16	$S_{goal-oriented}$ as $ep_i > t$
5	2, 3, 3, 3, 3, 3, 3, 3, 3, 17, 3, 17, 11, 17, 10, 10, 10, 10, 12, 1	$S_{goal-oriented}$ as $ep_i > t$

In order to classify each session we determine the empirical probability ep_i of every visited web-page in a web-session. Given an event s_i in a sample space S, the empirical probability is the absolute frequency normalised by the total number of events N. If the evaluated empirical probability $> t$, a threshold, then we classify the web-session as *goal-oriented* else it is classified as *general* category i.e.,

$$Session = \begin{cases} S_{goal-oriented} & \text{if } ep_i \geq t \\ S_{general} & otherwise \end{cases} \tag{1}$$

This holds true, since a user interested in a particular category of the website visits more pages related to that category compared to other. Hence, threshold based categorization allows to extract particular browsing sessions meeting the criteria. Each of the goal-oriented sessions are further categorized based on the interests of each user i.e., whether the user is browsing web-pages related to a particular topic, by employing the above explained procedure of empirical probability. Table I illustrates the classification of different web-sessions into general and goal-oriented categories.

B. Markov Model-based Design

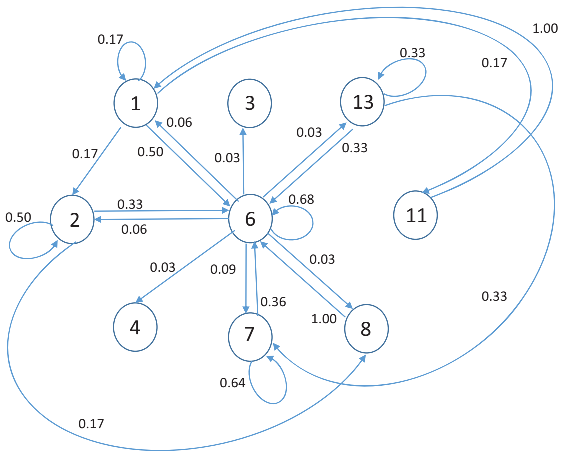
Once the web sessions are classified as goal-oriented and general, and further the goal-oriented sessions are categorised as persona-based, we model each of the sessions by first order Markov model. Consider a sequence of random variables $X_1, X_2, \dots X_n$ belonging to a finite state space S. The first order Markov model property states that the distribution of the next occurring variable X_{n+1} depends only on the current variable X_n. The state space S consists of finite states given as: $s_1, s_2, \dots s_m$. The probability distribution can be represented as follows:

$$\begin{split} P(X_{n+1} = x_{n+1} | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \\ P(X_{n+1} = x_{n+1} | X_n = x_n) \end{split} \tag{2}$$

Here each state represents a URL web page and the chain transits from one webpage to another. The changing of one state s_i to another state s_j is termed as transition and the probability with which the state changes is the transition

1, 1, 6, 6, 2, 2, 6, 6, 6, 1
6, 6, 6, 6, 6, 6, 6, 6, 6, 2, 2, 2
1, 2, 6, 6, 7, 7, 7, 7, 6, 6, 6
13, 13, 6, 13, 7, 7, 6, 7, 6, 6, 4
1, 6, 6, 1, 11, 1, 6, 6, 8, 6
6, 6, 6, 7, 7, 7, 7, 6, 6, 6, 3

(a) Web session logs of a goal-oriented category.



(b) Markov model-based transitions of the web pages.

Fig. 2: Web session logs and the corresponding Markov states.

probability p_{ij} given by:

$$p_{ij} = P(X_1 = s_j | X_0 = s_i) \tag{3}$$

The probability distributions of transition from one web page to another can be represented by a transition matrix M = $(p_{ij})_{i,j}$, where each element of *(i, j)* represent the transition probability $p_{ij} > 0$ i.e., a non-negative number.

Let us consider an example to illustrate this procedure. Figure 2a shows the web sessions of various users from a particular category. Here each sequence corresponds to a session of a particular user. A transition matrix is constructed for the sequences and Figure 2b shows the corresponding first order Markov model representation of the web sessions. The example has 9 states i.e., 9 categories of web pages visited by the user and the weight represents the transition probability from one web page to another.

V. Smart Web Traffic Generator

We present a smart tool to generate goal oriented web traffic that does not require any training. We present two different methods for assigning the probabilities for each of the selected categories, viz., (i) Dirichlet distribution, and (ii) Hybrid distribution in the following sections.

A. Dirichlet Distribution-based Method

Dirichlet is a stochastic process described by discrete probability distribution with base distribution H and a concentration parameter $\alpha > 0$. Dirichlet process can be explained by considering an example of a stick-breaking process [31], [32]. The stick-breaking process is surely a discrete process [33] that can be represented as follows:

$$P(\cdot) = \sum_{k=1}^{\infty} w_k \delta_{Z_k}(\cdot) \tag{4}$$

where w_k are the random weights such that $0 \leq w_k \leq 1$ and $\sum_{k=1}^{\infty} w_k = 1$ and independent of the location Z_k. The locations Z_k are independent and identically distributed according to H. The probabilities of w_k are given by:

$$w_k = w'_k \cdot \prod_{i=1}^{k-1} (1 - w'_i) \tag{5}$$

where w'_i are independent random variables with a beta distribution of (1, α).

If a stick of unit length is split into two pieces, keeping one piece aside and considering the weight of second piece, we break the second piece according to w'_k and assign it to w_k. This process is continued and can be an infinite process or limited to N. The summation of all the distributions of the sticks results to 1.

B. Hybrid Distribution-based Method

Similar to the above illustration, the Smart tool allows to set a particular threshold to selected category, and all other categories are assigned different weights by the Dirichlet distribution. In this stochastic procedure which we call as the Hybird distribution (modified Dirichlet), we set a concentration parameter $\alpha > 0$ to the intended category that is identified by an ID/number, and equal weights w_{eq} for all other locations Z_k. The weights w_α and w_{eq} range between 0 and 1.

$$P(\cdot) = w_\alpha \delta_{Z_1}(\cdot) + \sum_{k=2}^{\infty} w_{eq} \delta_{Z_k}(\cdot) \tag{6}$$

Considering the example of stick-breaking process, a stick of unit length is split into two pieces. Let the weight of first piece of stick be w_α, and the second piece is fragmented into sub-pieces each having equal weight w_{eq}. The summation of all the assigned weights will then be equal to 1.

VI. Evaluation

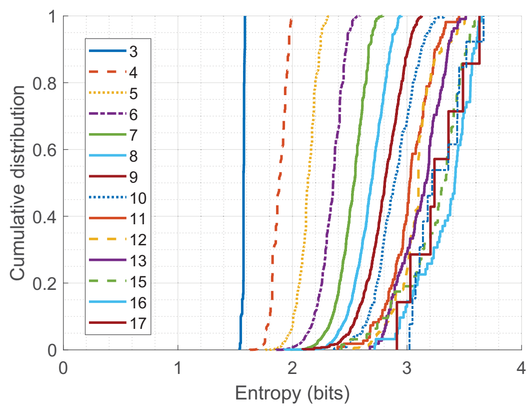
In this section, we present the detailed evaluation of our proposed web traffic generation methods.

A. Evaluation Metrics

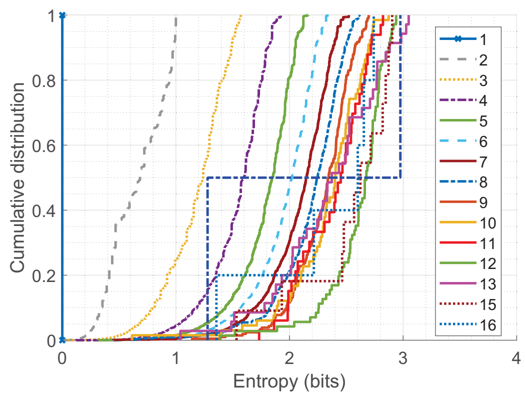
Following are the evaluation metrics we employ for analysing the performance of our models.

1) Entropy: In order to measure the predictability of the user clicking on a URL category, we measure the randomness associated with a web session. Randomness is the lack of certain pattern or predictability in events. As per information theory, entropy is used to quantify the randomness of any event. Shannon entropy H of a discrete random variable X with possible values $x_1, x_2, \ldots, x_n$ is defined as following:

$$H(X) = -\Sigma_x p(x) log_b p(x) \tag{7}$$



(a) The cdf plot of entropy for general category datasets.



(b) The cdf plot of entropy for goal-oriented category datasets.

Fig. 3: Entropy of general and goal-oriented web sessions for different number of unique URL categories present.

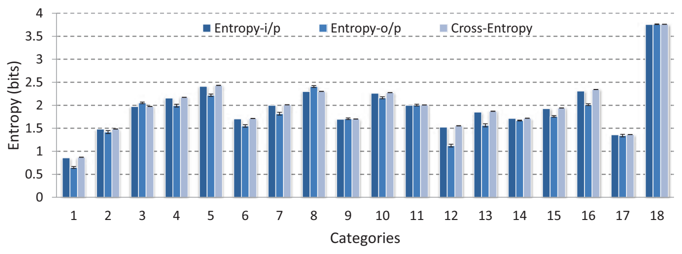


Fig. 4: Entropy of input data, generated output data, and cross entropy for different category datasets.

where b is the base of the logarithm and $p()$ is the probability mass function. We use b = 2 and hence, the unit of entropy is bits. Shannon entropy for a discrete probability distribution gives the amount of information required to identify random samples from the distribution. We evaluate the Shannon entropy for each web session which gives the intuition of predictability of next outcome. A web session with lower entropy implies that the session follows certain probability distribution with more number of items belonging to a particular category, and is predictable.

2) Cross Entropy: Cross entropy is a metric to measure the generative model. Consider two distributions p and q having the same set of states, but having distinct probabilities for the states, then the cross entropy is defined as:

$$H(p,q) = -\Sigma_x p(x) log_b q(x) \quad (8)$$

It is closely related to Shannon entropy and is defined as the amount of information needed to identify random samples from p, when using an optimal coding scheme constructed from q. Here p is the true distribution and q is the model distribution. If a model reproduces the target distribution exactly, then the difference between Shannon entropy and cross entropy can be considered as distance metric of the two distributions over the same set of samples as ≈ 0.

3) Perplexity: Perplexity is an information theory metric to evaluate the prediction of a sample based on the probability model. This value indicates how good the model is to predict the sample. The perplexity of a model q is defined as:

$$2^{\frac{-1}{N}\sum_{i=1}^{N} log_2(p_i)} \quad (9)$$

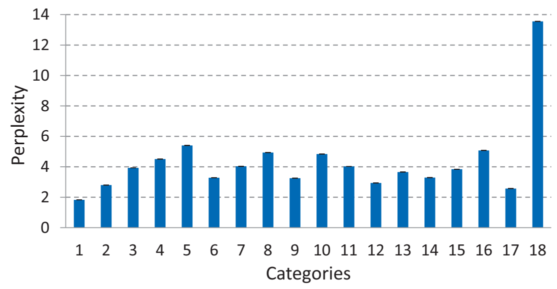


Fig. 5: Perplexity for different category datasets.

TABLE II: Web-page categories of msnbc.com dataset.

URL No.	Category	URL No.	Category
1	frontpage	9	msn-news
2	news	10	health
3	tech	11	living
4	local	12	business
5	opinion	13	msn-sports
6	on-air	14	sports
7	misc	15	summary
8	weather	16	bbs
		17	travel

where the size of the test sample is N and p_i is the probability that model q will predict for the i^{th} sample. The exponential term can be interpreted as the cross-entropy i.e.,

$$H = -\frac{1}{N}\sum_{i=1}^{N} log_2(p_i) \quad (10)$$

Perplexity is the inverse of the average probability assigned by the predicted model to each sample in the test. The perplexity of a not-so intelligent model is equal to the size of the test sample.

B. Analysis of Markov Model-based Design

1) Dataset: For evaluating our Markov model-based design, we have used a publicly available dataset of msnbc.com [34]. The dataset consists of sequences of web-pages visited by a large number of users during a period of 24 hours. Each sequence in the dataset corresponds to the

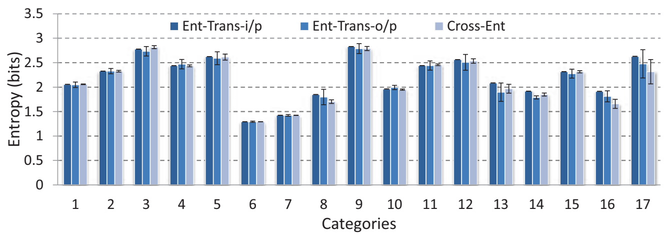


Fig. 7: Entropy of transition of items for *on-air* category.

webpages visited by a user and each item in the sequence is the page requested by the user. The webpages are assigned different IDs, e.g., unique numbers, based on different categories each page belongs to, and the fine-grained URL details are not taken into consideration. The dataset has 17 different categories as shown in Table II, which represents the organisation/structure of the website. The total number of sessions logged in the dataset are approximately 1 million and have variable lengths i.e., from minimum 1 to maximum 250. For our analysis, we consider the web sessions having sequence length $>= 10$, as we would like to analyse whether the user is interested in a specific category/topic. Hence, more the number of event items in a sequence, better will be the constructed persona of the user.

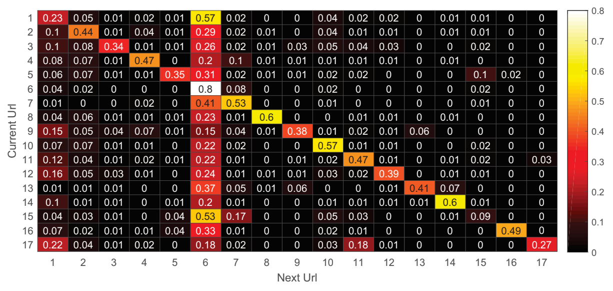
2) *Entropy Analysis of General and Goal-oriented Datasets:* In this subsection, we analyse the entropy of general and goal-oriented user web browsing behaviour. The first step is to classify all the sessions in the dataset into general or goal-oriented based on whether the entropy of the session is less than or greater than a threshold t, respectively (i.e., $> t$ implies goal-oriented). After this, we further analyse the entropy of sessions belonging to different categories. It is worth noting that, the entropy of each session depends on two factors - the total number of items/URLs and the number of unique items/URLs. As each of the web sessions have different number of webpages visited, it would be appropriate to first segregate the web sessions based on number of unique URL symbols present, and then compare the corresponding entropies of web sessions, in each of the goal-oriented and general categories. Recall that, the reference dataset we have considered has a total of 17 categories, and each user session consists of only a subset of these categories i.e., the user browses only those webpages in which he/she is interested. Hence, the number of unique URLs (categories) visited by user in a session can be less than 17. Figure 3a shows that for general category we obtained 3–17 unique symbols, and from Figure 3b we can observe that, we obtained 16 unique URL symbols for the goal-oriented web sessions ranging from 1–16. It is obvious that we do not obtain general category web sessions having only 1 or 2 unique symbols, as it is categorised as goal-oriented web session, as explained in Section IV. First, we will explain the web sessions starting from symbol 3 as it is common for both the categories. For the goal-oriented web sessions the entropy ranges from 0.19 to 1.6 bits, where as for general category the entropy ranges from 1.55 to 1.6 bits. About 95% of the general web-sessions have entropy > 1.56 bits where as it is reverse for goal-oriented, i.e., only 5% of the web sessions have entropy > 1.56 bits. Similarly for other web-sessions, we observe that the average entropy for goal-oriented is always less than the entropy of general web-sessions. This is because, in the goal oriented sessions, the users click on more web-pages with same category which reduces the entropy. For the web-sessions that are categorised as general, higher entropy is observed indicating that users click on a variety of web-pages.

3) *Entropy Analysis of Categorised Input Data, Generated Output Data and Cross Entropy:* Once the goal-oriented web-sessions are extracted, we further classify them into various persona and evaluate the entropy for every category i.e., bbs, business, travel, etc,. We call this as input entropy, since each of the categorised sessions will be input to our Markov model. As observed from Figure 4, the input entropy (first column of each category) ranges from 0.8 to 2.48 bits. The category 18 represents the general category and has an input entropy ≈ 3.75 bits, much greater when compared to goal-oriented entropies. Next, we build persona based model for each category as explained in Section IV to generate realistic web traffic. In order to verify whether our model generates web sessions that are very similar to the input data, we evaluate the entropy of the generated output data. We run our web traffic generation tool 10 times for each category. For each run, we generate an average of 1000 user sessions each having 30 URLs[1]. From Figure 4, we observe that for all the categories, the entropy of the generated output data (second column) is approximately equal to the entropy of input data. We also evaluated the cross-entropy (third column) for each of the categories and observed that it approximates to the corresponding input entropy. This implies that the model is a good predictor and has more associativity with the input data.

4) *Perplexity Analysis:* Figure 5 presents the perplexity of the URL items. For goal-oriented categories, the perplexity ranges from 1.9 (category 1) to 5.2 (category 5). In our dataset we have a total of 17 states and the maximum perplexity we obtained is 5.2, which is less than total number of states. This implies that our model makes stronger predictions. Only the category 18 which represents the general category has a perplexity of 13.8, which shows that the general model has more variance than the goal-oriented model, and in every state it has to consider more number of webpages to predict the next webpage state.

5) *Analysis of Transition Between URL Categories:* In this subsection we explain the analysis of all the evaluation metrics with respect to the transition of each URL category in a web session. The aim of this evaluation is to demonstrate that our scheme generates the web traffic which not only consists similar entropy as per the original training dataset, but also closely follows the transition between each URL categories as per the training dataset. Consider the URL category *on-air* represented by the URL number 6. Once the Markov model-based input

[1]We set 30 as it is an average number of visists in a session as observed in a reference dataset.


(a) Transition matrix for input data.

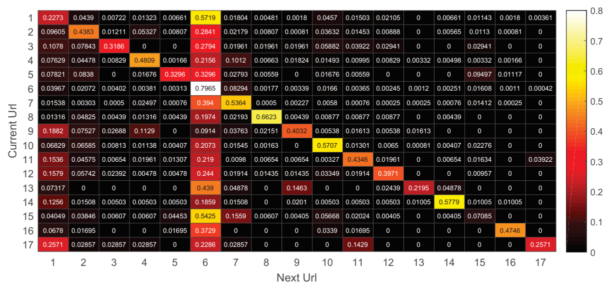

(b) Transition matrix for output data.

Fig. 6: Transition matrices generated for input and output items for one of the Markov model-based user model for goal oriented category showing close association.

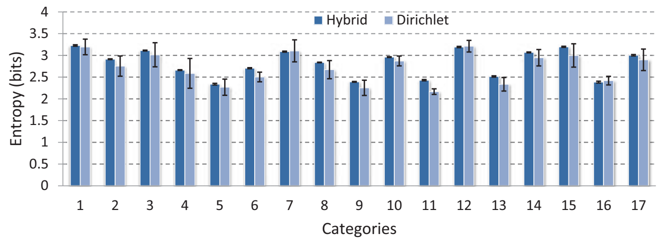


Fig. 8: Entropy of Dirichlet and Hybrid distribution-based traffic generators.

transition matrix is constructed, an output URL browsing data is generated and we obtain the corresponding transition matrix. We have constructed the Markov-model for on-air category having a total of 2820 web sessions. The minimum length of the web-session is 10. We generated 1000 different web sessions depicting number of users, each having a sequence length of 30 (i.e., 30 URL visits in each session). The process is repeated for 10 iterations. Figure 6 shows the input and output transition matrix for on-air category. As observed from the figure, the output transition matrix is very similar to input transition matrix i.e., $Output\_Trans\_Matrix(i, j) \approx Input\_Trans\_Matrix(i, j)$ which shows that our model generates the web traffic close enough to the input data.

Figure 7 shows the input, output and cross entropy for one of the categories *on-air*. From the figure, it can be observed that the transition entropy for input and output for on-air is $\approx$ 1.3 bits the least when compared to other categories. This shows that the original and generated datasets contain majority of visits to on-air category web pages. The cross-entropy of transition of items is approximately equal to the entropy of input transitions, which shows that our model generates a distribution almost or nearly same as input distribution. Similar observation was noticed for all the other categories.

C. Analysis of Smart Web Traffic Generator

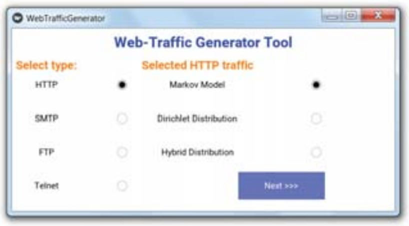
As described in Section V, the smart web traffic generator facilitates generating different persona-based web traffic as per the selected settings. We have evaluated the performance of both Dirichlet and Hybrid distribution based methods for generating various category-based web traffic. Particularly, for each category (goal oriented traffic), we generated 1000 user sessions each having 30 URL visits, similar to Markov model based scheme analysis. To ensure that the generated traffic is realistic, we set the probability of the selected category (goal) to a random number between 40–65%[2]. For each traffic generated, we calculated the entropy of items in the data. The process is repeated for 10 iterations. Figure 8 shows the entropy of items (URL belonging to 17 different categories) in the web traffic generated using Dirichlet and Hybrid distributions for the same parameter settings. It can be observed that the mean for both the cases, for all categories, is nearly the same, whereas, the standard deviation is more in case of Dirichlet distribution. This is because, the Dirichlet distribution assigns slightly different probabilities for each categories in different iterations even after the administrator sets particular parameters, whereas, the Hybrid distribution retains the settings. However, both the types of traffic generation are useful in different scenarios, and our smart tool has the capability to generate the desired traffic.

VII. GUI Design for Web traffic generator tool

In this section, we present the design and implementation details of our interactive tool for web traffic generation. The white team that controls the background traffic must be able to visualise and analyse the traffic generated in the CR network. We have built a web traffic generator tool in Python [35] that incorporates the designs/models presented in previous sections. Our tool is platform independent and does not require any customisation or change in the source code. Figure 9 shows some of the screen-shots of the GUI. Initially, the operator has options to select among different types of traffic like HTTP, SMTP, etc[3]. Once the type of traffic is selected, our tool presents options to choose between various models. For instance, as shown in Figure 9a, HTTP traffic is selected and the available models are Markov, Dirichlet and Hybrid. Once the model is chosen, a new menu depending on the selected option is displayed. Figure 9b shows whether a controller can select goal oriented or general category. If the goal-oriented is selected, then different categories of persona are shown by the GUI. One can then mention the number of URL to be generated and click on the 'Generate' button. The sliding bar shows the progress of the web traffic generation process.

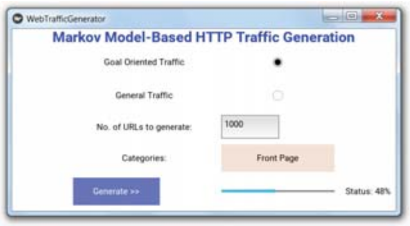
[2]This probability is similar to the feature/range observed in a real web traffic dataset used for training Markov model.

[3]Note that in this work, we focus mainly on the HTTP/HTTPS traffic generation only.



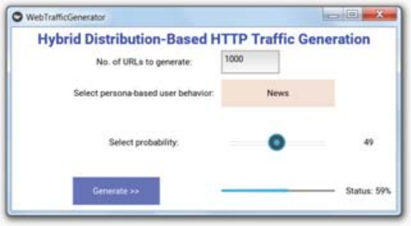
HTTP ●
Markov Model ●

(a) GUI showing the selection of different models: markov, dirichlet and hybrid



Goal Oriented Traffic ●
No. of URLs to generate: 1000
Categories: Front Page
Status: 48%

(b) GUI selecting goal-oriented traffic for markov model.



No. of URLs to generate: 1000
Select persona-based user behavior: News
Select probability: 49
Status: 59%

(c) GUI selecting hybrid distribution-based HTTP traffic for *news* category.

Fig. 9: GUI for HTTP traffic generator tool.

Figure 9c shows the distribution selection having a numerical probability value as entered by the controller. Thus, our GUI-based tool is easy to handle even for the people with non-technical background, and no special training is required.

VIII. Conclusion

Generating realistic web traffic that simulates the real world network traffic is very essential for cyber range environment. In this paper, we have presented the design, implementation and evaluation web traffic generators based on Markov model, Dirichlet and Hybrid distribution. The results show that the generated data has same entropy and cross-entropy as reference input data and the perplexity ranges from 1.9 to 5.2, which implies that the model is a good predictor and has more associativity with the input data. Our interactive GUI tool facilitates easy deployment on any OS platform to generate realistic web traffic data. In our future work, we would like to incorporate the probability of events i.e., visiting webpages in an interval of time and the behaviour of user when he/she is browsing in multiple windows.

References

[1] (2018) Singhealth cyber attack. [Online]. Available: https://graphics.straitstimes.com/STI/STIMEDIA/Interactives/2018/07/sg-cyber-breach/index.html

[2] (2018) Uber data breach. [Online]. Available: https://www.cnbc.com/2018/09/26/uber-to-pay-148-million-for-2016-data-breach-and-cover-up.html

[3] J. Davis and S. Magrath, "A Survey of Cyber Range and Testbeds," *Defence Science and Technology Organisation (DSTO)*, Oct. 2013.

[4] (2013) Tcpreplay. [Online]. Available: http://tcpreplay.synfin.net/

[5] J. Sommers, H. Kim, and P. Barford, "Harpoon: A Flow-level Traffic Generator for Router and Network Tests," in *Proc. of Measurement and Modeling of Computer Systems*, 2004.

[6] K. V. Vishwanath and A. Vahdat, "Swing: Realistic and Responsive Network Traffic Generation," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 712–725, June 2009.

[7] L. M. Rossey, R. K. Cunningham, D. J. Fried, J. C. Rabek, R. P. Lippmann, J. W. Haines, and M. A. Zissman, "LARIAT: Lincoln Adaptable Real-time Information Assurance Testbed," in *Proc. of IEEE Aerospace Conference*, 2002.

[8] J. Blythe and L. J. Camp, "Implementing Mental Models," in *IEEE Symposium on Security and Privacy Workshops*, 2012.

[9] A. R. Benson, R. Kumar, and A. Tomkins, "Modeling user consumption sequences," in *Proc. of WWW*, 2016.

[10] C. Lo, D. Frankowski, and J. Leskovec, "Understanding Behaviors That Lead to Purchasing: A Case Study of Pinterest," in *Proc. of ACM KDD*, 2016.

[11] Y. Zhang, W. Chen, D. Wang, and Q. Yang, "User-click Modeling for Understanding and Predicting Search-behavior," in *Proc. of ACM SIGKDD*, 2011.

[12] M. Richardson, E. Dominowska, and R. Ragno, "Predicting Clicks: Estimating the Click-through Rate for New Ads," in *Proc. of WWW*, 2007.

[13] G. E. Dupret and B. Piwowarski, "A User Browsing Model to Predict Search Engine Click Data from Past Observations," in *Proc. of ACM SIGIR*, 2008.

[14] J. Cheng, C. Lo, and J. Leskovec, "Predicting Intent Using Activity Logs: How Goal Specificity and Temporal Range Affect User Behavior," in *Proc. of WWW Companion*, 2017.

[15] (2008) Network traffic generator tools. [Online]. Available: http://www.icir.org/floyd/papers/webpages/trafficgenerators.html

[16] Song Luo and G. A. Marin, "Modeling Networking Protocols to Test Intrusion Detection Systems," in *Proc. of IEEE LCN*, 2004.

[17] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "Modeling Internet Backbone Traffic at the Flow Level," *IEEE Trans. on Signal Processing*, vol. 51, no. 8, pp. 2111–2124, Aug 2003.

[18] S. B. Fredj, T. Bonald, A. Proutière, G. Régnié, and J. W. Roberts, "Statistical bandwidth sharing: A study of congestion at flow level," in *Proc. of SIGCOMM*, 2001.

[19] A. Kumar, J. Kim, S. C. Suh, and G. Choi, "Incorporating Multiple Cluster Models for Network Traffic Classification," in *Proc. of IEEE LCN*, 2015.

[20] (2008) Specweb99 benchmark. [Online]. Available: https://www.spec.org/web99

[21] (2004) Web polygraph. [Online]. Available: http://www.web-polygraph.org

[22] D. Mosberger and T. Jin, "httperf: a Tool for Measuring Web Server Performance," *SIGMETRICS Perform. Eval. Rev.*, vol. 26, no. 3, pp. 31–37.

[23] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation," in *Proc. of ACM SIGMETRICS*, 1998.

[24] G. Banga and P. Druschel, "Measuring the Capacity of a Web Server," in *Proc. of the USENIX Symposium on Internet Tech. and Systems*, 1997.

[25] E. M. Nahum, M.-C. Rosu, S. Seshan, and J. Almeida, "The Effects of Wide-area Conditions on WWW Server Performance," in *Proc. ACM SIGMETRICS*, 2001.

[26] (2012) Flowreplay. [Online]. Available: http://tcpreplay.synfin.net/wiki/flowreplay

[27] P. B. Danzig and S. Jamin, "tcplib: A library of internetwork traffic characteristics," Tech. Rep., 1991.

[28] Y.-C. Cheng, U. Hölzle, N. Cardwell, S. Savage, and G. M. Voelker, "Monkey See, Monkey Do: A Tool for TCP Tracing and Replaying," in *Proc. of the USENIX Annual Technical Conference*, 2004.

[29] "Network Testing with Simulated Traffic. Does Realism Matter? An Ixia BreakingPoint Case Study," *White Paper*, Aug. 2014.

[30] (2008) Scalable network technologies. [Online]. Available: https://web.scalable-networks.com

[31] H. Ishwaran and L. F. James, "Approximate Dirichlet Process Computing in Finite Normal Mixtures: Smoothing and Prior Information," *Journal of Computational and Graphical Statistics*, vol. 11, no. 3, pp. 508–532, 2002.

[32] ——, "Gibbs Sampling Methods for Stick-Breaking Priors," *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 161–173, 2001.

[33] D. J. Navarro, T. L. Griffiths, M. Steyvers, and M. D. Lee, "Modeling Individual Differences Using Dirichlet Processes," *Journal of Mathematical Psychology*, vol. 50, no. 2, pp. 101–122, 2006.

[34] "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[35] (2018) Kivy: Cross-platform python framework for nui development. [Online]. Available: https://kivy.org