Methods, Models and Techniques to Improve Information System's Security in Large Organizations

Vladislavs Minkevics and Janis Kampars
Riga Technical University, Kalku 1, Riga, Latvia

Keywords: IS Security, Big Data, Malware, Security Methods and Techniques, DGA.

Abstract: This paper presents the architecture of a modular, big-data based IS security management system (ISMS) and elaborates one of its modules – the domain generation algorithm (DGA) generated domain detection module. The presented methods, models and techniques are used in Riga Technical University, and can be used in any other large organization to stand against IS security challenges. The paper describes how organization can construct IS security management system using mostly free and open source tools and reach it's IS security goals by preventing or minimizing consequences of malware with little impact on employee's privacy. The presented DGA detection module provides detection of malicious DNS requests by extracting features from domain names and feeding them into random forest classifier. ISMS doesn't rely solely of DGA detection and instead uses an ensemble of modules and algorithms for increasing the accuracy of the overall system. The presented IS security management system can be employed in real-time environment and its DGA detection module allows to identify infected device as soon as it starts to communicate with the botnet command and control centre to obtain new commands. The presented model has been validated in the production environment and has identified infected devices which were not detected by antivirus software nor by firewall or Intrusion Detection System.

1 INTRODUCTION

In our digital society, where every person relies on Internet in one or another way, securing information systems have become a challenge like never before. For security reasons institutions like banks, healthcare, insurance organisations must meet certain security standards for example PCI DSS, ISO27001 ("Information technology-Security techniques-Code of practice for information security controls"). These standards define IT security's technical and organisational measures to ensure minimization of IT security risks on data confidentiality, integrity and availability. Nevertheless many reports have surfaced of IT security breaches in companies that complied with the appropriate certifications (The 18 biggest data breaches of the 21st century, 2018) (PCI DSS: Lessons to learn from recent payment card breaches, 2018). Traditional protection methods alone, like firewalls, IDS/IPS systems, are no longer adequate to deal with ever growing number of threats.

According to Trustwave 2019 security report (Trustwave, 2019), one of the major problems in IT security is the substantial amount of time that elapses from penetration until the breach is finally identified. The report shows that median number of latencies in days for internally detected incidents is 11 days. The other critical problem in large organizations is high number of vulnerabilities, especially if different operating systems are used. According to this report 100% of globally scanned resources contained at least one vulnerability and 9% of discovered vulnerabilities were high risk or critical.

To address these risks this paper proposes a modular, Big Data based IS security management system (ISMS). The goal of this paper is to define the architecture of ISMS and elaborate one of its modules – the domain generation algorithm (DGA) detection module, by identifying the best DGA domain name detection features and machine learning algorithm. One of the distinguishing features of this research is use of real-time data from production environment in Riga Technical university (RTU) and the ability to check for true and false positives. Our DNS data originates from devices used by RTU students, employees and guest researchers. This provides more realistic evaluation of machine learning based DGA domain name detection module if compared with existing studies that use publicly available datasets.

Minkevics, V. and Kampars, J.
Methods, Models and Techniques to Improve Information System's Security in Large Organizations.
DOI: 10.5220/0009572406320639
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 632-639
ISBN: 978-989-758-423-7

DGA detection is necessary because currently modern malware is trying to be as stealthy as possible. Especially this applies to botnets. DGA was introduced by botnets, so that the domain name is continuously changing and is difficult to block. Infected device always knows which domain it should resolve to reach the botnet's command and control centre. Most of these algorithmically generated domains are very awkward to human eye.

This work is structured as follows. Section 2 provides a review of the related work in the field of IS security management and DGA detection. Section 3 presents the IS security management background in Riga Technical University and further motivates the development of ISMS. Section 4 defines the architecture of the ISMS. Section 5 develops and evaluates the DGA detection module in production environment. Section 6 concludes and provides directions for future research.

2 RELATED WORK

To fight with IS security challenges unified security systems must be used. Alguliyev et al (Alguliyev & Imamverdiyev, 2014) suggest to use Big Data for IT security, addressing challenges like advanced persistent threats, detections of data leakage, incorporation of forensics, fraud and criminal intelligence. Nevertheless, they list several challenges in the Big Data field at time of writing: privacy, lack of Big Data based detection algorithms, security visualization problems and lack of skilled personnel. Some of these challenges, like Big Data based detection algorithms have already been addressed today.

There are several studies concentrating on the DGA detection problem. Researchers have defined the most important features that can be used to train machine learning algorithms for DGA domain name identification. Feature selection is covered by (Barbosa, Souto, Feitosa, & El-Khatib, 2015) (Jonathan Woodbridge, Hyrum S. Anderson, Anjum Ahuja, 2016) (Jose Selvi, Ricardo J.Rodríguez, 2019). Not all of the proposed DNS features and resulting models contribute to better detection of DGA domain names.

Mowbray et al. (Mowbray & Hagen, 2014) suggest that unusual distribution of second-level string lengths in the domain name is a good indicator for it being a DGA generated domain name. The proposed algorithm looks for domains with an unusual distribution of lengths, however it can only detect DGAs that are used for second-level domain fluxing. Once domains from a new malware DGAs have been detected, they use it to retrain a classifier so that the new DGAs can be detected with a better accuracy.

Truong et al. (Truong & Cheng, 2016) propose to use decision trees to train a DGA classifier. The algorithm reports accuracy of 92.3%. The research suggests using length as one of the features for DGA detection. Authors conclude that DGA detection with machine learning algorithms should not be used as the only solution for botnet detection, since new generation botnets tend to use a technique called domain fluxing.

A research by Ahluwalia et al. (Ahluwalia, Traore, Ganame, & Agarwal, 2017) shows that domain length plays an important role in DGA detection and has great impact on the detection accuracy. Decision trees based DGA generated domain name detection model accuracy for 6-character long domain names is 85,15%, while it reaches 94% for 10-character long domain names. Authors use n-grams to construct features. The experimental results show that random forests have slightly better performance in DGA generated domain name detection.

Selvi et al. (Jose Selvi, Ricardo J.Rodríguez, 2019) analyse 32,000 malware domains and propose to use a set of lexical features based on masked n-grams as features detecting DGA generated domains. The research distinguishes 18 features and concludes that a combination of lexical and statistical features together with bigrams shows the best results in terms of accuracy. The performed experiments indicate that the most important features are unigrams, bigrams and trigrams, their standard deviation as well as number of consonants divided by domain name length. Three different machine learning models are examined: nearest neighbours, decision trees, and random forests, while the latter achieves the best accuracy.

Majority of the previously done studies in this area relies on publicly available datasets, therefore due to poor generalization models might not be efficient if used in the production environment.

J. Peck et al. (Peck, u.c., 2019) suggest that using machine learning algorithm as the only means for detecting DGA generated domain names is not sufficient and other IS security management methods should also be employed. Another reason for using additional approaches for malware detection is masking of DNS traffic using DNS over TLS (Z. Hu,L. Zhu,J. Heidemann,A. Mankin,D. Wessels, 2016) and DNS over HTTPS (Mozilla.org, 2020). Many companies like Cloudflare, Quad9, Google have already announced public DNS resolver services

via DNS over TLS (Quad9, 2020) (Cloudflare, 2020) (Google, 2019).

There are studies concentrating on the IP Flow analysis problem in real time. IP Flow data is valuable because it provides information about connections, TCP/IP flags and data amount sent and received. This data can be used to detect suspicious activity and compromised devices. Jirsik et al. (Jirsik, Cermak, Tovarnak, & Celeda, 2017) claim that IP Flow data aids traditional monitoring with the ability to run analytical queries that are evaluated in real time with high throughput, low latency, and good scalability, all at the same time and allows security analysts to perform real-time analysis on network data and detect network attacks instantly, and provides them with a deep understanding of the network via in-depth situational awareness.

3 BACKGROUND

The development of first generation IS security management platform (ISSMP) in RTU started in 2014. Initially it consisted of Suricata IDS (Suricata, 2020) and Python scripts that helped chief security officer to analyse IDS generated data.

Platform relied on open source products and the main automation was based on cronjob script execution. Python scripts were used to cover all phases of the automated threat detection logic. The system sent notifications to specific user whose device was classified as infected and provided additional details to Chief security officer.

Platform allowed RTU to dramatically decrease the number of notifications by Information Technology Security Incident Response Institution of the Republic of Latvia (CERT) regarding compromised IPs (see Fig.1.).

The ability to automatically block IPs in firewall, which was introduced in 2019, enabled further reduction of incidents, since it was very effective measure in scenarios where the notified user is not taking action to clean his/her device from malware.

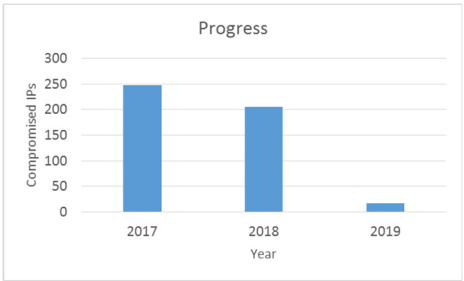

Figure 1: Compromised IPs in RTU by Latvian CERT.

Despite the achieved progress there is still room for improvement, considering that not all infections are detected by CERT and amount of data for processing increased. Several issues have been discovered and ISSMP was unable to address them due to various architectural limitations. The amount of processable data in some of system's components cannot be handled by standalone Python scripts. The other discovered problem is the latency of data processing, which needs to be minimized. To address these challenges Big Data concept was adopted and a new generation of IS security management platform (ISMS) is presented in Section 4.

4 ISMS ARCHITECTURE

The next generation of IS security management platform (ISMS) is being developed at RTU. The system is based on well proven Big Data technologies and it will incrementally replace the previously used system.

The ISMS has both preventive and detective capabilities. It contains vulnerability management component which provides regular manual scans and preparation of analytical reports for the responsible personnel. Automation of vulnerability identification process is used to:

1) Identify hosts that should be scanned, for example using specific open port. For this purpose, Nmap (Nmap, 2020) is used;
2) Normalize the results from Nmap scan and feed IP addresses into Nessus vulnerability scanner (Tenable, 2020);
3) Scan IP addresses for vulnerabilities;
4) Parse the result of Nessus scanner and extract Critical vulnerabilities with corresponding IP's;
5) Sort the results according to responsible personnel and send an email with IP addresses and vulnerabilities that their devices have.

The architecture of the proposed ISMS is based on capability driven development (Sandkuhl & Stirna, 2018) which allows better traceability of organisational IS security management goals and their fulfilment (Minkevics & Kampars, 2018). ISMS contains open source products and is capable of handling security incidents in an efficient manner with respect to privacy according to EU General data protection regulation (EU, 2016) consists of two major and four minor parts (see Fig.2.).

The Big Data paradigm is used in the ISMS to address the following challenges:

1) data are very large and cannot be processed by traditional methods;
2) data are produced with great velocity and must be captured and processed rapidly;
3) data have different structure. (Alguliyev & Imamverdiyev, 2014).

These challenges can be addressed with parallel processing and tools built to handle Big Data like Hadoop, Apache Spark, Cassandra, Apache Kafka.

In Big Data context, an open source, distributed platform Apache Kafka (Kafka, 2020) which handles messages in real time was adopted. Introducing Kafka into the automated security management process in RTU has allowed to decrease the reaction time by up to 10 seconds (previously it took up to 80 seconds to react on malicious activity). Further improvement of the ISMS will provide even greater reduction of the latency.

The privacy goals in the ISMS are reached by not linking the IP address to a user prior to registered suspicious activity. While constant IP address user lookups for all users can speed up the process of incident analysis, it has negative effect on the user privacy. Authors believe that the potential delay of a few seconds for IP user lookup when malicious activity is identified is neglectable and the value of improved privacy is much greater, especially if DNS analysis is in place.

The ISMS provides the following means of automation:

1) Automated Suricata IDS (Suricata IDS , 2020) rule update and message transfer to Kafka;
2) Automated message forwarding from Firewall to Kafka;
3) Automated DNS request transfer to Kafka;
4) Automated message retrieval from Kafka and further analysis:
   a. Checking if known malware DNS requests (Malware domain list, 2020) are in DNS data;
   b. Checking if DGA generated domain name detection module has identified malicious DNS requests (see Section 5);
   c. Checking if the generated alert is not false positive;
5) Notifying user and Chief security officer.

The ISMS consists of analysis and actions modules, where the analysis module consists of many sub-modules for vulnerability identification, open or vulnerable service identification, detecting malicious behavior by a device or user. The user activity sub module consists of portal login information analysis, where information from successful portal logins are collected and analyzed to identify if user has logged in from other country and home country within last 24 hours. A similar analysis is performed for Office 365 user logins. DHCP lease logs are collected to obtain information about device, so change of IP address does not impact accuracy of device identification. Suricata IDS and Firewall logs and DNS traffic data are collected for detecting infected devices. Switch logs are collected to obtain information about the device's physical location if it is part of the local network. IP flow data is obtained and sent to Apache Kafka. At the moment IP Flow data is used to manually analyze and confirm suspicious IP address activity. During the analysis the full picture of connections and DNS requests is created for time period of possible malicious activity. Availability of this information supports well-motivated data-driven decisions by the Chief security officer. The traffic data module is used when brute force is detected and collects full network data of a suspected device for the period of one minute.

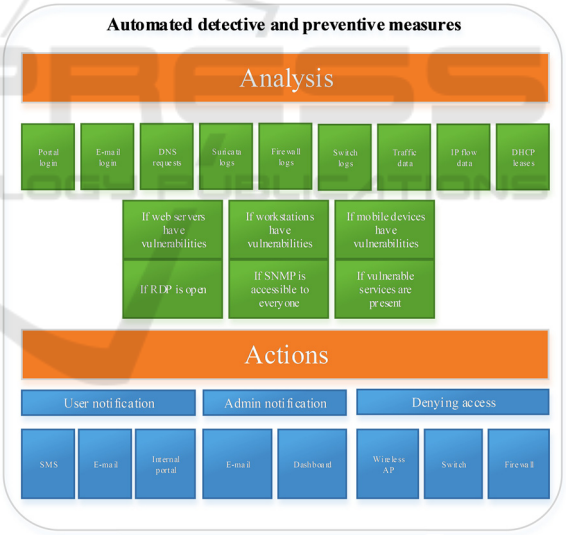

Figure 2: Architecture of ISMS system.

The action component consists of different notification modules which provide user notification via SMS, e-mail and internal portal. The access denial module interacts with firewall API to deny access for specific device in case user is not taking action to stop the detected malicious activity.

5 DGA DETECTION MODULE IMPLEMENTATION AND EVALUATION

The following steps were performed to develop the DGA generated domain name detection model:

1) preparation of training dataset,
2) feature and classifier selection based on related work research and experiments,
3) performance evaluation.

The training data was collected for 3 months (from 04.10.2019-04.01.2020) by recording actual DNS resolution requests in RTU production environment. To enable training of the classifier, the acquired domain names needed to be classified into legitimate and malicious domains. This was done by following the logic described in Table 1 (rule-based automated semantic analysis and information retrieval from external sources like virustotal.com and quad9.com) and Table 2 (semi-automated classification of DNS record based on output from Table 1). Prior to that, the gathered data was cleaned from DNS requests for non-existing domains (the requested domain name contains a spelling error). This was done by checking whether the requested domain names is in icann.org root zone database (ICANN root zone, 2020).

Table 1: Specific ruleset No.1. dataset creation.

Rule No.	Description	Examples
1	IF DOMAIN CONTAINS NO WOWELS THEN SUSPICIOUS++	0sntp7dnrr.com, pnghst.com
2	IF DOMAIN CONTAINS NO CONSONANTS THEN SUSPICIOUS++	3458ee.com, o5o4o6.com
3	IF DOMAIN CONTAINS ONLY NUMBERS THEN SUSPICIOUS++	127777.com, 10000114.com, 12688888.com
4	IF DOMAIN CONTAINS ONLY HEXIDECIMAL THEN SUSPICIOUS++	442d9f2ac50ca502.com, 8cb0309458c7b35e.com
5	IF DOMAIN CONTAINS 3 WOWELS IN A ROW THEN SUSPICIOUS++	zwyr157wwiu6eior.com, zy16eoat1w.com
6	IF DOMAIN CONTAINS 3 CONSONANTS IN A ROW THEN SUSPICIOUS++	yqezqofkb1nnmz.com, 6l1twlw9fy.com
7	IF DOMAIN CONTAINS 5 WOWELS IN A ROW THEN SUSPICIOUS++	booooooom.com, iiiiiiiiiii.net
8	IF DOMAIN CONTAINS 5 CONSONANTS IN A ROW THEN SUSPICIOUS++	yqezqofkb1nnmz.com, eclkmpbn.com
9	IF DOMAIN CONTAINS NUMBERS THAT ARE NOT IN A ROW THEN SUSPICIOUS++	eh8jq4cmq8j9g5.com, 5kv261gjmq04c9.com

Table 2: Specific ruleset No.2. dataset creation.

Rule No.	Pseudo Code of specific rule used
1	IF SUSPICIOUS == 0 THEN IF = EXPERT IDENTIFIED IT AS MALICIOUS DOMAIN_MALICIOUS = 1; EXIT ELSE DOMAIN_MALICIOUS = 0; EXIT ELSE GOTO RULE2
2	IF DOMAIN IN ALEXA TOP 100000 THEN DOMAIN_MALICIOUS = 0; EXIT ELSE GOTO RULE3
3	IF DOMAIN IN QUAD9 AS MALICIOUS THEN DOMAIN_MALICIOUS = 1; EXIT ELSE DOMAIN_MALICIOUS = 0; EXIT
4	IF DOMAIN IN VIRUSTOTAL AS MALICIOUS THEN DOMAIN_MALICIOUS = 1; EXIT ELSE DOMAIN_MALICIOUS = 0; EXIT

Additionally, top 30000 domains from alexa.com were added to the training dataset. The final dataset consisted of 67749 DNS records, where 65999 records were marked as legitimate and 1750 were marked as malicious. The following experiments were conducted after training dataset preparation:

Experiment 1 - selecting the most appropriate classifier algorithm for DGA. Algorithm is selected based on existing studies and practical experiments with selected training set. The initial list of features were chosen based on research by Selvi et al. and are summarized in Table 3.

Table 3: Features used to detect DGA.

Feature No.	Description	Example for 7hu8e1u001.com
F1	length	10
F2	Unigram average	28.39
F3	Bigram average	0.257
F4	Trigram average	0.229
F5	Unigram standard deviation	0.0
F6	Bigram standard deviation	3.459
F7	Trigram standard deviation	3.430
F8	Vowels to length	0.300
F9	Consonants to length	0.100
F10	Unique chars to length	0.001

Due to the fact that natural distribution between legitimate and malicious DNS requests could result in a model that has poor DGA detection capabilities, experiments were performed to determine the optimal training dataset distribution between legitime and malicious domain names. We chose four classifiers: Random forest (RFC), Decision trees (DTC), Neural networks (NNC), and C-support vector classification (SVM). To get more objective results, we split our

dataset into 18 datasets. We used python *train_test_split* library for splitting. Only legitimate domains were split. Malicious domains were the same in every dataset. For every training set 10fold cross validation has been performed using *sklearn* python library.

Classificators' performance for every iteration is shown in Figures 3-6. Y-axis shows precision, recall, F1 score and accuracy percentage values, where's x-axis corresponds to number of legitimate domains in the specific dataset.

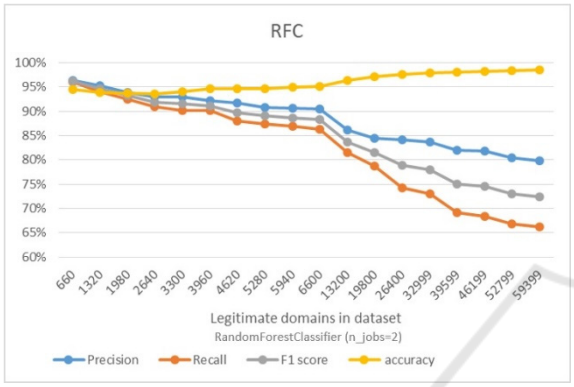

Figure 3: RFC performance.

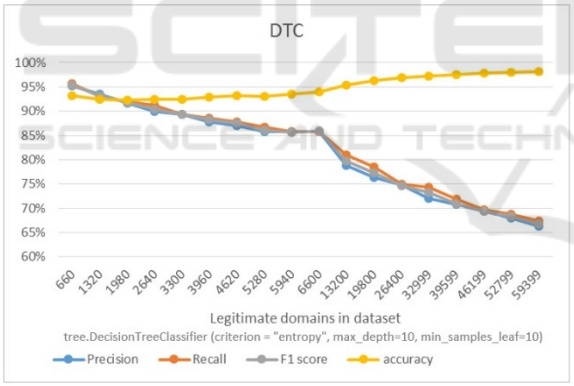

Figure 4: DTC performance.

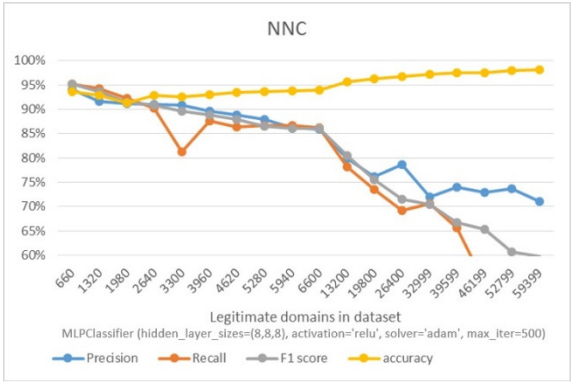

Figure 5: NNC performance.

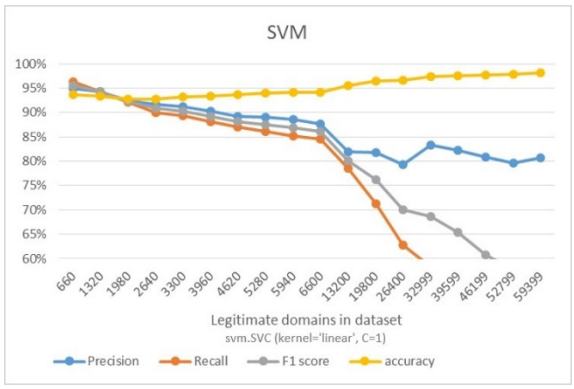

Figure 6: SVM performance.

The performance was evaluated by getting the mean values of every iteration's precision, recall, F1 score and accuracy (see Table 4).

Table 4: Performance measures of experiment 1.

-	Precision	Recall	F1-Score	Accuracy
RFC	0.883	0.822	0.851	0.959
DTC	0.816	0.823	0.819	0.948
NNC	0.831	0.782	0.804	0.949
SVM	0.866	0.753	0.798	0.951

Results of Experiment 1. Our results, similarly to Ahluwalia et al (Ahluwalia, Traore, Ganame, & Agarwal, 2017) proved that better performing machine learning algorithm for our task is Random forest classifier (RFC). Random forest classifier is further used to identify DGA in real-time environment.

Experiment 2 – identification of feature set where our chosen classifier performs best. Specific feature sets were selected (see Table 5).

Table 5: Feature sets.

-	SET1	SET2	SET3	SET4	SET5
F1	1	1	1	0	1
F2	0	1	0	0	1
F3	1	1	1	1	1
F4	1	1	1	1	1
F5	0	1	0	0	1
F6	1	1	1	1	1
F7	1	1	1	1	1
F8	1	1	1	1	1
F9	1	1	1	1	1
F10	0	0	1	0	1

The performance was measured using 10fold cross validation for every feature set and getting the mean values of precision, recall, F1 score and accuracy. Additionally, 2 datasets were created with 15

malicious and 15 legitimate domain names that were not in dataset. This was done to evaluate how many false positives will every classifier produce. The false positives were calculated by adding incorrectly identified domains of every iteration. The result is shown in Table 6.

Table 6: Performance measures of experiment 2.

-	Malware FP	Legit FP	Precision	Recall	F1-Score	Accuracy
SET1	54	0	0.881	0.825	0.851	0.959
SET2	55	0	0.877	0.823	0.848	0.958
SET3	50	0	0.880	0.816	0.846	0.957
SET4	59	0	0.878	0.820	0.847	0.957
SET5	47	0	0.883	0.822	0.851	0.959

Results of Experiment 2. Experiments showed that (see Table 6) the best performance of RFC classifier is achieved by using all 10 features, which is why all features were also used by the automated DGA discovery module. It was implemented in production real-time environment using Apache Spark. DNS traffic is being aggregated in 5-minute windows.
The RTU production environment tests using the proposed DGA discovery module were performed from December 24, 2019 until December 29, 2019.

Table 7: Practical results of implemented DGA detection module.

DGA detection module performance	No.
Detected as malware DNS, and not in training database	865
Virustotal.com detected DNS as malicious	53
Virustotal.com detected DNS as suspicious	33
Quad9.com detected DNS as malicious	75
Detected as malware DNS, and not in training database	865

The results (see Table 7) show that the DGA detection module can be successfully adopted in real time environment using Big Data concept. Since the beginning of experiments, more than 30 DGA detected infection cases (mostly student devices) were confirmed by the end-device owner using other antimalware systems.

6 CONCLUSIONS AND FUTURE WORK

Methods and techniques described in this paper were adopted by RTU and enabled to address IT security challenges every organisation faces nowadays. Big Data is a valuable source for improving overall IT security in organization. Use of Big Data concepts are a must for handling large amounts of data in real time, as it is in cybersecurity. New opportunities arise to use IP flow data with machine learning algorithms and improve threat identification even more.
By using the ISMS:

1) The overall rate of false positives dropped;
2) Malware identification rate increased;
3) Malware unknown by security vendors have been discovered;
4) Reaction speed on incidents raised, and now it is approximately 10 seconds;
5) We gained ability to scale and extend the solution as necessary.

The adoption of the new system has stepped up RTU to real time automated IS security risk management process adoption. More modules are being developed and integrated in the platform to increase the accuracy of threat detection. Experiments with using machine learning in other areas, like IP flow data analysis are being performed. The level of the ISMS autonomousity will be raised by providing better integration with systems like corporate firewalls, which will enable automatic blocking of infected devices. The platform, its methods and techniques can be adapted by any large organisation facing similar challenges.

ISMS has already proven its effectiveness, by detecting approximately 40 new malware infections per month, which would remain undetected by the previous generation of IS security management system. The DGA detection module alone has allowed to identify more than 30 infections. In future we will continue to expand ISMS system by adding machine learning based IP flow data analysis module.

REFERENCES

"Information technology-Security techniques-Code of practice for information security controls". (n.d.). 27002:2015, ISO/IEC.

Ahluwalia, A., Traore, I., Ganame, K., & Agarwal, N. (2017). Detecting Broad Length Algorithmically Generated Domains. In I. Traore, I. Woungang, & A. Awad (Ed.), *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments* (pp. 19-34). Cham: Springer International Publishing.

Alguliyev, R., & Imamverdiyev, Y. (2014). Big Data: Big Promises for Information Security. *2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT)*, (pp. 1-4).

Barbosa, K., Souto, E., Feitosa, E., & El-Khatib, K. (2015). Identifying and Classifying Suspicious Network

Behavior Using Passive DNS Analysis. *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, (pp. 160-167).

Cloudflare. (2020, 01 28). *DNS over TLS*. Retrieved from https://developers.cloudflare.com/1.1.1.1/dns-over-tls/

EU. (2016, 04 27). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Da*. Retrieved 10 24, 2018, from https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1532348683434&uri=CELEX:02016R0679-20160504

Google. (2019). *Google Public DNS now supports DNS-over-TLS*. Retrieved from https://security.google blog.com/2019/01/google-public-dns-now-supports-dns-over.html

ICANN root zone. (2020, 01 28). (ICANN.org) Retrieved 01 17, 2020, from http://stats.research.icann.org/dns/tld_report/archive/index.html

Jirsik, T., Cermak, M., Tovarnak, D., & Celeda, P. (2017). Toward Stream-Based IP Flow Analysis. *IEEE Communications Magazine, 55*(7), 70-76.

Jonathan Woodbridge, Hyrum S. Anderson, Anjum Ahuja, D. (2016). Predicting Domain Generation Algorithms with Long Short-Term Memory Networks. 13. Arlington,VA 22201: Endgame, Inc.

Jose Selvi, Ricardo J.Rodríguez, E.-O. (2019). Detection of algorithmically generated malicious domain names using masked N-grams. *Elsevier, Volume 124*, Pages 156-163.

Kafka, A. (2020, 01 28). (Apache Kafka) Retrieved 08 10, 2019, from https://kafka.apache.org/

Malware domain list. (2020, 01 28). Retrieved 08 10, 2019, from http://www.malwaredomainlist.com

Minkevics, V., & Kampars, J. (2018). IS Security Governance Capability Design for Higher Education Organization. *2018 59th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, (pp. 1-5).

Mowbray, M., & Hagen, J. (2014). Finding Domain-Generation Algorithms by Looking at Length Distribution. *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, (pp. 395-400).

Mozilla.org. (2020, 01 07). *About DNS-over-HTTPS*. (Mozilla.org) Retrieved 01 07, 2020, from https://support.mozilla.org/en-US/kb/firefox-dns-over-https#w_about-dns-over-https

Nmap. (2020, 01 28). (NMAP.ORG) Retrieved 08 20, 2019, from https://nmap.org

PCI DSS: Lessons to learn from recent payment card breaches. (2018, 09 14). (itgovernance) Retrieved 01 31, 2020, from https://www.itgovernance.co.uk/blog/pci-dss-lessons-to-learn-from-recent-payment-card-breaches

Peck, J., Nie, C., Sivaguru, R., Grumer, C., Olumofin, F., Yu, B., . . . Cock, M. (2019). CharBot: A Simple and Effective Method for Evading DGA Classifiers. *IEEE Access, 7*, 91759-91771.

Quad9. (2020, 01 28). *DoH with Quad9 DNS Servers*. Retrieved from https://www.quad9.net/doh-quad9-dns-servers/

Sandkuhl, K., & Stirna, J. (2018). *Capability Management in Digital Enterprises.* Springer International Publishing.

Suricata. (2020, 01 28). *Suricata*. Retrieved 01 22, 2020, from https://suricata-ids.org/

Suricata IDS . (2020, 01 28). (The Open Information Security Foundation.) Retrieved 08 20, 2019, from https://suricata-ids.org/

Tenable. (2020, 01 28). (Tenable) Retrieved 08 10, 2019, from https://www.tenable.com/products/nessus/nessus-professional

The 18 biggest data breaches of the 21st century. (2018, 12 20). (CSO) Retrieved 01 31, 2020, from https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html

Truong, D.-T., & Cheng, G. (2016, 9 25). Detecting domain-flux botnet based on DNS traffic features in managed network. *Security and Communication Networks, 9*(14), 2338-2347.

Trustwave. (2019). *2019 Trustwave Global Security Report*. (Trustwave) Retrieved 08 20, 2019, from https://www.trustwave.com/en-us/resources/library/documents/2019-trustwave-global-security-report/

Z. Hu,L. Zhu,J. Heidemann,A. Mankin,D. Wessels, P. (2016). Specification for DNS over Transport Layer Security (TLS). 18. IETF Tools.