Journal of Computations & Modelling, vol.9, no.2, 2019, 1-26
ISSN: 1792-7625 (print), 1792-8850 (online)
Scienpress Ltd, 2019

Cognitive Web Application Firewall to Critical Infrastructures Protection from Phishing Attacks

Konstantinos Demertzis[1] and Lazaros Iliadis[2]

Abstract

Phishing scams and attacks attempt to trick people into providing sensitive personal information such as account login credentials, credit card numbers, banking details and other identifying data. This is done for malicious reasons, by disguising as a trustworthy entity in an electronic communication. It is an example of social engineering techniques used to deceive users and to exploit weaknesses in current web security. It is very popular with cybercriminals, as it is far easier to trick someone into clicking a malicious link in a seemingly legitimate URL than trying to break through a computer's defenses. Nowadays, phishing scammers continually target many critical infrastructures and major financial institutions, companies, government departments, and online service providers around the world. For those infrastructures specifically, skilled phishers use advanced techniques to target both vigilant and naive employees, with destructive often zero-day attacks, including ransomware, malware, bots, spam, spoofing and

[1] School of Engineering, Department of Civil Engineering, Democritus University of Thrace, Greece. E-mail: kdemertz@fmenr.duth.gr
[2] School of Engineering, Department of Civil Engineering, Democritus University of Thrace, Greece. E-mail: liliadis@civil.duth.gr

Article Info: *Received:* August 31, 2018. *Revised:* October 10, 2018.
Published online : April 1, 2019.

pharming. This paper proposes an innovative, ultra-fast and low requirements' Intelligence Web Application Firewall (IWAF) for Critical Infrastructure Protection (CIP). It discusses the design and development of an intelligent tool which employs an evolving *Izhikevich* spiking neurons' approach, for the automated identification of phishing web sites. Additionally, it builds Group Policy Objects (GPO) under Windows Domain for automated prevention of phishing attacks. The reasoning of its core is based on advanced computational intelligence approaches.

Keywords: Phishing Attacks, Machine Learning Web Application Firewall, Izhikevich Spiking Model, Group Policy Objects, Windows Active Directory

1. Introduction

1.1 Critical Infrastructure Protection

Protecting critical infrastructure is of utmost importance for national security, since any kind of future miss (e.g. terrorist attack or system failure) can create complex and dynamic interdependencies, with potential incalculable consequences [1]. The sectors with the most significant Critical Infrastructures are: Energy production and distribution, Information Technology (IT), Transportation, National Defense, Government's Infrastructure and Industry [1], [2].

Today, in the 21st century era, automation and remote control are the most important methods by which critical infrastructure improves the productivity and quality of services provided [1], [2], [3]. From this point of view, the efficient management of industrial IT systems and the introduction of automation systems, require sophisticated Network Control Devices (NCD) that operate with precision, reliability and security. Typical automated control devices are SCADA systems and sensors, used in control loops for the collection of measurements and for the

automation of processes [4]. These systems comprise of interconnected active devices, embedded in real-time industrial networks that allow remote monitoring and process control, even in cases where devices are distributed in remote locations.

1.2 Phishing Attacks and Social Engineering

Phishing (PHI) [5] is an act of deceiving Internet users, in which the 'offender' pretends to represent a trustworthy entity, abusing the incomplete protection afforded by electronic tools and exploiting the ignorance of the user. This is done aiming to obtain the fraudulent acquisition of personal data, such as sensitive private data and codes. The malicious user sends an e-mail (usually a direct message to the 'victim') in which he/she is recommended as a trusted person belonging to a company or organization. This is done many times though the email service and every time the victim is asked to provide some personal information [6]. The basic tool of phishing is *link manipulation* [7], [8]. The user is connected to a web page, e-mail, or instant message that points to a superficially reliable link, which is designed to lead to a different site than the expected one. This is very critical but at the same time it is very easy to create, since in a simple *Html* code it is possible to convert the title of the link at will. This is the basic idea behind fake websites, which lead users to pages visually identical to authentic ones through misleading links, but they belong to the malicious user's server.

PHI can become even more complicated when attackers use almost untraceable malicious methods. Examples of such approaches are the so-called *IDN spoofing*, through which identical URLs can lead to different webpages. This is possible when International Domain Names (IDN) are handled unproperly [8].

Solutions using authenticity certificates are not sufficient, as malicious users themselves can obtain true certificates of authenticity. Often, Phishers even deceive anti-phishing programs, or they can cover their traces using filters such as images or flash files instead of text (an image is placed over the fake URL that shows the true

URL). They can even use JavaScript to cover the true URL with another. The offender can also exploit problems in the code of the authentic website and can cause the attack through it [7], [8].

Other phishing techniques use pop-up windows, tab-nabbing (multiple cards) or even false public networks such as airports, hotels and cafes.

The term *Social Engineering* [9] is used to describe the basic way of misleading and all methods commonly used in PHI. It includes all acts of verbal manipulation of individuals aiming in posting information. The term is mainly related to cheating people for posting insider information that is necessary for access to a computer system. Usually the person who applies it never comes face to face with the deceived one. It is mainly based on human curiosity or greed and ignorance. Many people (due to their credulity or courtesy) do not refuse to give any information to someone who kindly requests them or demands them under alleged "pressure"[10]. The main goal is not always to reveal a code that will allow the malicious user to penetrate a computing system, but more often it is enough to post simple "innocent" information such as simple knowledge of the operating system and its version number. With this information, one can find out if there are "*holes*" in the programs which can be exploited.

Other information that may be collected and which are likely to be useful, such as birth dates or children's names. This data is collected either through conversation or by the so-called social networks and corporate websites [7], [8].

The average person knows the basic functions of the computer and the how to use the internet, without knowing its functional processes. Such a user, cannot recognize phishing footprints, like a varied email address, or a fake URL. At the same time, due to the ignorance of risk, the use of anti-phishing programs is neglected. Even in situations where the user has the appropriate knowledge to detect malware, he/she will often not notice the signs, as he/she may be abstract or busy with something else [7], [8].

The final objective of the cybercriminals is achieved as follows [5],[8]:

- *Misleading Text*, which usually contains misleading links. It may use misspelling (eg www.fasebook.com), or spelling anagrams (eg www.yutoube.com), or replacement of similar letters such as English small L (L) with the capital I.
- *Misleading Images*, which may be the same as the images used by a website, for example the google logo, but when you click on them they lead you elsewhere. An equally common method is images that mimic the operating system of the computer (e.g. Windows logo, Ubuntu).
- *Misleading design*. With the help of misleading text and images, as well as editing the original website code, the hacker can create an entire website with the same design as the authentic one.

It is worth mentioning that if a phishing website manages to combine all the above, in most cases it has 90% successful attacks.

1.3 Critical Infrastructure Protection and Phishing Attacks

For critical infrastructures, specialized phishers use advanced techniques that combine Social Engineering, targeting both the lack of specialized active system security measures and the lack of employee awareness or alertness [8], [9], [10].
The consequences are usually devastating, including 0-days malware or ransomware, aiming to violate SCADA systems and industrial control systems (ICS) in general. Generally, phishing is used to allow a malicious user to gain access to a SCADA or an ICS network [11], [12]. There it remains for a period of secret recognition and from this position it is recording the wider network, until the most appropriate time is found to start its widespread attack [13], [14].
It is important to note that the majority of SCADA and ICS systems used in critical infrastructures, were created to communicate only with machines and equipment in

a single location, when their interconnection was simply a future and perhaps utopian thought [14].

It should also be stressed that these systems were designed with their own protocols to enable automation and control of critical processes in which reliability and availability were extremely important, while security was a secondary factor [12], [14].

Finally, as IT and industrial technology continue to converge, CIP-based officers do not have the specialized expertise to deal with cyber security, so they are unable to cope with specialized threats [14].

2. Literature Review

Given the growing complexity of threats, the ever-changing environment and the need for Critical Infrastructures, such attacks could cause massive economic damages, through data leakage or misuse. In the worst case they could cause even the loss of life of innocent people directly or indirectly. This is another supporting factor for the adoption of intelligent solutions that could prevent, detect and deal with threats or anomalies under the conditions and operating parameters of critical infrastructures [1], [2], [4], [14]. Also, given the passive operation of traditional security systems, which in most cases are unable to detect serious threats, alternative more active and more effective security methods are required [7], [12].

Our research team is specialized in solving such complex digital security problems and it has previously proposed many innovative Artificial Intelligence security applications [15], [16], [17], ,[18], [19], [20],[21], [22], [23].

Qian and Sherif [24] applies autonomic computing technology to monitor SCADA system performance, and proactively estimate upcoming attacks for a given system model of a physical infrastructure. Soupionis et al. [25] proposes a combinatorial method for automatic detection and classification of faults and cyber-attacks occurring on the power grid system when there is limited data from

the power grid nodes due to cyber implications. In addition, Tao et al. [26] described the network attack knowledge, based on the theory of the factor expression of knowledge, and studied the formal knowledge theory of SCADA network from the factor state space and equivalence partitioning. This approach utilizes the factor neural network (FNN) theory which contains high-level knowledge and quantitative reasoning described to establish a predictive model including analytic FNN and analogous FNN. This model abstracts and builds an equivalent and corresponding network attack and defense knowledge factors system.

One the other hand Madhusudhanan et al. [27] proposes a new technique called PHONEY which automatically detects and analysis the phishing attacks. The main idea behind this technique is protecting the users by providing the fake information to the website. This tool is able to detect majority of attacks. This tool can be used as a browser extension to mitigate web based phishing attacks. Craig et al. [28] explained a new method for detecting the phishing site by using web bugs and honey tokens. Web Bugs will be in the form of images that will be used to gather information about the user. Ajlouni et al. [29] proposes two classification algorithms Multi-class Classification based on Association Rule (MCAR) and Classification based on Association (CBA) to detect the phishing websites. Author implemented these algorithms on phishing datasets and the result obtained was very accurate and outperformed SVM and algorithms. Finally, Aanchal and Richariya [30] implemented a prototype web browser which is used as an agent and processes the data from phishing attacks. The user uses the web browser to open the email and if any attack is detected the user will be notified and asked to delete the email. The proposed prototype of web browser will help the user to get notified of possible phishing attacks and will prevent them from opening the suspicious websites.

3. Proposed Framework

3.1 The IWAF approach

Most of the modern threats come from Phishing attacks, which, as alleged, can easily trick even the most suspicious users. A typical example is *Advanced Persistent Threads* (APT) attacks, which can take the mechanical control, the dynamic configuration of the centrifugation or they can reprogram ICS, SCADA and PLC. In this way they can speed up or slow down such operations, leading the critical infrastructures' equipment to destruction or permanent damage, with incalculable consequences [31], [32].

This work proposes the creation of an innovative Computational Intelligence system, which significantly enhances critical infrastructure security mechanisms, with minimal consumption of resources. More specifically, we propose the Intelligent Web Application Firewall (IWAF) which contributes significantly towards Critical Infrastructure Protection (CIP). It is an advanced Phishing Attack detection system.

It is an innovative and fully automated tool for energetic security, which uses an Evolving *Izhikevich* Spiking neurons' model, for the automated identification of the Phishing websites. It also builds Group Policy Objects (GPO).

In general, it is based on well-known theoretical literature such as those outlined below, which are best combined to create a comprehensive intelligent learning system. This system optimally implements a decision rule for the classification and detection of phishing attacks, while this knowledge is transformed into firewall rules to enhance the active security of the infrastructure.

3.2 Izhikevich spiking neuron model

A typical spiking neuron model consists of "*Dendrites*", which simulate the input level of the network, which collects signals from other neurons and transmits them to the next level, which is called soma. The "*Soma*" is the process level at

which when the input signal passes a specific threshold, an output signal is generated. The output signal is taken from the output level called the "*Axon*", which delivers the signal (short electrical pulses called action potentials or spike train) to be transferred to other neurons. A spike train is a sequence of stereo-typed events generated at regular or irregular intervals. Typically, the spikes have an amplitude of about100 mV and a duration of 1-2 msec. Although the same elements exist in a linear perceptron, the main difference between a linear perceptron and a spiking model is the action potential generated during the stimulation time. Furthermore, the activation function used in spiking models is a differential equation that tries to model the dynamic properties of a biological neuron in terms of spikes. The form of the spike does not carry any information, and what is important is the number and the timing of spikes. The shortest distance between two spikes defines the absolute refractory period of the neuron that is followed by a phase of relative refractoriness where it is difficult to generate a spike [33].

Several spiking models have been proposed in the last years aiming to model different neurodynamic properties of neurons. Among these models, we could mention the well-known integrate-and-fire model, resonate-and-fire and Hodgkin-Huxley model. One of the simplest and versatile models is the one proposed by Izhikevich. This model has only nine dimensionless parameters, and it is described by the following equations [33]:

$$C\dot{v} = k(v - v_r)(v - v_t) - u + I \tag{1}$$

$$if \quad v \geq v_{peak} then \begin{Bmatrix} v \leftarrow c \\ u \leftarrow u + d \end{Bmatrix} \tag{2}$$

$$u = \alpha\{b(v - v_r) - u\} \tag{3}$$

Depending on the values of *a* and *b* it can be integrator (b <0) or resonator (b> 0). The parameters *c* and *d* do not affect the sub-threshold behavior (in a steady-state) whereas they affect the general model in the after-spike behavior. The parameter *u* is the membrane potential (membrane potential is the difference in electric potential between the interior and the exterior of a biological cell. With respect to the exterior of the cell, typical values of membrane potential range from –40 mV to –80 mV), *u*

is the recovery current that represents a membrane recovery variable, which accounts for the activation of K+ ionic currents and inactivation of Na+ ionic currents, and it provides negative feedback to u. After the spike reaches its apex (+30 mV), the membrane voltage and the recovery variable are reset according to the equation (5). C is the membrane capacitance of a neuron influences synaptic efficacy and determines the speed with which electrical signals propagate along dendrites and axons, υ_r is the resting membrane potential in the model that is between 70 and 60 mV depending on the value of b, and υ_t is the instantaneous threshold potential which is the critical level to which the membrane potential must be depolarized to initiate an action potential. The parameter k occurs when the neuron's rheobase (*rheobase* is the minimal current amplitude of infinite duration) and input resistance. The recovery time constant is α. The spike cut off value is υ_{peak} and voltage reset value is c. The parameter d describes the total amount of outward minus inward currents activated during the spike and it is affecting the after-spike behavior [33]. Various selections of these parameters can lead to various native operating standards, depending on the objective and the problem it is required to solve.

Following the hypothesis “patterns from the same class produce similar firing rates in the output of the spiking neuron and patterns from other classes produce firing rates different enough to discriminate among the classes,” the Izhikevich model can be applied to solve the specified pattern recognition problem. Let $D=\{x^i, k\}_i^p=1$ be a set of associations composed of p input patterns, where $k= 1,...,$ is the class to which $x^i \in R^n$ belongs. The learning process adjusts the synaptic values of the model in such way that the output generates a different firing rate for each class k, reproducing the behavior described in the hypothesis. In order to use the Izhikevich neuron model to solve the phishing pattern classification problem, it is necessary to compute the input current I that stimulates the model. In other words, the spiking neuron model is not directly stimulated with the input pattern $x^i \in R^n$ but with the input current I. If we assume that each feature of the input pattern x^i

corresponds to the presynaptic potential of different receptive fields, then we can calculate the input current I that stimulates the spiking neuron as [33]

$$I = \mathrm{x} \cdot \mathrm{w} \tag{4}$$

where $w^i \in R^n$ is the set of synaptic weights of the neuron model. This input current is used in the methodology to stimulate the spiking model during T ms.

Instead of using the spike train generated by the spiking model to perform the pattern classification tasks, we compute the firing rate of the neuron defined as [33]

$$fr = \frac{N_{sp}}{T} \tag{5}$$

where N_{sp} is the number of spikes that occur within the time window of length T.

It is necessary to calculate the average firing rate $AFR \in R^K$ of each class, by using the firing rates produced by each input pattern. In this sense, the learning process consists of finding the synaptic values of the spiking model in such way that it generates a different average firing rate for each class k.

Suppose that the spiking neuron is already trained using a learning strategy. To determine the class to which an unknown input pattern x belongs, it is necessary to compute the firing rate generated by the trained spiking neuron. After that, the firing rate is compared against the average firing rate of each class. The minimum difference between the firing rate and the average firing rates determines the class of an unknown pattern. This is expressed with the following equation [33]:

$$cl = argmin_{k=1}^{K}(|AFR_k - fr|) \tag{6}$$

where fr is the firing rate generated by the neuron model stimulated with the input pattern $\tilde{x}$ [33].

In order to achieve the desired behavior at the output of the spiking neuron, it is necessary to adjust its synaptic weights. During the training phase, the synapses of the neuron model w, calculated using a powerful and efficient technique for optimizing non-linear and non-differentiable continuous space functions, which are called DEA [34]. This heuristic algorithm optimizes a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping

whichever candidate solution has the minimum score or error function on the optimization problem at hand. This approach has a lower tendency to converge to local maxima, it evolves populations with a smaller number of individuals and it has lower computation cost. In order maximize the accuracy of the spiking neuron model during a pattern recognition task, the best set of synaptic weights must be found using this algorithm. The function that uses the classification error to find the set of synaptic weights is defined as follows:

$$f\,(w, D) = 1 - Performance\,(w, D) \tag{7}$$

where w are the synapses of the model, D is the set of input patterns and $Performance\ (w, D\ )$ is a function which computes the classification accuracy in terms of (6), given by

$$Performance\ (w, D)=\frac{P_{cc}}{P_t} \tag{8}$$

where P_{cc} denotes the number of patterns correctly classified and P_t denotes the number of tested patterns.

The general training methodology used to train the Izhikevich spiking model with DEA, begins with the creation of a plurality of random populations of candidate solutions in the form of numerical vectors. The first of them are chosen as targets. Then the DEA creates a trial vector to perform the following four steps [33][34]:

Step 1. Randomly select two vectors from the current generation

Step 2. Use the selected to compute the difference vector

Step 3. Multiply the difference vector by the weighting factor

Step 4. Form the new trial vector by adding the weighted difference vector to a third one, randomly selected from the current population.

The trial vector replaces the target one in the next generation, if and only if the first produces a better solution than the current, after comparing the cost value obtained by the fitness function.

3.5 The proposed IWAF Algorithm

The proposed IWAF system, initially receives the network traffic as a *PCAP* file, from which the features of interest are extracted with the help of Java and Python techniques. This approach is discussed in the following chapter 4, from a technical point of view. The proposed *Izhikevich* spiking model, performs classification, based on the exported features, to detect Phishing attacks.

When such an attack is detected, a list of Indicators of Compromise (IOCs) is created. IOCs are pieces of forensic data, such as data found in system log entries or files, that identify potentially malicious activity on a system or network.

The IOCs are transformed to Group Policy Objects (GPOs). A GPO is a collection of settings that define what a system will look like and how it will behave for a defined group of users. Microsoft provides a program snap-in that allows to use the Group Policy Microsoft Management Console (MMC). The selections result in a Group Policy Object. The GPO is associated with selected Active Directory containers, such as sites, domains, or Organizational Units (OUs). The MMC allows the creation of a GPO that defines registry-based polices, security options, software installation and maintenance options, scripts, and folder redirection options. Following a scheduled task, these policies are forwarded to specific OUs of the Active Directory and they are applied for the whole set of users. Essentially, they create rules for the prevention and reduction of the Phishing attacks. Figure 1 presents the whole process.

4. Datasets

Appropriate datasets were selected that most closely simulate the problem under consideration, in order to carry out this research and to evaluate the proposed model. The two data sets used are described in the next chapter.

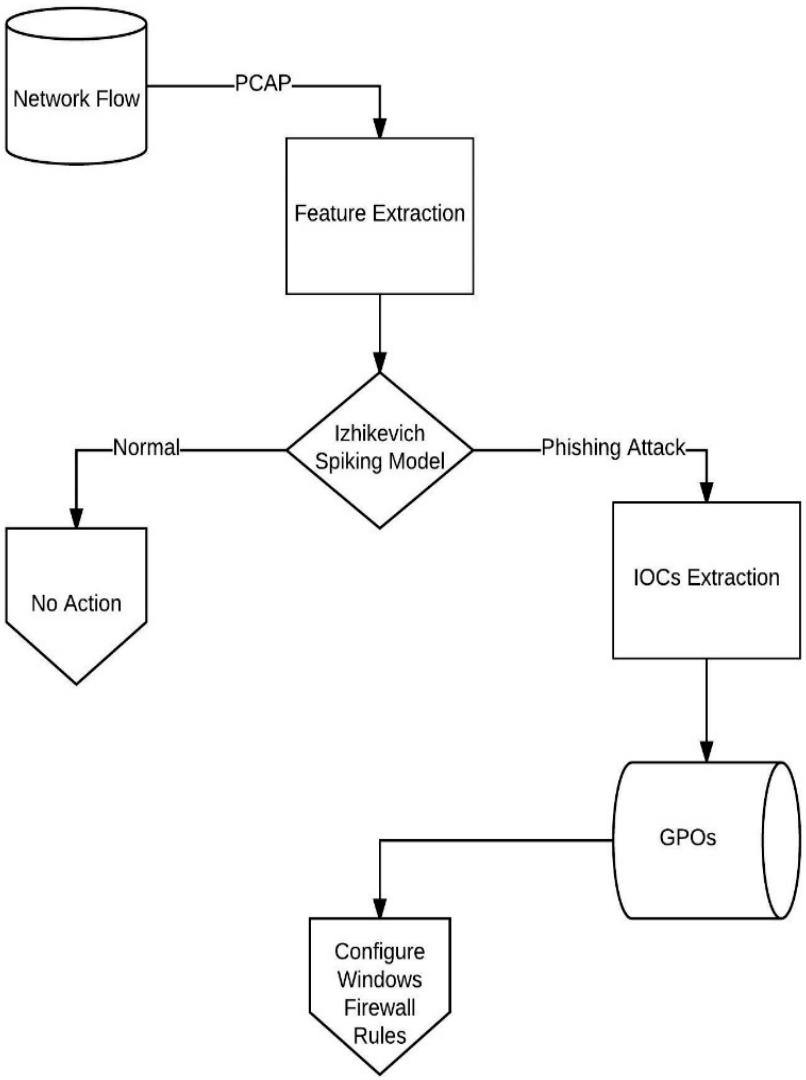

Figure 1: The IWAF Algorithm

4.1 DGA Dataset Preprocessing

A dataset namely Domain Generation Algorithms (DGA) was constructed and used for testing. Totally, 100,000 domain names were used as legitimate ones. They were chosen randomly from the database with the 1 million most popular domain names of *Alexa* [35]. For the malicious domains, the updated list of the

Black Hole DNS database was used [36]. This list includes 16,374 records from domains that have been traced and characterized as dangerous. Moreover 15,000 domain name records were labeled as malicious. They were created based on a time stamp DGA algorithm, with length from 4 to 56 characters of the form 18cbth51n205gdgsar1io1t5.com. Also, 15,000 domain name records which were created with the use of words of phrases coming from an English dictionary, were labeled as malicious. Their length varied from 4 to 56 characters of the form hotsex4rock69burningchoir.com. The full list of features with their corresponding classes is presented in the following Table 1 [37].

Table 1: DGA dataset: Extracted features from domain names
(5 Independent and 1 depended)

ID	***Feature Name***	***Interpretation***
1	length	The length of the strings of the domains.
2	entropy	The entropy of each domain as degree of uncertainty, with the higher values met in the DGA domains.
3	alexa_grams	The degree of coherence between the domain and the list of domains originating from Alexa. This is done with the technique of the probability linguistic model for the forecasting of the next n-gram element.
4	word_grams	The degree of coherence between the domain and a list of 479,623 words or widely used characters. It is estimated with the same method as in the previous one.
5	differences	The difference between the values of alexa_grams and word_grams.
6	**Classes**	**Legit or Malicious.**

Duplicate records and records with incompatible characters were removed. Also, the outliers were removed based on the Inter Quartile Range (IQR) technique [38]. After this preprocessing operation, the DGA dataset contains 136,519 patterns.

4.2 Phishing Dataset

To implement and test our approach, we have used two publicly available datasets i.e., the "*Ham Corpora*" from the "*Spam Assassin"* project [39] as legitimate messages and the emails from "Phishing Corpus" as phishing ones [40]. The total number of emails used in our approach is 4,000 out of which 973 were classified as phishing ones and 3027 as legitimate (*Ham*).

There exists a number of different structural features that allow the detection of phishing. In our approach, we have used 29 relevant features. We have used *Python* and *Javascripts* to parse the Phishing and Legitimate (ham) emails and we have extracted the 29 attributes for each email relation. The features used in our approach are described in Table 2 below:

Table 2: Phishing dataset: Extracted features from Emails
(29 Independent and 1 depended features)

ID	***Feature Name***	***Interpretation***
1	HTML Email	{True,False}
2	IP-based URL	{True,False}
3	Age of Domain Name	{normal>6 month, 1 month<suspicious<6 month, malignan<1 month}
4	Number of Domains	{normal<5, 5<suspicious<10, malignant>10}
5	Number of Sub-domains	{normal<10, 10<suspicious<20, malignant>20}
6	Presence of JavaScript	{True,False}
7	Presence of Form Tag	{True,False}
8	Number of Links	{normal<5, 5<suspicious<10, malignant>10}
9	URL Length	{small <5, 5<mid<15, large>15}
10	URL Based Image Source	{True,False}

ID	*Feature Name*	*Interpretation*
11	Shortening Service	{True,False}
12	Double Slash Redirecting	{True,False}
13	Request URL	{True,False}
14	URL of Anchor	{True,False}
15	Matching Domains	{True,False}
16	Prefix - Suffix	{True,False}
17	SSLfinal State	{True,False}
18	Favicon	{True,False}
19	HTTPS Token	{True,False}
20	Links in Tags	{True,False}
21	Submitting to Email	{True,False}
22	On Mouseover	{True,False}
23	Right Click	{True,False}
24	Pop Up Widnow	{True,False}
25	Iframe	{True,False}
26	Port Number	{Well-known ports, other}
27	Protocol	{http,https,other}
28	Domain in Hostname	{True,False}
29	Obfuscation Characters	{True,False}
30	**Classes**	**{Phishing, Normal}**

Details of the phishing dataset as well as the methodology for collecting, selecting and evaluating the data can be found in [39], [40].

5. Results

In the case of multi-class or binary classification (such as the one performed herein) the estimation of the actual error requires the probability density of the all categories [41][42]. The classification performance is estimated by the employment of a Confusion Matrix (CM), where the main diagonal values (top left corner to bottom right) correspond to correct classifications and the rest of the numbers correspond to very few cases that were misclassified. The number of misclassifications are related to the False Positive (FP) and False Negative (FN) indices appearing in the confusion Matrix. A FP is the number of cases where we wrongfully receive a positive result and the FN is exactly the opposite. On the other hand, the True Positive (TP) is the number of records where we correctly receive a Positive result. The True Negative (TN) is defined respectively. The True Positive rate (TPR) also known as Sensitivity, the True Negative rate also known as Specificity (TNR) and the Total Accuracy (TA) are defined by using equations 9, 10, 11 respectively [41], [42]:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{9}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \tag{10}$$

$$\mathrm{TA} = \frac{\mathrm{TP+TN}}{\mathrm{N}} \tag{11}$$

The Precision (PRE) the Recall (REC) and the F-Score indices are defined as in equations 12, 13 and 14 respectively [41], [42]:

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{12}$$

$$\mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{13}$$

$$\mathrm{F-Score} = 2\mathrm{X}\frac{\mathrm{PRE\ X\ REC}}{\mathrm{PRE\ +\ REC}} \tag{14}$$

The following table 3, presents an extensive comparison for both datasets, by employing competitive Neural Networks' approaches namely: Radial Basis

Function Neural Network (RBFNN), Group Method of Data Handling (GMDH), Polynomial Neural Networks (PNN), Feedforward Neural Networks using Genetic Algorithms (FFNN-GA), Feedforward Neural Networks using Particle Swarm Optimization (FFNN-PSO), Feedforward Neural Networks using Ant Colony Optimization (FFNN-ACO) and Feedforward Neural Networks using Evolution Strategy (FFNN-ES).

Table 3: Comparison between algorithms

Classifier	**DGA Dataset**				**Phishing Dataset**			
	ACC	**RMSE**	**F-Score**	**ROC Area**	**ACC**	**RMSE**	**F-Score**	**ROC Area**
IWAF (Izhikevich) SNM	**98.2%**	**0.3284**	**0.982**	**0.990**	**99.6%**	**0.2951**	**0.996**	**0.995**
RBFNN	89.8%	0.5766	0.900	0.980	91.3%	0.5514	0.910	0.985
GMDH	94.4%	0.5017	0.945	0.955	97.8%	0.3983	0.978	0.980
PANN	90.9%	0.5633	0.910	0.950	96.6%	0.4512	0.965	0.975
FFNN-GA	96.7%	0.4972	0.967	0.970	99.1%	0.3048	0.990	0.990
FFNN-PSO	96.2%	0.4911	0.962	0.975	99.2%	0.3009	0.992	0.990
FFNN-ACO	89.4%	0.5791	0.895	0.900	92.7%	0.5336	0.927	0.950
FFNN-ES	90.1%	0.5716	0.901	0.901	93.5%	0.5125	0.936	0.945

Table 3 shows clearly that the IWAF model has better performance for both datasets which is quite promising considering the difficulties encountered in this project. It is important to say that analyzing and identifying some parameters that can determine a type of threat such as phishing attacks is a partly subjective non-linear and dynamic process.

6. Discussion and Conclusions

This paper presents a reliable, new and low resources' system for the identification of Phishing Attacks. Its reasoning is based on Computational Intelligence methods. IWAF uses the advanced Izhikevich spiking neuron modeling algorithm to identify Phishing-type content, which in most cases carries serious APT cyber-attacks.

The implementation of IWAF was based on the philosophy of automatically creating firewall rules in Windows environment, aiming to protect critical infrastructure.

An important innovation element of the IWAF is the use of Spiking Neural Networks (SNN) in the implementation of the Phishing detection system. SNNs simulate in a most realistic way the functioning of biological brain cells and they realistically model spatiotemporal data.

It is also very important to add the automation system to the firewall rules, as this is the most realistic way of operating and using intelligent systems in the active security of modern information systems, where it is impossible to parameterize and supervise all of their operating systems.

It should also be borne in mind that an equally important innovation is the fact that artificial intelligence has been added to the real-time analysis of real-time networking, which greatly enhances the active defense mechanisms of information systems, especially in critical infrastructures.

It should be stressed that the philosophy of active security, greatly enhances the ways of controlling critical infrastructures, which, due to their significance, are the primary objective of sophisticated modern cyber-attacks. It is obvious that the implementation of the proposed method, which simplifies and minimizes the cost and timing of identifying anomalies in industrial networks, is an important precondition for establishing a risk management and prevention system aiming to protect critical infrastructures.

The performance of the proposed system, was tested on two multidimensional datasets of high complexity, which emerged after extensive research on Phishing Attacks methods, that offered us a realistic depiction of their operating states. The high accuracy of the emerged system, significantly supports the validity of the developed model. The final evaluation of the proposed method was carried out with in-depth comparisons to corresponding Neural Networks' algorithms and it has revealed the superiority of IWAF.

In any case, security critical infrastructure staff should be alert, as it is relatively simple to limit a learning algorithm to a very specific distribution framework. The problem arises from the fact that machine learning techniques are originally designed for stable environments where it is assumed that training and test data are generated by the same (possibly unknown) distribution.

However, in the presence of intelligent and adaptive opponents, this hypothesis is likely to be violated to some extent (depending on the opponent). In fact, a malicious opponent can handle input data that exploits specific vulnerabilities of learning algorithms to compromise the entire security of the system. This method is known as Adversarial Machine Learning (AML). AML is a research field that lies at the intersection of machine learning and computer security. It aims to enable the safe adoption of machine learning techniques in adversarial settings like spam filtering, malware detection, biometric recognition and phishing attacks.

Proposals for the development and future improvements of this system, should focus on further optimizing the parameters of the Izhikevich spiking neuron model used to achieve an even more efficient, accurate and quicker classification process. Also, it would be important to study the extension of this algorithm for analysis and categorization of data streams with online learning methods.

Finally, an additional element that could be studied in the direction of the future expansion of this application, is its operation with methods of self-improvement and re-determination of its parameters (meta-learning) which could fully automate the potential identification of unknown zero-days attacks.

References

[1] W. Hurst, M. Merabti , Fergus P. (2014) A Survey of Critical Infrastructure Security. In: Butts J., Shenoi S. (eds) Critical Infrastructure Protection VIII. ICCIP 2014. IFIP Advances in Information and Communication Technology, vol 441. Springer, Berlin, Heidelberg.

[2] F. Yusufovna, F. Alisherovich, M. Choi, E. Cho, F. Abdurashidovich and T. Kim, Research on critical infrastructures and critical information infrastructures, Proceedings of the Symposium on Bio-Inspired Learning and Intelligent Systems for Security, (2009), 97–101.

[3] W. Hurst, M. Merabti and P. Fergus, Behavioral observation for critical infrastructure security support, Proceedings of the Seventh IEEE European Modeling Symposium, (2013), 36–41.

[4] C. Wang, L. Fang and Y. Dai, A simulation environment for SCADA security analysis and assessment, Proceedings of the International Conference on Measuring Technology and Mechatronics Automation, vol. 1, pp. 342–347, 2010.

[5] Kenneth D. Nguyen, Heather Rosoff, Richard S. John, Valuing information security from a phishing attack, Journal of Cybersecurity, (2017), https://doi.org/10.1093/cybsec/tyx006

[6] K Parsons, A McCormac, M Pattinson, M Butavicius, C Jerram, The design of phishing studies: challenges for researchers, *Comput. Secur.* (2015), doi:10.1016/j.cose.2015.02.008

[7] BB Gupta, A Tewari, AK Jain, and DP Agrawal, Fighting against phishing attacks: state of the art and future challenges, *Neural Comput. & Applic.*, (2016), 1-26, doi:10.1007/s00521-016-2275-y

[8] Suganya, Viswanathan, A Review on Phishing Attacks and Various Anti Phishing Techniques, (2016).

[9] Cullen, Andrea J. and Lorna Armitage, The social engineering attack spiral (SEAS), International Conference On Cyber Security And Protection Of Digital Services (Cyber Security), (2016), 1-6.

[10] Ivaturi, Koteswara and Lech J. Janczewski, A Taxonomy for Social Engineering attacks, (2017).

[11] Cherdantseva, Yulia, Pete Burnap, Andrew Blyth, Peter Eden, Kevin Jones, Hugh Soulsby and Kristan Stoddart. "A review of cyber security risk assessment methods for SCADA systems." Computers & Security 56 (2016): 1-27.

[12] Samtani, Sagar, Shuo Yu, Hongyi Zhu, Mark W. Patton and Hsinchun Chen, Identifying SCADA vulnerabilities using passive and active vulnerability assessment techniques, IEEE *Conference on Intelligence and Security Informatics* (ISI), (2016), 25-30.

[13] Tzokatziou, Grigoris, Leandros A. Maglaras, Helge Janicke and Ying He., Exploiting SCADA vulnerabilities using a Human Interface Device, (2015).

[14] Miller, Bill and Dale C. Rowe, A survey SCADA of and critical infrastructure incidents, RIIT, (2012).

[15] K. Demertzis, L. S. Iliadis, V.-D. Anezakis, An innovative soft computing system for smart energy grids cybersecurity, *Advances in Building Energy Research*, Taylor & Francis, 1-22.

[16] Demertzis K., Iliadis L., *A Hybrid Network Anomaly and Intrusion Detection Approach Based on Evolving Spiking Neural Network Classification,* in: Sideridis A., Kardasiadou Z., Yialouris C., Zorkadis V. (eds) E-Democracy, Security, Privacy and Trust in a Digital World. e-Democracy 2013. Communications in Computer and Information Science, **441**, Springer, Cham, 2014.

[17] Demertzis K., Iliadis L., Evolving Computational Intelligence System for Malware Detection, In: Advanced Information Systems Engineering

Workshops, Lecture Notes in Business Information Processing, **178**, (2014), 322-334. doi: 10.1007/978-3-319-07869-4_30

[18] Demertzis K., Iliadis L., *Bio-Inspired Hybrid Artificial Intelligence Framework for Cyber Security*, in: Daras N., Rassias M. (eds) Computation, Cryptography, and Network Security, Springer, Cham, 2014.

[19] Demertzis K., Iliadis L., *Bio-Inspired Hybrid Intelligent Method for Detecting Android Malware*, in: Iliadis L., Papazoglou M., Pohl K. (eds) Advanced Information Systems Engineering Workshops, CAiSE 2014, Lecture Notes in Business Information Processing, **178**, Springer, Cham, 2014.

[20] Demertzis K., Iliadis L., *Evolving Smart URL Filter in a Zone-based Policy Firewall for Detecting Algorithmically Generated Malicious Domains*, in: Gammerman A., Vovk V., Papadopoulos H. (eds) Statistical Learning and Data Sciences. SLDS 2015, Lecture Notes in Computer Science, **9047**, Springer, Cham, 2015.

[21] Demertzis K., Iliadis L., SAME: *An Intelligent Anti-Malware Extension for Android ART Virtual Machine*, in: Núñez M., Nguyen N., Camacho D., Trawiński B. (eds) Computational Collective Intelligence, Lecture Notes in Computer Science, **9330**, Springer, Cham, 2015.

[22] Demertzis K., Iliadis L., Computational Intelligence Anti-Malware Framework for Android OS, *Vietnam J Comput Sci*, **4**, (2017), 245, https://doi.org/10.1007/s40595-017-0095-3

[23] Demertzis K., Iliadis L. (2016), Ladon: A Cyber-Threat Bio-Inspired Intelligence Management System, *Journal of Applied Mathematics & Bioinformatics*, **6**(3), (2016), 45-64.

[24] Qian Chen and Sherif Abdelwahed, A model-based approach to self-protection in computing system, Proceeding CAC '13 Proc of the ACM Cloud and Autonomic Computing Conference, Arte No. 16, (2013).

[25] Y. Soupionis, S. Ntalampiras and G. Giannopoulos, Lecture Notes in Computer Science, **8985** (2016), DOI: 10.1007/978-3-319-31664-2_29

[26] X. Tao, H. Renmu, W. Peng, X. Dongjie, Applications of data mining technique for power system transient stability prediction, in: Electric Utility Deregulation, Restructuring and Power Technologies, 2004, (DRPT 2004). Proceedings of the 2004 *IEEE International Conference on*, **1**, (2004), 389–389.

[27] Madhusudhanan Chandrasekaran Ramkumar Chinchani Shambhu Upadhyaya," PHONEY: Mimicking User Response to Detect Phishing Attacks", WOWMOM '06 Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, Pages668-672, IEEE Computer Society Washington.

[28] Craig M. McRae Rayford B. Vaughn 2007 ," Phighting the Phisher:Using Web Bugs and Honeytokens to Investigate the Source of Phishing Attacks ",Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS'07).

[29] Aanchal Jain and Prof. Vineet Richariya Implementing a Web Browser with Phishing Detection Techniques, World of Computer Science and Information Technology Journal, 1(7), (2011), 289-291.

[30] Moh'd Iqbal AL Ajlouni1, Wa'el Hadi,Jaber Alwedyan, Detecting Phishing Websites Using Associative Classification, *European Journal of Business and Management* , **5**(23), (2013), www.iiste.org.

[31] Weiss, Joseph, Current Status of Cybersecurity of Control Systems, Presentation to Georgia Tech Protective Relay Conference, (2003).

[32] Demertzis K., Iliadis L., Spartalis S. *A Spiking One-Class Anomaly Detection Framework for Cyber-Security on Industrial Control Systems*, in: Boracchi G., Iliadis L., Jayne C., Likas A. (eds) Engineering Applications of Neural Networks, EANN 2017, *Communications in Computer and Information Science*, **744**, Springer, Cham, 2017.

[33] Vazquez R. Izhikevich neuron model and its application in pattern recognition, *Aust J Intell Inf Process Syst*, **11**(1), (2010), 35–40.

[34] Price K, Storn M, Lampinen A., *Differential evolution: a practical approach to global optimization*, (2005), Springer. ISBN: 978-3-540-20950-8

[35] http://www.alexa.com/

[36] http://www.malwaredomains.com/

[37] https://www.clicksecurity.com/

[38] Upton, G., Cook, I.: Understanding Statistics, (1996) Oxford University Press. p. 55

[39] http://csmining.org/index.php/spam-assassin-datasets.html

[40] J. Nazario, Phishing Corpus, https://monkey.org/~jose/phishing, Accessed June 2016.

[41] Mao, J., Jain, A.K., Duin, P.W., Statistical pattern recognition: A review, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **22**(1), (2000), 4–37.

[42] Fawcett, T., An introduction to ROC analysis, Pattern Recognition Letters, Elsevier Science Inc., **27**(8), (2006), 861-874,
doi: http://doi.org/10.1016/j.patrec.2005.10.010