ISSN 2278-3091

Volume 9, No.4, July – August 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse190942020.pdf

https://doi.org/10.30534/ijatcse/2020/190942020

Neural Network for Modeling the Process of Network Infection and its Counteraction

Cherniak Andrii[1], Pohoretskyi Mykola[2], Farynnyk Vasyl[3], Vilgushynskyi Mykhailo[4], Kaluhina Tetiana[5]

[1]Interagency Scientific and Research Centre on problems of combating organized crime under the National Security and Defense, Kyiv, Ukraine

[2]Department of Justice, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

[3]Department of Criminal Procedure, National Academy of Internal Affairs, Kyiv, Ukraine

[4]Department of Administrative Law, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

[5]Kyiv College of Communication, Kyiv, Ukraine

ABSTRACT

A network attack is an action aimed at increasing the rights over a remote or local computer system, destabilizing it, denying service, and obtaining data from that remote or local computer system.

The intrusion detection system has been learning in the normal operation of the network for some time, forming several clusters of data that characterize the normal operation of a properly configured network. To detect threats, the trained system checks how close the current data is to one of the clusters, which reveals anomalies in the deviation of the values of the new measurements from the average relative to the data on which it was trained.

The simulation system considers the interaction of the attacker and the anomaly detection system from the standpoint of game theory, where the attacker and the system build their strategy with respect to the information they know. The implementation of this simulation system was implemented and tested on an open set of real data on a computer network. The GNG algorithm performed much better than IGNG in the implementation presented. The effectiveness of clustering by the intrusion detection system is almost independent of the number of parameters observed. The speed and quality of the attacker simulation system and the intrusion detection system are satisfactory.

Key words : network attack, modeling process, network infection, simulation.

1. INTRODUCTION

Recent research by International Data Corporation suggests that the volume of digital information in 2019 reached 300 zettabytes. The growth rate of information that needs protection increases significantly faster than the growth of the volume of information as a whole[2]. Approximately 30-35% of digital information requires protection. Only half of them have the appropriate degree of protection [8].

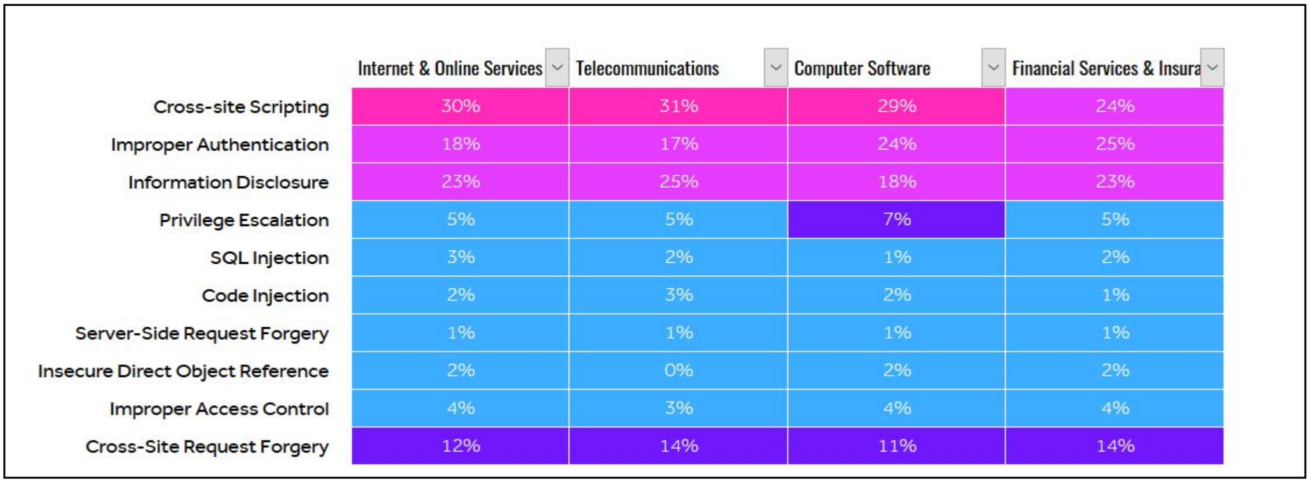
Web applications and computer networks are the most prone to risk from the point of view of information security. The percentage of critical and high risks, as before, remains too high – 24.9% for non-public networks and 19.2% for public web applications [1], and the risk density estimate of 24.3% for internal networks is a concern [4].

The development of modern network technologies is accompanied by an increase in requirements for ensuring the confidentiality of information processing and, as a result, involves a significant modernization of the regulatory and methodological base and standards for information security management. Meanwhile, these documents note that in some cases, methods of managing information security systems focused on the use of quantitative estimates still do not have a sufficiently developed mathematical apparatus for their justification [3-6]. In recent years, there has been a tendency to expand the existing mathematical approaches to substantiating the parameters of security systems by applying the methods of game theory to solving security problems of network technologies, including information security management [7].

Currently, intrusion detection systems are one of the most effective methods of protecting networks. SVV allows you to recognize patterns of abnormal actions, analyze system and network activity, monitor the integrity of files and other is resources, audit the system configuration, and provide control over security functions.

2. MATERIALS AND METHODS

Taking into account the data from HackerOne [11], which is shown in figure 1, let's consider the most influential of them.



	Internet & Online Services	Telecommunications	Computer Software	Financial Services & Insura
Cross-site Scripting	30%	31%	29%	24%
Improper Authentication	18%	17%	24%	25%
Information Disclosure	23%	25%	18%	23%
Privilege Escalation	5%	5%	7%	5%
SQL Injection	3%	2%	1%	2%
Code Injection	2%	3%	2%	1%
Server-Side Request Forgery	1%	1%	1%	1%
Insecure Direct Object Reference	2%	0%	2%	2%
Improper Access Control	4%	3%	4%	4%
Cross-Site Request Forgery	12%	14%	11%	14%

Figure 1: General report on the type of vulnerability.

The percentage corresponds to the volume of reports from industry representatives.

SQL injection is an attack that changes the parameters of SQL queries to the database. As a result, the query takes on a completely different value, and if the input data is not filtered enough, it can not only output confidential information, but also change or delete data. [4]

Tools to combat this type of attack:

For integer and fractional values, before using them in a query, just set the value to the desired type:

```
$id=(int)$id; $total=(float)$total;
```

For string parameters that are not used in the like or regular expressionand:

```
$str=addslashes($str);
```

The XSS attack is an attack on a vulnerability that exists on the server, which allows you to embed some arbitrary code in an HTML page created by the server, which can contain anything and pass this code as a variable, filtering for which does not work, that is, the server does not check this variable for the presence of forbidden characters in it-, <,>, ', ".

The value of this variable is passed from the created HTML page to the server in the script that called it when sending the request. And then the fun begins for the Attacker. In response to this request, the PHP script generates an HTML page that displays the values of variables needed by the attacker, and sends this page to the attacker's browser [9-11].

An active vulnerability is more dangerous, because an attacker does not need to lure the victim through a special link, just embed the code in the database or a file on the server. This way, all site users automatically become victims. It can be integrated, for example,by using SQL injection. Therefore, it is not necessary to trust the data stored in the database, even if when you insert they were processed [12].

An example of a passive vulnerability can be found at the very beginning of the article. Here you already need social engineering, for example, an important email from the site administration asking you to check your account settings after recovering from backup. Accordingly, you need to know the address of the victim or just arrange a spam mailing list or post on some forum, and even not the fact that the victims will be naive and click on your link [13].

The policy interpreter executes the script by setting event handlers. Finishers calculate static traffic parameters while preserving the context, and not just react to a single package. This way, the flow's time history is taken into account.

The principle of operation of the SVB core is shown in figure 2.

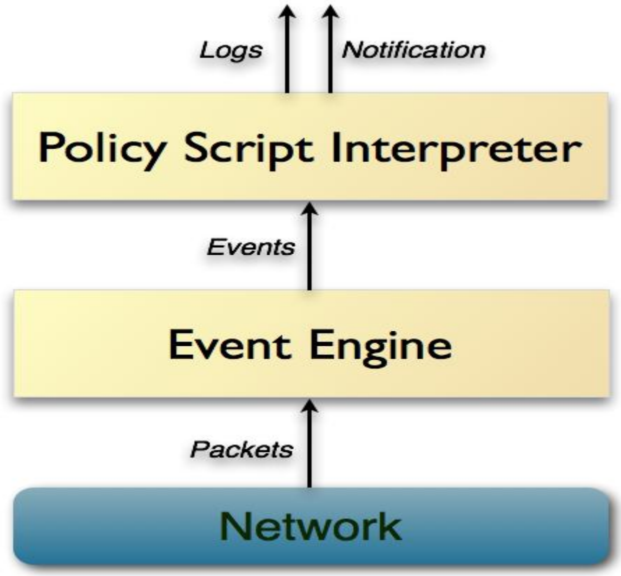


Figure 2: The operating principle of the SVB Bro core.

SVB TripWire, OSSEC and Samhain belong to the host-level systems. They combine the techniques of detection which are known, and methods of detection of anomalous events.

The search mechanism is based on the following: when installing SVV, the system stores in the database of hashes of system files with meta-information about them. Updates to OS packages result in a hash enumeration [14].

When changing any sposter-gated file, the SVB responds to an anomaly.In addition to the observedtion with the operating system files these STS observe the processes, interactions and changes in the system logs. TripWire, OSSEC and Samhain also use a threat database that is constantly updated. These

SVMS have a centralized interface that allows you to analyze data on multiple nodes simultaneously. An example of how OSSEC works is shown in figure 3.

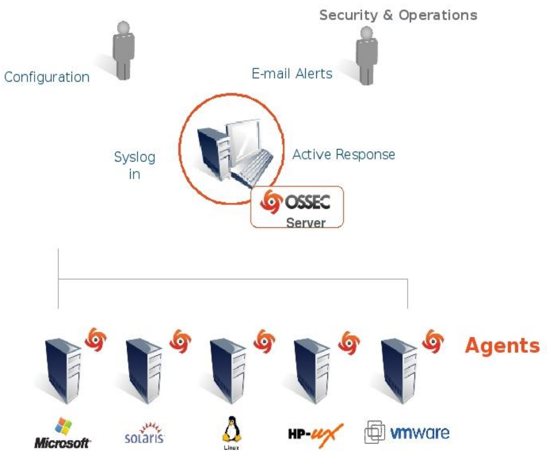


Figure 3.: Example of how OSSEC works

Prelude SIEM and OSSIM SVBS are classified as hybrid systems of the SIEM -Security information and event management type. These systems combine a sensor network and analysisof congestion anomalies, which increases the level of security exponentially due to numerous protection mechanisms [15]. The systems are easily scalable and compatible with many existing IAS based on IDMEF data format, which is shown in figure 4.

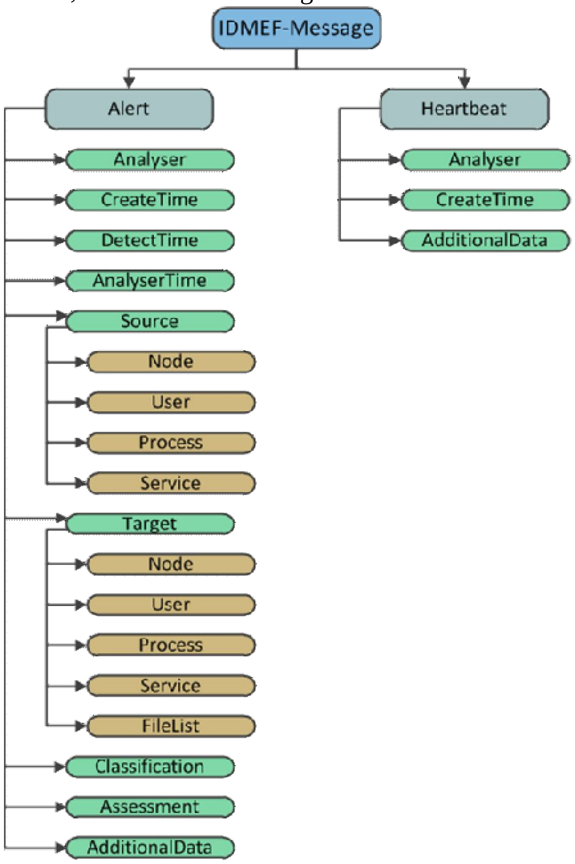


Figure 4: IDMEF message format

The main algorithms for the implemented SVB are GNG and IGNG. GNG or neural GAS – is an adaptive neural network algorithm minspired by the Kohonenself-organizing map, aimed at estimating the density of data distribution. Neurons are inserted into the data space, which adjust the location when the algorithm is running.

Neural programming does not use the hypothesis of data stationarity, which allows us to accept a function that depends on time rather than a fixed dataset [16].

Convention:

$\vec{x}$ - withone data point;

WW мis a vector with a position of neurons no larger than $\beta \times d$, where β is the maximum number of neurons, $and\ d$ is a vector no larger than β;

$error$ - i vector no larger than β;

n_{winner} – learning speed of the winning neuron;

$n_{neighbor}$ – the speed of learning neighbors of the winning neuron;

α_{max} – the maximum age of communication between neurons;

λ – the period between iterations when new neurons are created;

α -coefficient of attenuation of accumulated errors during the creation of new neurons;

γ -errors are extinguished at each iteration;

E – number of epochs.

The main steps of the algorithm that are performed before convergence:

To identify the vector $\vec{x}$.

Find the two nearest to $\vec{x}$, where s-is the nearest neuron, t-is the next one.

Accumulate the distance from the node to the identifiable data vector in the error array:

$$error_s \leftarrow error_s + \|\overrightarrow{w_s} - \vec{x}\| \qquad (1)$$

Update the location $of\ s$ and all neurons that are connected to it by edgesusing different learning speeds:

$$\overrightarrow{w_s} \leftarrow \overrightarrow{w_s} + n_{winner}(\vec{x} - \overrightarrow{w_s})$$
$$\overrightarrow{w_n} \leftarrow \overrightarrow{w_n} + n_{neighbour}(\vec{x} - \overrightarrow{w_n}), \forall w_n \qquad (2)$$

Increase the age of all arcs coming out of the neuron s

If s and$t$ are alreadyconnected to’ by an arc- , take the age of this arc 0, otherwise you must create an arc with age 0 between them.

To remove all arcs, if their age is greater than α_{max}.

To remove all nodes which don't come nand what of the arc.

If the current iteration is divided $by\ \lambda$ without remainder and the number of nodes has not reached the maximum value β, then it is necessary:

Find $the\ u\ neuron$ with the largest accumulated error.

Find the neuron v with the largest accumulated error among the neighbors $of\ u$.

Create a new node between u and v:

$$\overrightarrow{w_r} \leftarrow \frac{\overrightarrow{w_u} + \overrightarrow{w_v}}{2} \qquad (3)$$

Create edges between u--r and v--R delete an edge $u - v$. Reduce the errors $of\ u$ and $v\ neurons$ pass some of these errors to the new neuron:

$$error_u \leftarrow a \times error_u$$

$$error_v \leftarrow a \times error_v$$

$$error_r \leftarrow \frac{a \times error_u}{2} \quad (4)$$

Reduce the error vector:

$$error \leftarrow \gamma \times error \quad (5)$$

The main principles of the IGNG algorithm are as follows:

Using the adaptive resonance network. The closest neuron is examined on the primary eTAPI. If the difference does not exceed a certain threshold , the weight is corrected, otherwise the neuron's coordinate in the data space is changed. If the threshold was not exceeded, then new neurons are created that better approximate the values of the identified data.

Neurons also have an age parameter, unlike GNG, where this parameter is only present in arcs [17].

The new neuron is not active for some time. If the learning stage of the network is over, then the new inactive neurons do not participate in clustering.

The algorithm cycle starts with an empty graph. The parameter σ andits identification as the standard deviation of the training sample:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \quad (6)$$

Parameter $\bar{x}$ is the average between coordinates in the selection.

The main loop reduces $the\ \sigma\ value$, which is the proximity threshold, and calculates the difference between the previous clustering quality level and the level obtained as a result of clustering by the IGNG procedure. Clusterization quality is found by the CHI index – the higher this value, the better. If the difference between indexes after clusterization and before it is negative – the clusterization is completed successfully.

The algorithm has three mutually exclusive steps:

Not a single neuron was found.

One neuron was found that meets the conditions.

We found two neurons that meet the conditions.

After completing one of these steps, the others are not completed.

In the first step, the neuron ispinched, which best approximates the data sample:

$$c_1 = min\ (dist(\xi, w_c)$$

If none was found satisfying $the\ dist\ condition(\xi, w_c) \leq \sigma$ neurona-creates a new inactive neuron with sample coordinates in the data space. If such a neuron was found, the second neuron is searched in the same way. If it doesn't exist, it is created.

If two neurons are found that meet the conditions, their coordinates are corrected as follows:

$$\epsilon(t)h_{c,c_i} = \begin{cases} \epsilon_b, \text{if } c = c_i \\ \epsilon_n, \text{if there is a Cb}'\text{connection } c\text{--}c_I \\ 0, \text{in all other cases} \end{cases} \quad (7)$$

$$\Delta w_c = \epsilon(t)h_{c,c_i}\|\xi - w_c\|$$

$$w_c = w_c + \Delta w_c$$

Where:

$\epsilon(t)$-adaptation step.

c_i – number of the neuron.

h_{c,c_i} – the neighborhood function of neuron c with the winner. It returns 1 for direct neighbors, or 0 in other cases. This means that the adaptation step w is not zero only for direct neighbors.

In the next step, an arc between the winning neurons is created or updated, and the age of all other arcs increases.

Remove all arcs, if their age is greater than α_{max}.

All active neurons, which do not go out of any arc, and the age of all the neurons-neighbors of the winning neuron is growing.

If the age of an inactive neuron exceeds age_{mature} , it becomes active.

The final graph includes only active neurons.

The main advantage of IGNG over GNG is the high speed of learning and data processing and clustering accuracy.

3. CONCLUSION

In progress testing the model was created a database SVB knowledge that consists of out of 10 scenarios attacks. Strategies, based on it, differ from each other how to monitor objects and parameters for monitoring, so and by value m t.

Each scenario was implemented 100 times, and the results are reflect average values prices and percentages detections. Actions SVV and striker they were elected accordingly to balance Nash. At the same time for comparison for each the script was calculated appropriate quantity resources for traditional introduction SVV that has 95 percent accuracy.

Quantity resources needed for the traditional one implementations, many times exceeds offered approach. On the other side accuracy identifying multiple users concedes normal implementations, but, nevertheless, allows you to use such a game-theoretical one optimization in systems with limited resources. How further research, promising the direction will be implementation features automatic time selection monitoring, for example, by entering reflections in the model.

REFERENCES

1. Bernat F., **Immigration and Crime**. Oxford Research Encyclopedias: Criminology and Criminal Justice, 2017.
2. Carlo D. D., Schulte-Cloos J., Saudelli G., Has immigration really led to an increase in crime in Italy? EUROPP, 2018.
3. Ewing W., Martínez D. E., Rumbaut R. G., **The Criminalization of Immigration in the United States**: special report. American Immigration Council, 2015.
4. Stupi E. K., Chiricos T., Gerz M., **Perceived criminal threat from undocumented immigrants**: Antecedents

and consequences for policy preferences. Justice Quarterly, 33, 2016, pp. 239–266. https://doi.org/10.1080/07418825.2014.902093

5. Bernasco W., Block, R., **Robberies in Chicago: A block-level analysis of the influence of crime generators, crime attractors, and offender anchor points**. Journal of Research in Crime and Delinquency, 48(1), 2011, pp. 33–57.
6. Bernasco W., Ruiter S., Block, R., **Do street robbery locations vary over time of day or day of week?** A test in Chicago. Journal of Research in Crime and Delinquency, 54(1), 2017, pp. 244–275.
7. Groff E., Lockwood B., **Criminogenic facilities and crime across street segments in Philadelphia: Uncovering evidence about the spatial extent of facility influence**. Journal of Research in Crime and Delinquency, 51, 2014, pp. 277–314.
8. Haberman C. P., Ratcliffe J. H., **Testing for temporally differentiated relationships among potentially criminogenic places and census block street robbery counts**. Criminology, 53(3), 2015, pp. 457–483. https://doi.org/10.1111/1745-9125.12076
9. Lee Y., Eck J. E., Soohyun O., Martinez N. N., **How concentrated is crime at places?** A systematic review from 1970 to 2015. Crime Science, 6, 2016, p. 6.
10. Weisburd D., **The law of crime concentration and the criminology of place.** Criminology, 53(2), 2015, pp. 133–157.
11. Yu S. V., Maxfield M. G., **Ordinary business: Impacts on commercial and residential burglary.** British Journal of Criminology, 54, 2014, pp. 298–320.
12. Den Hengst M., Staffeleu E., **Different Information Organizations to Produce the Same High Quality Intelligence:** An Overview of the Police Forces in the Netherlands. Policing: A Journal of Policy and Practice., 6(2), 2012, pp. 187–193.
13. Duijn P. A. C., Kashirin V., Sloot P., **The relative ineffectiveness of criminal network disruption.** Scientific Reports, 4 (4238), 2014, pp. 19–29.
14. Framis A., **Illegal networks or criminal organizations: Structure, power, and facilitators in cocaine trafficking structures.** Crime and Networks. Routledge, 2014, pp. 131–147.
15. Azman, A., Alksher, M., Alshari, E., Yaakob, R., & Doraisamy, S. (2019). **Optimization of idea mining model based on text position weight.** International Journal of Advanced Trends in Computer Science and Engineering, 8(1.4 S1), 120–125. https://doi.org/10.30534/ijatcse/2019/1881.42019
16. Azman, N., Ghani, M. K. A., Wicaksono, S. R., & Salahuddin, L. (2019). **The development of iot tele-insomnia framework to monitor sleep disorder.** International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 2831–2839. https://doi.org/10.30534/ijatcse/2019/25862019
17. Babker, A. M., Altoum, A. E. A., Tvoroshenko, I., & Lyashenko, V. (2019). **Information technologies of the processing of the spaces of the states of a complex biophysical object in the intellectual medical system health.** International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 3221–3227. https://doi.org/10.30534/ijatcse/2019/89862019