Two-Parametric Model for Detecting Slow HTTP DDoS Attacks based on Predicting User Behavior

Vitalii Savchenko[1], Oleksander Matsko[2], Ivan Havryliuk[2], Kseniia Yerhidzei[2], and Iryna Novikova[2]

[1] *State University of Telecommunications, 7 Solomianska str., 03110, Kyiv, Ukraine*
[2] *The National Defense University of Ukraine named after Ivan Cherniakhovskyi, 28 Povitroflotsky ave., 03049, Kyiv, Ukraine*

Abstract

The paper explores the problem of slow DDoS attacks detecting based on predicting of user behavior. Author's approach exploits the offered two-parametric forecast model with Number of Connections and the Average Real Network Delay as the prognostic values. Forecasting algorithm uses the method of calculating the a posteriori trajectory of the time series depending on a priori statistical observations. Experimental results show that the method is suitable for early detection of attacks such as Slow HTTP Headers, Slow HTTP Body, Slow HTTP Read. Modelling of traffic parameters confirms the opportunities of the method for predicting slow attacks at different time intervals, as the accuracy of the forecast depends on the timeliness of observations. With enough statistics, the deviation of the forecast curve may be less than 5%.

Keywords

Slow HTTP DDoS attack, user behavior, individual trajectory, forecast model.

1. Introduction

Typical problems faced by any network architecture, including cloud computing, are DDoS attacks. In DDoS attacks, an attacker tries to influence the victim's server or running applications from compromised computers from various locations. An attacker installs malware on many computers over the Internet using a controlled host. Hacked machines act like an army of bots for DDoS attacks. When an attacker wants to attack a victim's server or application, he sends a command to start the attack to the bot hosts so that the bot hosts start attacking the victim. Unlike traditional attacks (UDP and SMTP floods), which limit network capacity or attack network protocols, slow DDoS attacks target on applications. It is well known that slow DDoS attacks can run unnoticed for a long time due to significant similarities with the behavior of legal users with a slow connection. Therefore, because of the complex process of their detection, such attacks require special consideration [1].

1.1. Problem Statement

A slow HTTP DDoS attack at the application level includes many incomplete HTTP requests to the server. Compared to other attacks, slow HTTP DDoS attacks are very difficult to detect because [2]:

- A loyal user may be a source of the attack.
- The attack does not require a wide bandwidth.
- The focus of the attack is on depleting program resources, by simulating a slow-connect network.

There are currently three types of slow HTTP DDoS attacks:

CPITS-II-2021: Cybersecurity Providing in Information and Telecommunication Systems, October 26, 2021, Kyiv, Ukraine
EMAIL: savitan@ukr.net (V. Savchenko); macko2006@ukr.net (O. Matsko); ivan.havryliuk@gmail.com (I. Havryliuk); ergidzey@ukr.net (K. Yerhidzei); irina_nov@ukr.net (I. Novikova)
ORCID: 0000-0002-3014-131X (V. Savchenko); 0000-0003-3415-3358 (O. Matsko); 0000-0002-3514-0738 (I. Havryliuk); 0000-0003-4634-133X (K. Yerhidzei); 0000-0003-4854-0682 (I. Novikova)

- Slow HTTP Headers, or Slowloris attacks. During an attack, the web browser sends HTTP GET requests along with the HTTP header. The HTTP protocol is designed so that the web server waits until it receives the full HTTP header to process GET requests. Each time a web server receives an incomplete header, it assumes that requests are being sent from a slow network and continues to wait for the full header. The slow HTTP header attack uses this behavior and sends requests with incomplete HTTP headers. Because an attacker never sends complete header information, the server continues to reserve the allocated resource. This leads to resource depletion and inability to service any other legitimate requests.
- Slow HTTP Body, or aRe-yoU-Dead-Yet (RUDY) attack. This attack contains HTTP POST requests with the full header, but the length field contains such a large value that the server reserves all allocated resources to receive the entire message. During this attack, an attacker sends messages with the size of the content in small pieces, such as one or more bytes, at very slow intervals. An attacker also opens several such connections to a web server. Thus, the attacker consumes resources and makes them inaccessible to real users.
- Slow HTTP Read. This is a type of slow HTTP attack where attackers send HTTP GET requests with the relevant header and the body to a web server. The core of the attack is to read the answer very slowly. Any HTTP requests begin with the creation of a TCP session between the browser and the server. The size of the TCP window controls the flow of data between the browser and the web server. This value shows the number of bytes that the browser can receive during a TCP session. An attacker often tells a web server that it can only get a few bytes, or that it is not ready to accept any bytes. An attacker opens many such sessions on the server to consume the entire resource so that the server cannot process any genuine requests from a legitimate user.

As you can see, the detection of a slow HTTP attack is a significant problem, because the behavior of an attacker can mimic the behavior of a legitimate user with slow resources. Filtering such behavior is significant, although the consequences can be like conventional DDoS attacks - denial of service to a significant number of legitimate network users.

2. Related Works

A significant number of works are devoted to the problem of counteracting slow DDoS attacks.

In [3], authors offered an architecture of the SDN controller for recognizing slow HTTP attacks. Their approach is based on three stages of the decision making: 1) detection; 2) identification; 3) mitigation. At the identification stage, the system calculates the packet transmission rate and the uniformity of the distance between packets. In case of exceeding the specified indicators of some established average values, the conclusion on existence of attack is made.

The article [4] proposes a method for detecting slow DDoS attacks based on the analysis of TCP flow parameters and their classification by decision trees. The model includes elements of artificial intelligence, where a combination of two data sets is used for training, one of which is generated from a simulated network, and the other from a public data set of DoS attacks. This approach has some advantages, although it requires a significant amount of statistics to train the neural network.

In [5], machine learning methods for recognizing slow DDoS attacks are also considered: a multilayer perceptron (MLP), a reverse propagation neural network, K-nearest neighbors (K-NN), a vector support machine (SVM), and a polynomial naive Bayesian (MN) algorithm. As in the previous case, the application of this approach requires numerous patterns for recognition.

Work [6] considers the general mechanism of protection against slow DDoS-HTTP attacks. Here, for the first time, the parameter of the number of connections and the duration of the connection is entered. Such parameters are one of the most available for calculation and subsequent forecasting.

Publication [7] shows an approach for selecting data based on which effective mechanisms for classifying slow HTTP DoS attacks can be created. At the same time, the authors note that in order to achieve high recognition efficiency, it is necessary to use almost 2 million attack patterns, which are quite difficult to implement in real life.

The most appropriate is the method proposed in [8], which is based on the parameters of TCP connections. The authors propose a model for determining the probability and time of transition of the web server to overload mode. They used observation statistics and forecasting results for this purpose.

At the same time, forecasting in such a model is based on linear trends, which allows you to implement it only for short periods of time.

The publication [9] presents the capabilities of the OpenStack cloud platform for analysis and evaluation of DDoS attacks. At the same time, fixing of network parameters is carried out by means of Wireshark. This approach is one of the simplest and most accessible for simulating slow DDoS attacks of various types. However, there are some difficulties in implementing measures to counter slow DDoS attacks in the cloud.

Traffic patterns are studied in [10], where an algorithm for detecting slow DDoS attacks depending on the server load status is proposed. This approach is effective enough for post-factual analysis, but does not allow to decide on the presence or absence of an attack.

In [11], an OpenStack cloud test model was proposed to evaluate DDoS attacks in a cloud environment. This work explored various attack opportunities for applications running in a cloud environment. The results proposed in [11] were continued in [12] to detect, mitigate and prevent slow DDoS HTTP attacks in the cloud.

The most promising is the approach proposed by the same authors in [13], where they explore a new method and model of classification to protect against slow HTTP-attacks in the cloud. Their solution detects slow HTTP header attacks (Slowloris), slow HTTP body attacks (RUDY) or slow HTTP read attacks. The paper proposes a four-zone attack detection architecture based on the analysis of two parameters: the number of connections to the server and the average delay time of the client response. This approach does not guarantee effective detection of attacks in the early stages of their development, as it is based on a posteriori observations of network traffic.

Thus, most work on counteracting slow DDoS attacks is based on statistical models, does not address the prediction of host behavior, and therefore is not effective enough to detect attacks in the early stages.

The issues of traffic forecasting for DDoS-attack detection have already been considered in [14–15], where the authors proposed a forecast model based on the decomposition of a random process. They used the following as an argument: total network traffic and user response delays, which determine user's individual behavior. In that works, a single-parameter approach was implemented based on only one factor that determines the essence of a slow DDoS attack.

In this publication, the authors aim to consider a two-parametric model for predicting the slow DDoS attack in their combined usage.

3. Slow HTTP DDoS Classifier Architecture

In [13] a solution based on a four-stage zonal architecture of customer classification is proposed. A request from any client to a web server is classified by any of the zone states at any given time using a multi-step zonal model. The model of multi-stage zonal classification architecture is shown in Figure 1. Initially, all incoming requests are monitored for their activity in the monitoring unit. In case of lawful behavior of customers, the request is sent to the green zone. From the green zone, all client requests are forwarded to the web server.

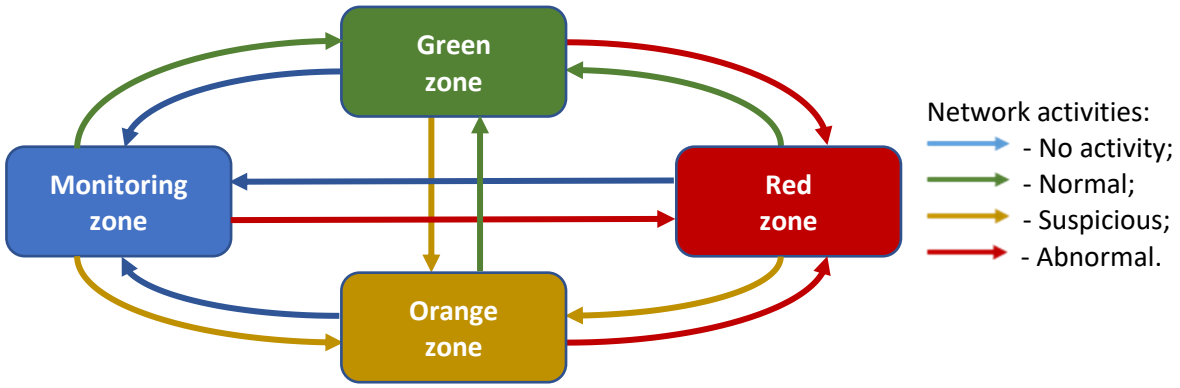


Figure 1: Multi-stage HTTP DDoS classifier architecture

In case the client's activity is suspicious (opening more connections, sending frequent GET requests with an incomplete header, POST requests with fewer bytes with a longer interval before the transition time), such requests are moved to the orange zone of the block and continue to be monitored.

Customers who look abnormal in the monitoring block, green or orange areas (for example, when opening a very large number of connection requests (8 times larger than some average value), connection requests with an incomplete title, publication requests with a few bytes or a request to the Server with a very long interval, for example over 80% of the maximum save interval, compared to the time of its processing), are considered as DDoS-attacks, and these clients fall into the red zone.

Any requests from customers in the red zone are blocked. Customers from the green, orange or red zones without activity or without an active connection request are moved back to the monitoring unit. A customer from the green zone, who constantly shows suspicious activity, moves to the orange zone, and vice versa - if the client shows normal activity after entering the orange zone, he moves to the green block.

The 4-block zonal model contains a limit on the number of connections allowed per client and the number of ping requests to calculate the average network delay.

To recognize a slow HTTP DDoS attack on a web server, the Zonal Model uses:

- Number of Connections (NC) allowed for each client.
- Average Real Network Latency (ARNL) of client availability.

Customers who try to open 6 times more permitted connections are subject to careful monitoring. Here, the model sends the client 5 ping requests. The ping response is used to calculate the average network latency.

The ARNL value is compared to the average time of the client request interval. Intruders who simulate a slow network and a client with a real slow network are identified using this technique. Clients with slow requests exceeding 0.8 of the maximum allowable ARNL are considered an attack and move to the red block. This value may be different, depending on the network settings at the time of application processing.

Customers who try to open 6–8 times more allowed connections and have an average request delay of 0.6 to 0.8 times the maximum allowable ARNL switch to the orange block. Clients in this state remain under supervision and are classified accordingly based on a continuous analysis of behavior.

Customers with less than 6 times of the average connections and a connection delay of up to 0.6 relative to the maximum allowable ARNL are in the green zone.

The system stores newly connected client in the monitoring unit to identify the activity pattern and goes into any of the states based on its behavior defined above. When a client tries to open 6 times more connections than the average number, or the response time to such a client is greater than 0.6 of the maximum allowable waiting time, so such client activity is detected as suspicious. Dangerous activity is classified when the number of open connections exceeds the average by 8 times, and the response time is greater than 0.8 of the maximum value.

Thus, to detect the onset of Slow HTTP attacks successfully, it is necessary to monitor 2 main parameters: 1) Number of Connections (NC) allowed for each client and 2) Average Real Network Latency (ARNL) of client availability. Based on constant monitoring of these parameters, it is possible to detect most Slow HTTP attacks, including Slowloris, RUDY and HTTP Read.

Counteracting such attacks should include two primary measures: 1) diagnose the attack at the earliest stages; 2) separate harmful user behavior from normal behavior. By understanding which user requests result from a DDoS attack, you can configure the settings for firewalls, routers, or other security measures.

The problem of early detection of low or slow DDoS attacks remains relevant. The sooner we find that the traffic parameters are incompatible with their normal values, the sooner it will be possible to take measures to neutralize the attack. Here, it is necessary to add parameter prediction modules to existing detection systems.

We should note that the parameters of NC and ARNL are independent and, during the attack, can occur both separately and together. Also, attacks such as Slow HTTP Headers are quite difficult to detect through ARNL, because in this case the server's response time will be maximum and the impact on the server by the attacker is mainly through a significant number of connections. To organize a successful response to such attacks, it is necessary to organize a system that would allow a parallel

analysis of both factors. The most interesting are the cases of combined use of both factors by an attacker using different strategies to combine them.

4. Development of a Method for Detecting Combined Slow DDoS Attacks based on Parameter Prediction

4.1. Traffic Settings to Detect a Slow DDoS Attack

Most of the work on countering slow DDoS attacks is based on statistical models and does not address the prediction of host behavior, and therefore is not effective enough to detect attacks in the early stages.

This work forms a system for detecting slow DDoS attacks based on predicting traffic elements in the network. As already mentioned, of particular interest are cases where both factors (NC and ARNL) are used together. This option is a combined attack, which is more complex than in the case of a single application of individual factors. To successfully solve the identified problem, it is necessary to build a model and technology for predicting the behavior of traffic parameters, considering the history of host interaction in the network, as well as to offer technology for recognizing combined slow DDoS attacks.

The architecture proposed in [14] is the most appropriate for detecting slow DDoS attacks. Such an IDS should comprise four modules: 1) Traffic Collection Module; 2) Calculating Module; 3) Forecasting Module; 4) Attacks Detection Module.

The system works:

1. During some period, the Traffic Collection Module records the main parameters required for further calculations: IP-addresses of the sender and recipient; TCP window size; number of open connections; package arrival time.

2. For each IP address the Calculating Module counts: the number of open connections and the average delay between transmitted packets

$$\bar{T} = \frac{1}{k-1}\sum_{i=1}^{k}(t_{i+1} - t_i), \quad (1)$$

where:

t_i – i-th package arrival time;

t_{i+1} – $i+1$-th package arrival time;

k – number of packets received during the analyzed period.

The beginning and end of the session are recorded by a built-in timer, after which the duration of open connections is calculated.

3. The Module of Attacks Detection makes a decision about possible slow HTTP attack based on a comparison of the received indicators with average statistical values. In particular, if customers try to open 6–8 times more than the average number of connections and / or the average request time delay is 0.6 to 0.8 times the maximum allowable ARNL, then such a client belongs to the orange zone. Clients in this state remain under supervision and are classified accordingly with results of continuous analysis of behavior. If the number of connections exceeds the number of allowed over 8 times and/or the average request time delay is from 0.8 to 1.0 of the maximum allowable ARNL, the client belongs to the red zone.

However, as shown in [14], the decision about a slow DDoS attack is more appropriate to make based on the forecast of traffic parameters, which will allow to expect such attacks for some time and take relevant measures. Such an approach is implemented in the IDS Forecasting Module.

4.2. Method of Predicting the Parameters of a Slow DDoS Attack

The interaction of computer systems in the network forms an individual trajectory of user behavior for each pair of interactions. Such trajectories have their own features both in normal mode and during a slow DDoS attack. In order to take timely action to neutralize the slow DDoS attack, it is necessary to provide a time trajectory of user behavior, which depends on the actions of the interacting system.

We have already studied prediction of a separate trajectory of time series in [15], where the parameters of motion were checked at long intervals (week or month). The same approach was used to predict slow DDoS attacks in [14]. In both cases, we examined only some single indicators: in [15] – the amount of information per unit time, in [14] – the average delay between transmitted packets. In both cases, a single-parameter forecast model was run and we investigated only one traffic parameter.

Slow DDoS attacks are characterized by small deviations in traffic and, therefore, to detect them, we need to use some extra parameters. Given that a two-parametric model may be more effective and we can cite NC and ARNL as predictive factors. Here, we can perform the NC evaluation relative to some average value of the number of connections, and we can measure the ARNL relative to the maximum connection time allowed by the system for the legitimate user.

Besides the direct values (number of connections and average delay time), we also should calculate the values of the correlation function for each of the measurements using the method of canonical decomposition of a random process, which makes the method more effective for predicting weak disturbances.

To monitor the attack parameters, as in [14], we will use NC and ARNL, which can form a vector of parameters $X = (X_1, X_2, \dots, X_H)$ where $X_i = \{X_{NC_i}, X_{ARNL_i}\}$. Condition fulfillment $X \in S_0$, where S_0 is the tolerance area of the vector X. Random process $X(t)$ reflects the change of parameters over time. Process $X(t)$ is statistically defined in the range $t \geq t_1$, where t_1 is the beginning of observations and $t_k \geq t_1$.

We pose the forecasting problem as follows: for the parameter $x_\omega(t) \in S_0$, which is observed in the interval $t_1 \leq t \leq t_k$, we need to determine the release time of a specific implementation $x_\omega(t)$ beyond the limits S_0 based on the calculation of a posteriori process $X(t)$.

The probability that a particular trajectory of a parameter ω guaranteed to fall within the acceptable range $s \leq t_k$, if by then t_k including its condition was described as $x_\omega(t), t_1 \leq t \leq t_k$, will be

$$P^{ps}(s) = P\{X(s) \in S_0 / x_\omega(t)\}, t_1 \leq t \leq t_k, s \leq t_k. \quad (2)$$

To solve the forecasting problem, the process under study must be represented by the formula

$$X(t) = m(t) + \sum_v V_v \varphi_v(t), \quad (3)$$

where $m(t)$ – mean function of the process;

$\varphi_v(t)$ – non-random (coordinate) time functions;

V_v – random, uncorrelated coefficients $M[V_v] = 0$, $M[V_v, V_\mu] = 0$, $v \neq \mu$.

This representation, proposed in [14, 15], allows it to be applied to any traffic parameter that can be defined as a time series. Process $X(t)$ can be addressed as a random sequence $X(t_i) = X(i), i = \overline{1, I}$ in a discrete series of observations t_i:

$$X(i) = m(i) + \sum_{v=1}^{i} V_v \varphi_v(i), i = \overline{1, I}, \quad (4)$$

where V_v – random coefficient with parameters $M[V_v] = 0$, $M[V_v, V_\mu] = 0$, $v \neq \mu$; $M[V_v^2] = D_v$;

$\varphi_v(i)$ – non-random coordinate function, $\varphi_v(v) = 1$, $\varphi_v(i) = 0$ while $v > i$.

The formulas for variance and correlation function can be written as

$$D(i) = \sum_{v=1}^{i} D_v \varphi_v^2(i), i = \overline{1, I}, \quad (5)$$

$$D(i, j) = \sum_{v=1}^{\inf(i,j)} D_v \varphi_v(i) \varphi_v(j), i, j = \overline{1, I}. \quad (6)$$

Thus, the representation of random parameters (2) allows to solve the problem of detecting a slow DDoS attack based on predicting the behavior of the parameters themselves. If more than one independent parameter is used, the operation is performed according to the parameter vector.

4.3. Algorithm for Detecting a Slow DDoS Attack based on Predicting Two Parameters

To detect slow DDoS attacks under approach (1) – (6), we propose the following algorithm for combined predicting number of connections and delays between transmitted packets.

0. **Start**

1. $X(t) \leftarrow X(t), t = \overline{1,T}$ – formation of an array of process $X(t)$ observations, $X_i = \{X_{NC_i}, X_{ARNL_i}\}$.

2. $x(\mu) \leftarrow x(\mu), \mu = \overline{1,k}$ – formation of an array of control results.

3. $L \leftarrow Length[X(t)], X_i = \{X_{NC_i}, X_{ARNL_i}\}$ – determining the number of trajectories observed.

4. $m(t) \leftarrow Mean[X(t)], X_i = \{X_{NC_i}, X_{ARNL_i}\}$ – calculating the mean of a random function $X(t)$.

5. $c \leftarrow Covariance[X(t)], X_i = \{X_{NC_i}, X_{ARNL_i}\}$ – calculating the covariance matrix for $X(t)$.

6. $d \leftarrow Variance[X(t)], X_i = \{X_{NC_i}, X_{ARNL_i}\}$ – calculating an array of variances of a process $X(t)$.

7. $\varphi \leftarrow Table[0, \{T\}, \{T\}]$ – calculating the initial value of the coordinate functions.

8. $\hat{X}(t) = X(t) - m(t), t = \overline{1,T}$ – centering the source data.

9. $V(t) = X_l(t) - m(t), t = \overline{1,T}, l = \overline{1,L}$ – calculating the initial values of random coefficients.

10. $\varphi_1 = \frac{c_{1,j}}{d_1}, j = \overline{1,T}$ – determining the first coordinate function.

11. **For** $i = 1$ to $i = T$

12. $d_i = c_{i,i} - \sum_{j=1}^{i-1} \varphi_{i,j}^2 \, d_j$ – variance override.

13. **For** $j = 1$ to $j = T$

14. $\varphi_i = \frac{1}{d_1}\left(c_{i,j} - \sum_{l=1}^{i-1} d_l \varphi_{i,l} \varphi_{j,l}\right)$ – redefining coordinate functions.

15. **end for** j

16. **end for** i

17. **For** $i = 2$ to $i \leq T$

18. **For** $k = 1$ to $k < i$

19. $\varphi_{i,k} = 0$ – redefining the coordinate functions of a random process.

20. **end for** k

21. **end for** i

22. **For** $i = 2$ to $i \leq T$

23. **For** $l = 1$ to $l = L$

24. $V_{l,i} = \hat{X}_{l,i} - \sum_{k=1}^{i-1} V_{l,k} \varphi_{k,i}$ – calculating the random coefficients.

25. **end for** l

26. **end for** i

27. $p_s \leftarrow Length[x(\mu)]$ – determining size of the array of control results.

28. $M_1 = Table[m_i + (x_1 - m_1)\varphi_{1,i}, \{i = \overline{1,T}\}]$ – calculating the initial predicted trajectory.

29. **For** $h = 2$ to $h = p_s$

30. $M_h = Table[M_{h-1,i} + \left(x_h - M_{h-1,h}\right)\varphi_{h,i}, \{i = \overline{1,T}\}]$ – calculating of forecast control points.

31. **end for** h

32. $X_{forecast} = Table[M_{k,i} + \sum_{j=k+1}^{i} V_{k,j} \varphi_{k,j}, \{k = \overline{1,p_s}, i = \overline{1,T}\}]$ – calculating the predicted trajectory.

33. **Stop**

The algorithm involves two-dimensional prediction of user behavior parameters. Given the independence of the NC and ARNL parameters from each other (because the attack can develop as by one parameter, and combined), this makes it possible to build a two-parametric classifier according to the above criteria. The developed algorithm of the method allows to determine accurately the random process at the control points and to provide a minimum of the mean square of the approximation error in the intervals between these points.

In a result of application of algorithm, the forecast of development of one or both parameters at the same time can be constructed and the moment when the separate parameter goes beyond critical values can be defined. If the parameter shows a transition to an orange or red zone, which indicates the possibility of a slow DDoS attack, security measures must be taken. The decision about a beginning of a slow DDoS attack should be made for each sender's IP address based on a comparison of the predicted NC and/or ARNL with the critical values to determine when the parameter enters the critical zone. The presence in the model of the average number of requests and the maximum response time to the request makes it possible to consider the individual behavior of interacting hosts in comparison with the behavior of other hosts in similar situations with slow DDoS attacks.

5. Modeling of a Two-Parametric Algorithm for Detecting Slow DDoS Attacks based on Prediction

We performed a simulation of DDoS attack detection using the OpenStack cloud environment architecture with parameter fixation using Wireshark. Slow HTTP Headers attacks were analized by the NC parameter, as well as Slow HTTP Body and Slow HTTP Read attacks by NC and ARNL simultaneously. As in [13] the parameters used for these attacks were taken: the total number of connections (NCtotal = 10000); the interval between follow-up data (ARNLmax = 10 seconds). We evaluated the possibilities of the method using the ratio NC and ARNL, which were calculated by formulas $NC_{Rate} = NC_{Current}/NC_{Total}$ and $ARNL_{Rate} = ARNL_{Current}/ARNL_{Max}$.

Figure 2 shows the initial patterns of the attacks. Here, the initial values of the observations are individual points in the time series. Attacks can develop both on one of parameters, and in combination, on two parameters simultaneously. The multi-stage zonal classifier decides about suspicious or malicious user's behaviour, putting trajectories into orange or red zones by NC or ARNL rate.

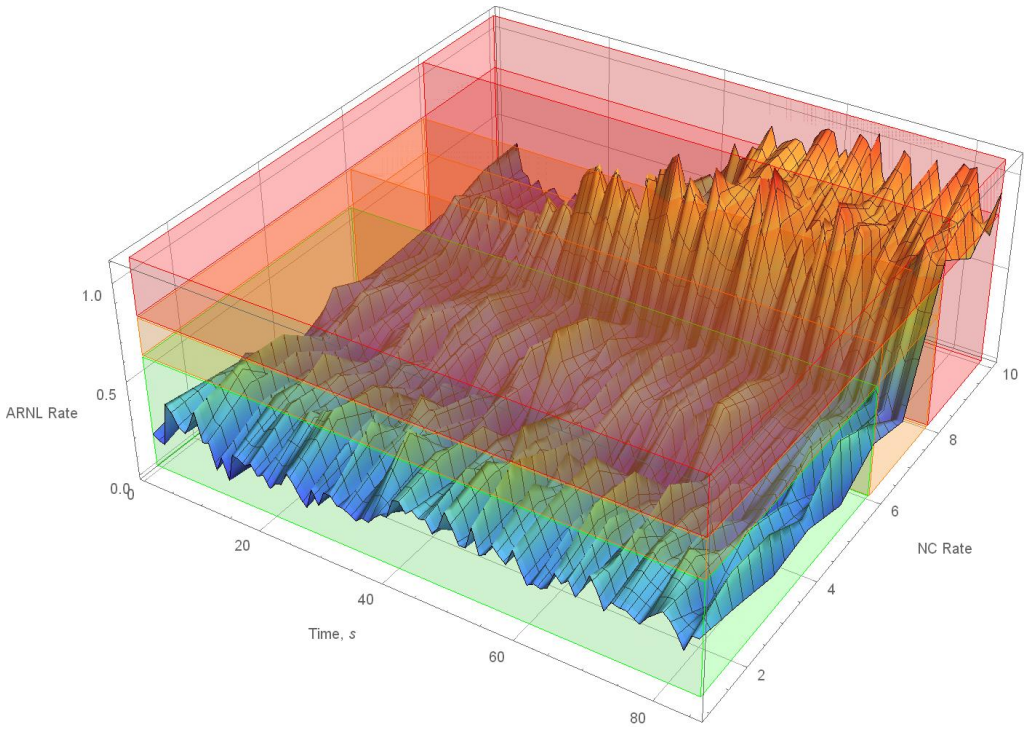
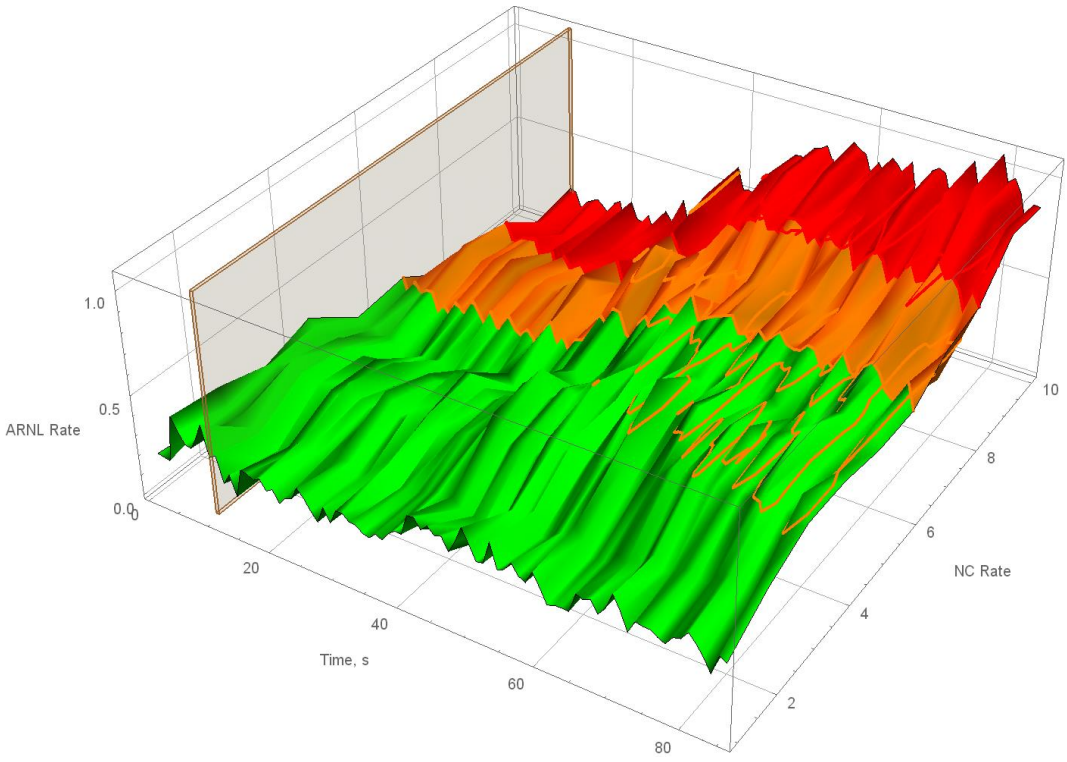


Figure 2: Observations of Slow HTTP statistics

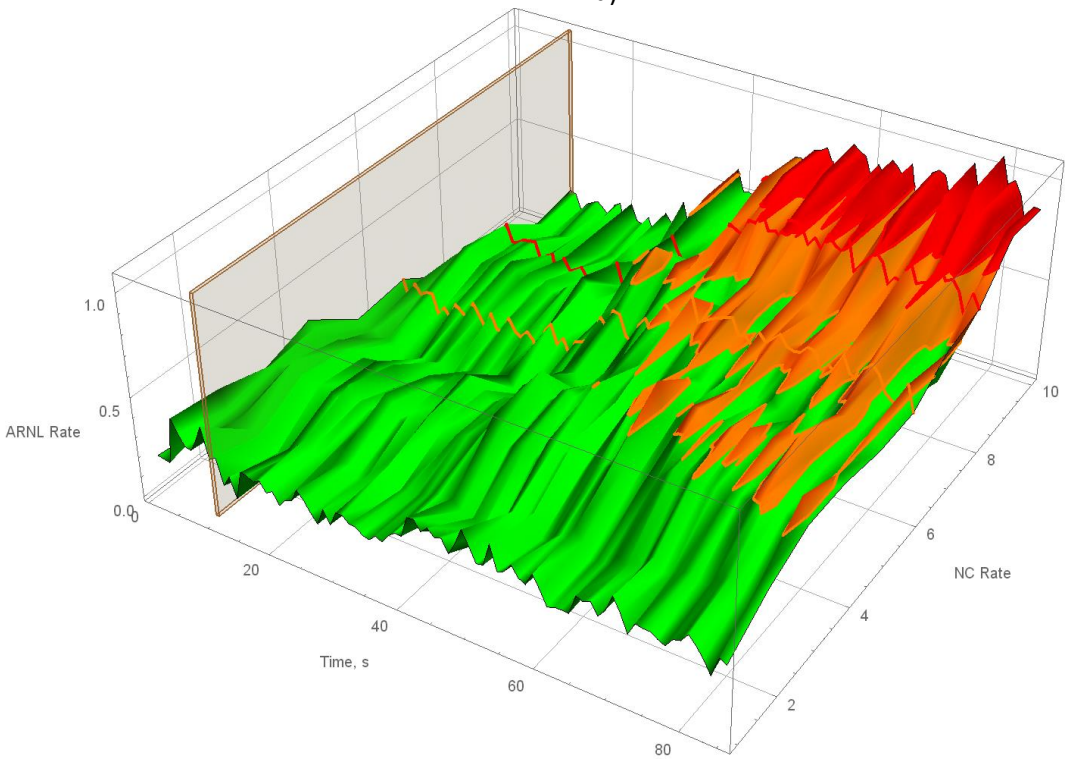
The IDS Forecasting Module, using prediction algorithm applied to the a priori process, decides about suspicious or malicious behaviour based on the results of the forecast. Since against the background of the studied process, the development of process can further lead to multiple trajectories. Predicting the attack means to determine the moment of time when the curve falls into one of the critical zones, as shown in Figure 3.

The study of a separate trajectory of the forecast deserves attention. So, if we take as a basis one particular trajectory (red line in Figure 4) and construct a prediction (blue line) you can see that the method gives a fairly accurate result, which depends on the time of a priori observation.

As we mentioned in [14–15] the number of initial observations affects the accuracy of the forecast. In Figures 4a and 4b, the field of curves shows how forecasts will perform when receiving data from other control points preceding the forecast moment (20s). The probability of error in choosing the correct trajectory depends on the amount of observed raw data. It is logical to assume that, in this case, the accuracy of the forecast will depend too much on the characteristics of the behavior of the trajectory, which leads to abnormal motion, as well as on the observed frequency of anomalies. Thus, the method "selects" the desired trajectory depending on the entry point and the average trajectory.



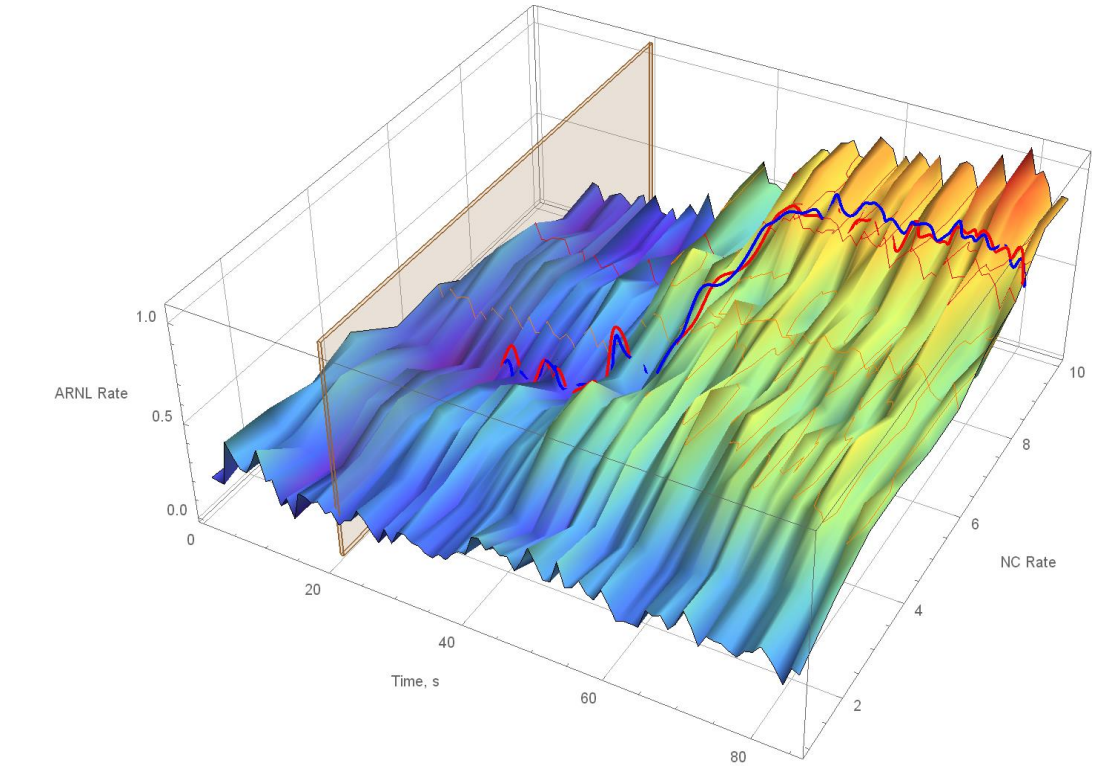
a)



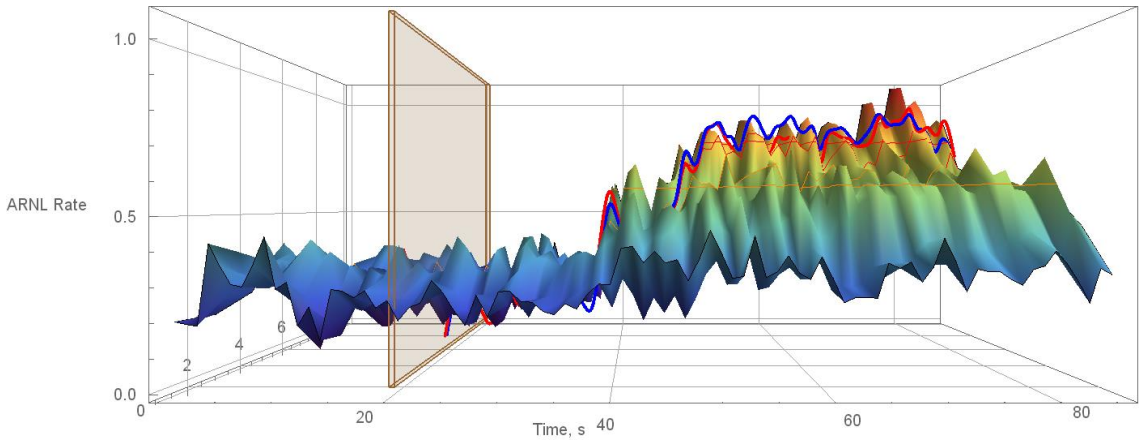
b)

Figure 3: Forecasting a posteriori trajectories at the time of observations of 10 s:
a) by the NC Rate parameter; b) by the ARNL Rate parameter.

An important question is how the accuracy of forecasting depends on the number of a priori observation. We have already considered this issue in [14], where it was shown that in 60 ... 90 s the deviation of the predicted trajectory from the control decreases to 5 ... 0 %. This confirms the adequacy of the prediction model for detecting slow DDoS attacks based on two-parametric predictions.



a)



b)

Figure 4: Research of a separate trajectory of the forecast: a) two-parametric view; b) ARNL view

6. Conclusions

1. Slow HTTP DDoS attacks remain complex enough to detect due to minor changes in traffic parameters. The behavior of attackers who imitate the behavior of legitimate users causes the development of sophisticated technologies for detecting attacks. Existing methods for detecting slow DDoS attacks, based on artificial intelligence, require significant statistical data for training. However, as it was proven in this publication, more promising are methods based on predicting user's behaviour by the Number of Connections and the time of the Average Real Network Latency.

2. Predicting user behavior parameters allows you to detect slow DDoS attacks in advance based on an algorithm for finding unknown future values for a time series of parameters. Using the relative values of NC and ARNL as forecast parameters makes it possible to build a flexible recognition system adapted

to the specifics of a particular system. The proposed method is a combination of artificial intelligence and statistical analysis and uses a self-learning algorithm with sufficient statistics of attacks.

3. Further research in counteracting slow HTTP DDoS attacks can be devoted to issues of multidimensional forecasting at intervals that are not covered by statistics, data noise and for the case of an arbitrary number of parameters.

7. References

[1] M. Y. Arafat, M. Alam, M. Fakrul, A practical approach and mitigation techniques on application layer DDoS attack in web server, International Journal of Computer Applications 131 (2015) 13–20. doi:10.5120/ijca2015907209.

[2] E. Cambiaso, et al., Slow DoS attacks: definition and categorisation, Int. J. Trust Management in Computing and Communications 1(3/4) (2013) 300–319. doi:10.1504/IJTMCC.2013.056440.

[3] T. Lukaseder, et al., SDN-assisted network-based mitigation of slow DDoS attacks, Secure Communications 18.04 (2018). arXiv:1804.06750.

[4] M. Siracusano, S. Shiaeles, B. V. Ghita. Detection of LDDoS attacks based on TCP connection parameters, in: 2018 Global Information Infrastructure and Networking Symposium, GIIS, 2018, pp. 1–6. doi:10.1109/GIIS.2018.8635701.

[5] V. M. Rios, et al., Detection of reductionof-quality DDoS attacks using Fuzzy Logic and machine learning algorithms, Computer Networks 186 (2021) 107792. doi:10.1016/j.comnet.2020.107792.

[6] T. Hirakawa, et al., A defense method against distributed slow HTTP DoS attack, in: 19th International Conference on Network-Based Information Systems, 2016. doi:10.1109/NBiS.2016.58.

[7] L. Calvert, T. M. Khoshgoftaar, Impact of class distribution on the detection of slow HTTP DoS attacks using Big Data, Journal of Big Data 6 (2019). doi:10.1186/s40537-019-0230-3.

[8] I. V. Duravkin, A. Carlsson, A. S. Loktionova, Method of slow-attack detection, Information processing systems 8 (2014) 102–106.

[9] A. Bhardwaj, et al., Experimental analysis of DDoS attacks on OpenStack cloud platform, in: 2nd International Conference on Communication, Computing and Networking, Lecture Notes in Networks and Systems 46 (2019). doi:10.1007/978-981-13-1217-5_1.

[10] I. V. Ruban, D. W. Pribylnov, E.C. Loshakov, A method of detecting a low-speed denial-of-service attack. Science and technology of the Air Force of the Armed Forces of Ukraine 4 (2013) 85–88.

[11] A. Dhanapal, P. Nithyanandam, An OpenStack based cloud testbed framework for evaluating HTTP flooding attacks, Wireless Networks (2019) 570–575. doi:10.1007/s11276-019-01937-4.

[12] A. Dhanapal, P. Nithyanandam, The slow HTTP Distributed Denial of Service Attack Detection in Cloud, Scalable Computing, 20/2 (2019) 285–297. doi:10.12694/scpe.v20i2.1501.

[13] A. Dhanapal, P. Nithyanandam, The slow HTTP DDOS attacks: detection, mitigation and prevention in the cloud environment, Scalable Computing: Practice and Experience 20/4 (2019) 669–685. doi:10.12694/scpe.v20i4.1569.

[14] V. Savchenko, et al., Detection of slow DDoS attacks based on user's behavior forecasting, International Journal of Emerging Trends in Engineering Research 8/5 (2020) 2019–2025. doi:10.30534/ijeter/2020/90852020.

[15] V. Savchenko, et al., Network traffic forecasting based on the canonical expansion of a random process, Eastern European Journal of Enterprise Technologies 3/2(93) (2018) 33–41. doi:10.15587/1729-4061.2018.131471.