DOI: 10.15514/ISPRAS-2019-31(5)-15

Machine Learning Use Cases in Cybersecurity

[1] *S.M. Avdoshin, ORCID: 0000-0001-8473-8077 <savdoshin@hse.ru>*
[2] *A.V. Lazarenko, ORCID: 0000-0001-5812-0134 <lazarenko@group-ib.com>*
[2] *N.I. Chichileva, ORCID: 0000-0002-3012-8043 <chichileva@group-ib.com>*
[2] *P. A. Naumov, ORCID: 0000-0002-9323-9074 <naumov@group-ib.com>*
[3] *P. G. Klyucharev, ORCID: 0000-0001-9536-8083<pk.iu8@yandex.ru>*

[1] *National Research University Higher School of Economics, 20,*
Myasnitskaya st., Moscow, 101000 Russia
[2] *Group-IB,*
1, Sharikopodshipnikovskaya Ulitsa, Moscow, 115080 Russia
[3] *Bauman Moscow State Technical University,*
5/1, 2nd Baumanskaya Ulitsa, Moscow, 105005 Russia

Abstract. The problem regarding the use of machine learning in cybersecurity is difficult to solve because the advances in the field offer many opportunities that it is challenging to find exceptional and beneficial use cases for implementation and decision making. Moreover, such technologies can be used by intruders to attack computer systems. The goal of this paper to explore machine learning usage in cybersecurity and cyberattack and provide a model of machine learning-powered attack.

Keywords: cyberattack; cybersecurity; deep learning; machine learning; machine learning-powered cyberattack

Для цитирования: Avdoshin S.M., Lazarenko A.B., Chichileva N.И., Naumov P.A., Klyucharev P.Г. Machine Learning Use Cases in Cybersecurity. Trudy ISP RAN/Proc. ISP RAS, vol. 31, issue 5, 2019, pp. 191-202. DOI: 10.15514/ISPRAS-2019-31(5)-15

Примеры использования машинного обучения в кибербезопасности

[1] *С.М. Авдошин, ORCID: 0000-0001-8473-8077 <savdoshin@hse.ru>*
[2] *А.В. Лазаренко, ORCID: 0000-0001-5812-0134 <lazarenko@group-ib.com>*
[2] *Н.И. Чичилева, ORCID: 0000-0002-3012-8043 <chichileva@group-ib.com>*
[2] *П.А. Наумов, ORCID: 0000-0002-9323-9074 <naumov@group-ib.com>*
[3] *П.Г. Ключарев, ORCID: 0000-0001-9536-8083<pk.iu8@yandex.ru>*

[1] *НИУ Высшая школа экономики,*
101978, Россия, г. Москва, ул. Мясницкая, д. 20
[2] *Group-IB,*
115080, Россия, г. Москва, ул. Шарикоподшипниковская, д. 1
[3] *Московский государственный технический университет им. Н.Э.Баумана,*
105005, г. Москва, 2-я Бауманская ул., д. 5, стр. 1

Аннотация. Проблему использования машинного обучения в кибербезопасности трудно решить, поскольку достижения в этой области открывают так много возможностей, что сложно найти действительно хорошие варианты решения реализации и принятия решений. Более того эти технологии также могут использоваться злоумышленниками для кибератаки. Цель этой статьи – сделать обзор актуальных технологий в кибербезопасности и кибератаках, использующих машинное обучение, и представить модель атаки на основе машинного обучения.

Ключевые слова: кибератака; кибербезопасность; глубокое обучение; машинное обучение; кибератака с машинным обучением

Для цитирования: Авдошин С.М., Лазаренко А.В., Чичилева Н.И., Наумов П.А., Ключарев П.Г. Примеры использования машинного обучения в кибербезопасности. Труды ИСП РАН, том 31, вып. 5, 2019 г., стр. 191-202 (на английском языке). DOI: 10.15514/ISPRAS-2019-31(5)-15

1. Introduction

Cybersecurity is gaining more and more attention each Cybersecurity is gaining more and more attention each year. The number of cyberattacks has significantly increased since 2009 due to the digitalization of everything in the modern world. According to the Gartner Hype Cycle [1], machine learning (ML) is of great interest in the world of technology. ML is concerned with intelligent behaviour in a system, including perception, reasoning, learning, communication and acting in a complex environment [2]. Such widespread interest in ML is due to two critical factors: First, it can automate processes that previously required human participation, for example, control of robotic mechanisms in production (i.e. ML assumes human responsibilities). Second, it can quickly process and analyze huge amounts of information and calculate options using many variables. In these areas, ML provides qualitatively better results compared to humans.

ML has much to offer cybersecurity. Current implementations are widely used in IDS systems, sandbox systems and many different areas of cybersecurity – from threat intelligence data collection to advanced automated digital forensics. In fact, 71% of US businesses plan to use ML in their cybersecurity tools in 2019 [3] as over one-third (36%) [3] of organizations experienced damaging cyberattacks in 2018. The majority (83%) [3] confides that cybercriminals use ML to attack organizations. The problem of ML use in cybersecurity is difficult to solve because the advances in the field offer so many opportunities that it is hard to find good and beneficial use cases for implementation and decision making. Moreover, it is difficult to determine how secure a security system is, which is used in production, and how to protect the organization from cyberattacks conducted through ML. The main goal of the current work explore ML usage in cybersecurity and research use cases related to the adversary's use of ML in cyberattacks.

2. Basic definitions

ML is the process by which machines learn from given data, building the logic and predicting output for a given input [4]. ML has three sub-categories: supervised learning, unsupervised learning and reinforcement learning [5]. Supervised learning uses a dataset labelled with the correct answers for study. Such labels identify the characteristics of each dataset. Once the model is trained, it can start predicting or deciding on new data that is given to it. In unsupervised learning, there is no need for such a marked set of data. Once the model is given the dataset, it automatically finds patterns and relationships by creating clusters in it. However, such type of learning cannot predict anything. When new data is added, the model assigns them to one of the existing clusters or creates a new one. Reinforcement learning is the ability of a system to interact with the environment and identify the best outcome. The system is either rewarded or penalized with a point for a correct or a wrong answer, and based on positive reward points gained, the model trains itself. Similarly, once trained, it prepares to predict new data presented to it.

Deep learning (DL) is a class of ML algorithms [6] that uses multiple layers to progressively extract higher-level features from the raw input. The main differences between ML and DL are as follows: ML algorithms almost always require structured data, whereas DL networks rely on layers of the Artificial Neural Networks (ANNs). Often in ML, human intervention is necessary to produce further outputs with more sets of data, while in DL, this is not necessary. One of the central concepts in DL is ANNs. The ANN is a model that is built on the principle of organization and the functioning of the human brain (i.e. networks of nerve cells in a living organism). In other words, a neural network algorithm tries to create a function to map one's input to one's desired output. Neural networks (NNs) are typically organized in layers (fig. 1). Layers consist of a number of interconnected 'nodes' that contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is the output.

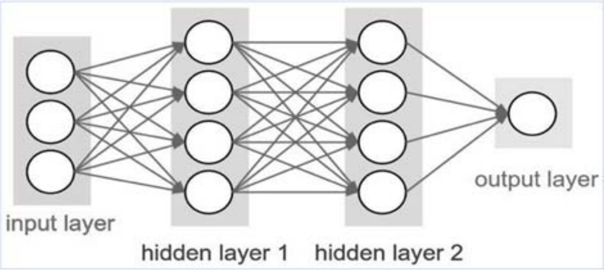

Fig. 1. Neural networks

For example, in image processing, lower layers may identify edges, while higher layers may identify concepts relevant to a human, such as digits, letters or faces. If NNs have more than two hidden layers, they are called deep neural networks (DNNs) [7]. DNN is used for image recognition, speech recognition and other applications. Moreover, technologies have been created to generate new photographs that look at least superficially authentic to human observers through many realistic characteristics. For example, there is a known attempt to synthesize photographs of cats that has misled an expert to think they are real ones [8]. This is an example of the technology called generative adversarial network (GAN), an ML algorithm of unsupervised learning built on a combination of two NNs: one network G (generator) generates new examples and one network D (discriminator) tries to classify examples as either real or fake (generated) [9].

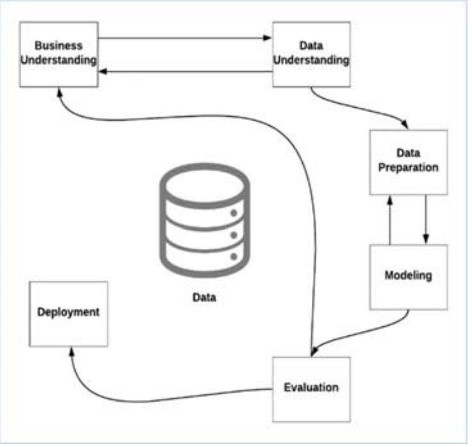

Fig. 2. CRISP-DM process of data mining

One of the processes that is inextricably linked with ML and DL is data mining. Using data mining in large datasets can identify new patterns by utilizing statistics and database systems methods [10]. The Cross-Industry Standard Process for Data Mining (CRISP-DM) describes the cross-industry process for data mining [11]. CRISP-DM breaks the process into six main phases: business understanding, data understanding, data preparation, modelling, evaluation and deployment (fig. 2). The first two phases are connected to each other. Their main aim is to determine the goals of the project, set the task for ML and collect data. These aims can be adjusted based on the data. The next phase refers to the process of working with data: cleaning the data, combining the data, if necessary, and formatting the data.

In the modelling phase, various modelling techniques are applied to the data. Models are built, and their parameters are adjusted to optimal values. Because of special data requirements in different models, we can return to the Data Preparation phase. In the evaluation phase, the model has already been built, and quantitative assessments of its quality have been obtained. Before implementing this model, we need to make sure that we have achieved all business goals. Depending on the requirements, the deployment phase may be simple (e.g. preparation of the final report) or complex (e.g. automation of the data analysis process to solve business problems).

3. Using ML for protection

The scope of ML usage in cybersecurity is huge, starting with identifying anomalies and suspicious or unusual behaviours and ending with detecting zero-day vulnerabilities and patching known ones. Dilek et al. [12] presented the most comprehensive review of applications of ML techniques.

Reathi and Malathi [13] presented a set of ML algorithms trained on the NSL-KDD intrusion detection dataset for misuse detection. Meanwhile, Buczak et al. [14] focused on network intrusion detection using ML.

Melicher et al. [15] proposed using NNs to check password guessing resistance. They compressed the model to hundreds of kilobytes and developed a client-side JavaScript tool. The similar experiment was conducted by Ciaramella et al. [16]. To proactively check the strength of passwords, they use NNs, such as Multilayer Perceptron (MLP) and Single Layer Perceptrons (SLPs). Notably, MLPs provide better results than SLPs when testing datasets. Moreover, the number of layers equal to 10, and thus obtains better result.

User and entity behaviour analytics (UEBA) use ML capabilities to analyze behaviour logs and network traffic in real-time and respond appropriately in the event of an attack [17]. This process is done by getting the user to log in again, blocking an attack or assessing risk levels and alerting the company's information security officers so that they can take necessary action.

Most of the ML and DL methods, such as ensemble learning, clustering, and decision tree, [18] are used to detect misuse, anomaly and hybrid cyber intrusion.

As mentioned in the Eugene Kaspersky Official Blog [19], Kaspersky detects 99% of cyber threats using ML technology. The time interval between the disclosure of suspicious behaviour on the protected device and the release of the corresponding new 'tablet' lasts an average of 10 minutes.

DARPA collaborated with BAE Systems to develop a system that allow us to configure sensors and apply protective measures 'at machine speed'. This initiative called the CHASE program, which stands for Cyber Hunting at Scale, seeks to develop automated tools to detect and characterize novel attack vectors, collect the right contextual data, and disseminate protective measures both within and across enterprises [20].

Cyberattacks performed by hacktivists relate to a common opinion about high-profile news. Information gathered from social media can help predict such incidents using NLP and ML techniques [21].

Moreover, we can use ML to identify the author of the program. Rachel Greenstadt and Aylin Caliskan developed a system that can 'deanonymize' programmers [22] by analyzing source code or compiled binary files [23]. Identifying the developer of malware is now much easier.

Another way to monitor systems and networks for malicious activity or policy violation is through the intrusion detection system (IDS). Intrusion prevention system (IPS) is a system connected with IDS; these systems perform intrusion detection and stop the detected incidents. Both systems use supervised and unsupervised ML techniques to detecting point anomaly, contextual anomaly, and collective anomaly [24].

The main task of firewalls [25] is to ensure a network security system that monitors and controls incoming and outgoing network traffic. Firewalls allow or block traffic by comparing its characteristics with predefined patterns (i.e. firewall rules). In their paper, Ucar and Ozhan [26] presented the result of the automatic detection of anomalies in firewall rule repository based on ML and high-performance computing methods, such as Naive Bayes, kNN, Decision Table and HyperPipes. All six firewall rules from the given 93 rules were detected by the system and verified by the experts as an anomaly. Firewalls filter the content between servers, and there is also a solution specifically meant for the content of web applications. Web application firewall (WAF) is deployed in front of web applications; it analyzes bi-directional web-based (HTTP) traffic and detects and blocks anything malicious [27]. WAF prevents vulnerabilities in web applications from being exploited by outside threats. To implement such functionality in WAF, developers use regular expressions, tokens, behavioural analysis, reputation analysis and ML technologies [27].

Among ML methods, special predictive ones can also be used for data loss/leak prevention (DLP) to reduce the risk for breaches or leaks [28]. DLP software solutions allows us to set business rules that classify confidential and sensitive information so that they cannot be disclosed maliciously or accidentally by unauthorized end users. This process can be done by using supervised learning algorithms and two types of examples: positive examples (i.e. content that needs to be protected) and counterexamples (i.e. documents that are similar to the positive set but should not be protected).

4. Using ML in cyberattacks

This section describes how cyberattack can succeed using ML. Automated vulnerability scanning is one of the most obvious and common tasks in a cyberattack. For example, CSRF is found in only 5% of applications, as reported in the 2017 OWASP Top 10, because most frameworks include CSRF defences [29]. Accordingly, Calzavara et al. presented Mitch [30], the first ML-based tool for the black-box detection of CSRF, which allows the identification of 35 new CSRF vulnerabilities on 20 websites from the Alexa Top 10,000 websites and three previously undetected CSRF vulnerabilities on production software already analyzed with the state-of-the-art tool Deemon [31]. Mitch is a binary classifier, labelling sensitive or insensitive requests using a random forest algorithm on a 49-dimensional feature space. Compared to the heuristic classifiers BEAP [32] and CsFire [33], Mitch shows the best F1-score and precision (Table 1).

Marketers use ML methods for profiling. Trustwave released an open source intelligence tool that uses face recognition to automatically track subjects across social media networks [34]. Facial recognition aids this process by removing false positives in the search results, making data review faster for a human operator.

Table 1. Validity measures for the tested classifiers (BEAP, CsFire, Mitch)

Classifier	Precision	Recall	F1
BEAP	0.30	0.89	0.45
CsFire	0.20	0.97	0.33
Mitch	0.78	0.67	0.72

Using collected data about the target, an attacker can hook a victim with specially created fake news. ML tools can help identify fake news, but to do so, researchers confirm that the best way is for that ML to learn to create fake news itself [35]. As such, they created a model for controllable text generation called Grover. In the research process, four classes of articles were used: human news, machine news, human propaganda and machine propaganda. Workers on Amazon Mechanical Turk rated each article, including overall trustworthiness. In the case of propaganda, the score increased from 2.19 (out of 3) on articles created manually to 2.42 on articles created by a machine.

SNAP_R was introduced at DEFCON 24. SNAP_R is the world's first automated end-to-end spear-phishing campaign generator for Twitter [36]. While previous tools were based on models with Markov chains, SNAP_R is based on a recurrent NN with LSTM architecture. Using Twitter as an environment offers some advantages for automatically generating text. For example, limiting the length of a post decreases the probability of grammatical errors. Moreover, Twitter links are often shortened, which allows masking of malicious domains. This, in turn, significantly increased the success rate from 5–14% on Markov chain-based tools [37, 38] to 30–66%, which is comparable to the 45% rate for manual spear-phishing [39].

In most cases, attackers do not know the malware detection algorithm but can figure out features it uses through carefully designed test cases in the black-box algorithm. MalGAN is a generative adversarial network-based algorithm that generates adversarial malware examples that are able to bypass black-box ML-based detection models. It can decrease the detection rate to nearly zero and make it hard for the retraining-based defensive method against adversarial examples to work [40]. The architecture of MalGAN is shown in fig. 3 [40].

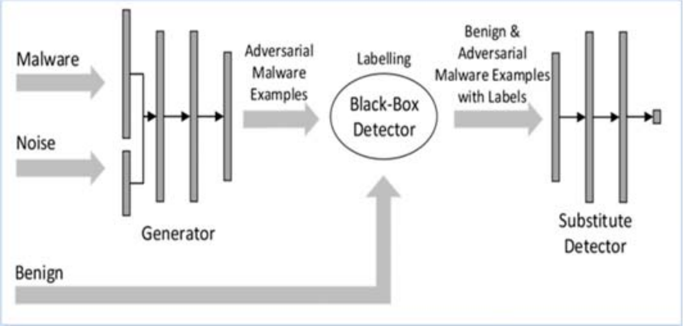
Malware
Noise
Benign
Generator
Adversarial Malware Examples
Labelling
Black-Box Detector
Benign & Adversarial Malware Examples with Labels
Substitute Detector

Figure 3. Architecture of MalGAN

The generator takes the malware feature vector and the noise vector to transform the former into its adversarial version. Substitute detector is used to fit the black-box detector and provide gradient information to train the generator. Both nets are represented as multi-layer feed-forward ANNs. Adversarial examples tested against the black-box detector according to different ML methods trained on 160-dimensional binary feature vectors representing system API calls include random forest, logistic regression, decision trees, support vector machines, and multi-layer perceptron as well as a voting-based ensemble of these algorithms. All these classifiers detect over 90% of original samples, but random forest and decision trees show the best result of less than 0.20% on adversarial examples. Anti-malware vendors retrain detectors after exploring such undetected examples, but MalGAN only needs one epoch retraining to obtain a 0% true positive rate. Kawai et al. later proposed some performance improvements [41].

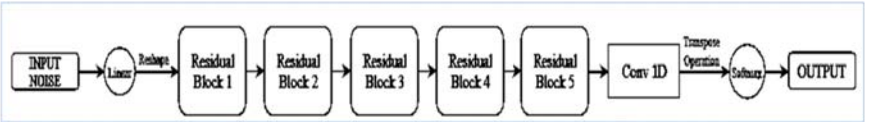

Fig. 4. Generator architecture of PassGAN

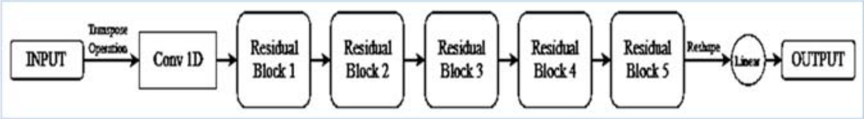

Fig. 5. Discriminator architecture of PassGAN

Another example use case for GAN in cybersecurity is the password guessing attack. There is a new way of generating password guesses based on DL and generative adversarial networks known as PassGAN [42]. The key difference in this approach is that NNs do not need a priori knowledge of the structure of passwords, in contrast to approaches based on rules, Markov models [43] and FLA [15]. PassGAN uses the improved training of Wasserstein GANs (IWGAN) of Gulrajani et al. [44] with the ADAM optimizer [45]. The generator and the discriminator in PassGAN are built from ResNets [46]. The architecture of the generator and the discriminator are shown in fig. 4 and fig. 5 [42], while residual block representation is shown in fig. 6 [42].

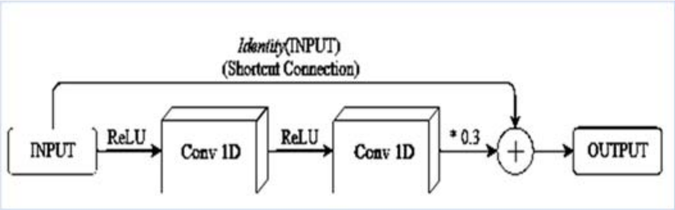

Fig. 6. Architecture of residual block in PassGAN

For maximum effectiveness, attackers most likely use several password-cracking tools, such as a HashCat [47], John the Ripper [48], PCFGs [49], OMEN [50] and FLA [15] to combine different attack methods. For example, by combining the output of PassGAN with the output of HashCat Best64 [51], researchers were able to guess between 51% and 73% additional unique passwords compared to HashCat [47] alone.

Traditional botnets wait for commands from the C&C, but now, attackers use automation to make decisions independently. Fortinet researchers predicted that cybercriminals will replace botnets with intelligent clusters of compromised devices called hivenets, a type of attack that is able to leverage peer-based self-learning to target vulnerable systems with minimal supervision [52].

In the initial stages of an attack, attackers often face the challenge of bypassing captcha. Suphannee et al. [53] designed a low-cost attack that uses DL technologies for the semantic annotations of images. The system requires about 19 seconds per challenge to solve challenges, with an accuracy of 70.78% for reCaptcha [54] and 83.5% for the Facebook image captcha. The system has to automatically identify which of the given images are semantically similar to the sample image. First, the system collects information for all the images through Google Reverse Images Search (GRIS) [55]; Clarifai [56], which is built on deconvolutional networks [57]; TDL [58], which is based on deep Boltzmann machines [59]; NeuralTalk [60] and Caffe [61]. Next, if a hint is not provided, the system searches for the sample image in the labelled dataset to obtain one, if possible.

5. Fully ML-powered cyberattack

As mentioned in the previous section, ML-powered cyberattacks are not a hypothetical future concept. This section describes how an automated cyberattack can be carried out using ML.

We considered two scenarios for the weaponization and delivery stages: First, in the case of humanless intrusion, attackers can use a similar tool but utilize information provided by Shodan [62] or Mitch [30] instead of features obtained using a computer vision. Second, attackers can use social engineering, using tools for profiling and for spear-phishing described in the previous section [34, 35] and creating click-bytes links to infect the victim [35, 36]. For automated exploit generation, adversaries can use open-sourced angr [63] framework developed by Shellphish and combine it with MalGAN to bypass defensive systems.

In the post-exploitation stage, attackers can guess stolen passwords using PassGAN [42]. The newest method is using intelligent evasion techniques proposed by Darktrace researchers [64] and further self-propagating with a series of autonomous decisions. It is also possible to turn infected systems into a hivenet [52].

As these examples demonstrate, ML can help hackers in every stage of the attack. With the advance level of development of the cybercriminal infrastructure, an advanced attack requires no hands-on-keyboard such as the case at present.

6. Conclusion

When introducing an ML-based system, we should remember that ML is not a panacea. No system is safe. Under certain conditions, ML both protects vulnerabilities and creates new gaps. ML can be compared to a dog: 'Machine learning can do anything you could train a dog to do – but you're never totally sure what you trained the dog to do'.

We should also note the consequences that more active implementation of ML can bring: First, automation and the resulting loss of human jobs and second, inevitable conflict with the existing legal framework, for example, when using technologies to prevent cybercrime or cyberterrorism. In such a situation, the accused is implicated for crimes that have not yet been committed, which are not regulated by any legal norm. Moreover, some of the information learned by ML may be private or confidential, which violates laws in some countries. Similarly, poor quality or inadequate quantity of ML in the cybersecurity of data on predictions are based can lead to wrong decisions and irreparable mistakes.

References

[1] P. Krensky, J. Hare. Hype Cycle for Data Science and Machine Learning, 2018. Gartner, 2018. Accessed: Sep. 10, 2019. [Online] Available at: https://www.gartner.com/en/documents/3883664/hype-cycle-for-data-science-and-machine-learning-2018.

[2] Nils J. Nilsson. Artificial Intelligence: A New Synthesis. Elsevier Inc, 1998, 513 p.

[3] Businesses recognize the need for AI & ML tools in cybersecurity. Helpnetsecurity.com. Accessed: Sep. 10, 2019. [Online] Available at: https://www.helpnetsecurity.com/2019/03/14/ai-ml-tools-cybersecurity/.

[4] T. Mitchell. Machine Learning. A Guide to Current Research. Tom M. Mitchell, Jaime G. Carbonell, Ryszard S. Michalski (Eds.). Springer Science & Business Media, 1986, 429 p.

[5] J. Grus. Data Science from Scratch: First Principles with Python. O'Reilly Media, 2015, 330 p.

[6] L. Deng, D. Yu. Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing, vol. 7, nos. 3–4, 2014, pp. 199- 200

[7] K. Warr. Strengthening Deep Neural Networks: Making AI Less Susceptible to Adversarial Trickery. O'Reilly Media, Inc., 2019, 246 p.

[8] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen. Improved Techniques for Training GANs. arXiv:1606.03498, 2016.

[9] Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S.Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks. arXiv:1406.2661, 2014.

[10] J. Han, J. Pei, M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 3rd edition, 2011, 744 p.

[11] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth. CRISPDM 1.0 step-by-step data mining guide. SPSS, 2000, 78 p.

[12] S. Dilek, H. Çakır, M. Aydın. Applications Of Artificial Intelligence Techniques To Combating Cyber Crimes: A Review. International Journal of Artificial Intelligence & Applications (IJAIA), vol. 6, vo. 1, 2015, pp. 21-39.

[13] S. Revathi and A. Malathi. A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. International Journal of Engineering Research and Technology, vol. 2, issue 12, 2013, pp. 1848-1853.

[14] L. Buczak and E. Guven. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, vol. 18, no. 2, 2016, pp. 1153–1176.

[15] W. Melicher, B. Ur, S.Segreti, S. Komanduri, L. Bauer, N. Christin, L. Cranor. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In Proc. of the 25th USENIX Security Symposium, 2016, pp. 176-191.

[16] Ciaramella, P. D’Arco, A. De Santis, C. Galdi, R. Tagliaferri. Neural Network Techniques for Proactive Password Checking. IEEE Transactions on Dependable and Secure Computing, vol. 3, no. 4, 2006, pp. 327-339.

[17] Chris Brook. What is User and Entity Behavior Analytics? A Definition of UEBA, Benefits, How It Works, and More. Accessed: Oct. 10, 2019. [Online]. Available at: https://digitalguardian.com/blog/what-user-and-entity-behavior-analytics-definition-ueba-benefits-how-it-works-and-more

[18] Anna L. Buczak, Erhan Guven. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, vol. 18, no. 2, 2016, pp. 1153-1176.

[19] E. Kaspersky. Laziness, Cybersecurity, and Machine Learning. Accessed: Oct. 10, 2019. [Online]. Available: https://eugene.kaspersky.com/2016/09/26/laziness-cybersecurity-and-machine-learning/.

[20] J. Roberts. Cyber-Hunting at Scale (CHASE). Accessed: Oct. 19, 2019. [Online]. Available: https://www.darpa.mil/program/cyber-hunting-at-scale.

[21] Hernandez-Suarez, G. Sanchez-Perez, K. Toscano-Medina, V. Martinez-Hernandez, H. Perez-Meana, J. Olivares-Mercado, V. Sanchez. Social Sentiment Sensor in Twitter for Predicting Cyber-Attacks Using ℓ1 Regularization. Sensors, vol. 18, no. 5, 2018, pp. 1380.

[22] Caliskan, F. Yamaguchi, E. Dauber, R. Harang, K. Rieck, R. Greenstadt, A/ Narayanan. De-anonymizing Programmers via Code Stylometry. In Proc. of the 24th USENIX Security Symposium, 2015, pp. 255-270.

[23] Caliskan, F. Yamaguchi, E. Dauber, R. Harang, K. Rieck, R. Greenstadt, A. Narayanan. When Coding Style Survives Compilation: De-anonymizing Programmers from Executable Binaries. arXiv:1512.08546, 2015.

[24] S. Repalle, V. Kolluru. Intrusion Detection System using AI and Machine Learning Algorithm. International Research Journal of Engineering and Technology (IRJET), vol. 04, issue 12, 2017, pp. 1709-1715.

[25] J. Vacca, S. Ellis. Firewalls: Jumpstart for Network and Systems Administrators. Digital Press, 2004, 448 p.

[26] E. Ucar, E. Ozhan. The Analysis of Firewall Policy Through Machine Learning and Data Mining. Wireless Personal Communications, vol. 96, issue 2, 2017, pp. 2891 - 2909.

[27] S. Prandl, M. Lazarescu, D. Pham. A Study of Web Application Firewall Solutions. Lecture Notes in Computer Science, vol. 9478, 2015, pp. 501-510.

[28] Introduction to Forcepoint DLP Machine Learning. Accessed: Oct. 10, 2019. [Online]. Available at: https://www.websense.com/content/support/library/data/v84/machine_learning/machine_learning.pdf

[29] OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks. Accessed: Nov. 5, 2019. [Online]. Available at: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

[30] S. Calzavara, M. Conti, R. Focardi, A. Rabitti, G. Tolomei. Mitch: A Machine Learning Approach to the Black-Box Detection of CSRF Vulnerabilities. In Proc. of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P), 2019, pp. 528-543.

[31] G. Pellegrino, M. Johns, S. Koch, M. Backes, C. Rossow. Deemon: Detecting CSRF with Dynamic Analysis and Property Graphs. arXiv:1708.08786, 2017.

[32] Z. Mao, N. Li, I. Molloy. Defeating Cross-Site Request Forgery Attacks with Browser-Enforced Authenticity Protection. Lecture Notes in Computer Science, vol. 5628, 2009, pp. 238-255.

[33] Philippe De Ryck, Lieven Desmet, Thomas Heyman, Frank Piessens. CsFire: Transparent Client-Side Mitigation of Malicious Cross-Domain Requests. In Proc. of the Second International Symposium on Engineering Secure Software and Systems, 2010, pp. 18-34.

[34] Jacob Wilkin. Mapping Social Media with Facial Recognition: A New Tool for Penetration Testers and Red Teamers. Accessed: Oct. 19, 2019. [Online]. Available at: https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/mapping-social-media-with-facial-recognition-a-new-tool-for-penetration-testers-and-red-teamers/.

[35] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi. Defending Against Neural Fake News. arXiv:1905.12616, 2019.

[36] J. Seymour, P. Tully. Weaponizing data science for social engineering: Automated E2E spear phishing on Twitter. Accessed: Oct. 19, 2019. [Online]. Available at: https://www.blackhat.com/docs/us-16/materials/us-16-Seymour-Tully-Weaponizing-Data-Science-For-Social-Engineering-Automated-E2E-Spear-Phishing-On-Twitter-wp.pdf

[37] S. Thompson. Phight Phraud. Accessed: Nov. 6, 2019. [Online]. Available at: https://www.journalofaccountancy.com/issues/2006/feb/phightphraud.html

[38] M. Jakobsson, J. Ratkiewicz. Designing ethical phishing experiments: a study of (ROT13) rOnl query features. In Proc. of the 15th International Conference on World Wide Web, 2006, pp. 513-522.

[39] E. Bursztein, B. Benko, D. Margolis, T. Pietraszek, A. Archer, A. Aquino, A. Pitsillidis, S. Savage. Handcrafted fraud and extortion: Manual account hijacking in the wild. In Proc. of the 2014 Conference on Internet Measurement, 2014, pp. 347-358.

[40] W. Hu, Y. Tan. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. arXiv:1702.05983, 2017.

[41] M. Kawai, K. Ota, M. Dong. Improved MalGAN: Avoiding Malware Detector by Leaning Cleanware Features. In Proc. of the 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2019, pp. 40 - 45.

[42] Hitaj, P. Gasti, G. Ateniese, F. Perez-Cruz. PassGAN: A Deep Learning Approach for Password Guessing. arXiv:1709.00440, 2017.

[43] Narayanan, V. Shmatikov. Fast dictionary attacks on passwords using time-space tradeoff. In Proc. of the 12th ACM Conference on Computer and Communications Security, 2005, pp. 364 - 372.

[44] Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville. Improved training of wasserstein GANs. In Proc. of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 5769-5779.

[45] Kingma, J. Ba. Adam: A Method for Stochastic Optimization. arXiv:1412.6980, 2017.

[46] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. arXiv:1512.03385, 2015.

[47] Hashcat – advanced password recovery. Accessed: Oct. 19, 2019. [Online]. Available at: https://hashcat.net/hashcat/

[48] John the Ripper password cracker. Accessed: Oct. 19, 2019. [Online]. Available at: https://www.openwall.com/john/

[49] M. Weir, S. Aggarwal, B. Medeiros, BGlodek. Password cracking using probabilistic context-free grammars. In Proc. of the 30th IEEE Symposium on Security and Privacy, 2009, pp. 391-405.

[50] M. Dürmuth, F. Angelstorf, C. Castelluccia, D. Perito, C. Abdelber. OMEN: Faster Password Guessing Using an Ordered Markov Enumerator. Lecture Notes in Computer Science, vol. 8978, 2015, pp. 119-132.

[51] hashcat/rules/best64.rule. Accessed: Nov. 10, 2019 [Online]. Available at: https://github.com/hashcat/hashcat/blob/master/rules/best64.rule.

[52] Derek Manky. Fortinet Predicts Highly Destructive and Self-learning “Swarm” Cyberattacks in 2018. Accessed: Nov. 10, 2019 [Online]. Available at: https://www.fortinet.com/corporate/about-us/newsroom/press-releases/2017/predicts-self-learning-swarm-cyberattacks-2018.html.

[53] S. Sivakorn, J. Polakis, A.Keromytis. I'm not a human: Breaking the Google reCAPTCHA. Accessed: Nov. 10, 2019 [Online]. Available at: https://www.blackhat.com/docs/asia-16/materials/asia-16-Sivakorn-Im-Not-a-Human-Breaking-the-Google-reCAPTCHA-wp.pdf.

[54] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-based character recognition via web security measures. Science, vol. 321, no. 5895, 2008, pp. 1465-1468.

[55] A. Krizhevsky, I. Sutskever, G. Hinton. ImageNet classification with deep convolutional neural networks. Communications of the ACM, June 2017, vol. 60, issue 6, pp. 84-90.

[56] Clarifia. Accessed: Nov. 10, 2019 [Online]. Available at: https://www.clarifai.com.

[57] M. Zeiler, G. Taylor, Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In Proc. of the International Conference on Computer Vision, 2011, pp. 2018-2025.

[58] Toronto Deep Learning Demos, Accessed: Nov. 10, 2019 [Online]. Available at: http://deeplearning.cs.toronto.edu.

[59] N. Srivastava, R. Salakhutdinov. Multimodal Learning with Deep Boltzmann Machines. Journal of Machine Learning Research, vol. 15, 2014, pp. 2949-2980

[60] Andrej Karpathy. Deep Visual-Semantic Alignments for Generating Image Descriptions. Accessed: Nov. 10, 2019 [Online]. Available at: https://cs.stanford.edu/people/karpathy/cvpr2015.pdf.

[61] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv:1408.5093, 2014.

[62] Shodan. Accessed: Oct. 19, 2019 [Online]. Available at: https://www.shodan.io/.

[63] Angr. Accessed: Oct. 19, 2019 [Online]. Available at: https://angr.io/.

[64] The Next Paradigm Shift AI-Driven Cyber-Attacks. Accessed: Oct. 19, 2019 [Online]. Available at: https://www.darktrace.com/en/resources/wp-ai-driven-cyber-attacks.pdf.

Information about authors / Информация об авторах

Sergey Mikchailovitch AVDOSHIN – Candidate of Technical Science, Professor, Head of the School of Software Engineering at National Research University Higher School of Economics since 2005. Research interests include design and analysis of computer algorithms, simulation and modeling, parallel and distributed processing, deep Web, blockchain technology.

Сергей Михайлович АВДОШИН – кандидат технических наук, профессор, руководитель департамента программной инженерии факультета компьютерных наук НИУ ВШЭ с 2005 года. Сфера научных интересов: разработка и анализ компьютерных алгоритмов, имитация и моделирование, параллельные и распределенные процессы, теневой интернет, технология блокчейн.

Aleksandr LAZARENKO – head of R&D department of Group-IB, cybercrime investigator from 2015, author of scientific papers on the privacy, anonymity, security of blockchain projects.

Александр Вячеславович ЛАЗАРЕНКО – руководитель департамента инноваций и разработки продуктов Group-IB, расследует киберпреступления с 2015 года, автор научных работ по вопросам конфиденциальности, анонимности, безопасности блокчейн-проектов.

Nataliia Igorevna CHICHILEVA – junior specialist of R&D department Group-IB, student of System and Software Engineering master's programme at HSE. Her research interests include information security, discrete mathematics, optimizations problem and travelling salesman problem.

Наталия Игоревна ЧИЧИЛЕВА – младший специалист департамента инноваций и разработки продуктов Group-IB. В настоящее время также является студенткой магистерской программы «Системная и программная инженерия». Профессиональные интересы – информационная безопасность, дискретная математика, задача оптимизации, задача коммивояжера.

Pavel Andreevich NAUMOV – Junior Specialist in the Department of Innovation and Product Development Group-IB, a student with a degree in Computer Security at Moscow State Technical University. His professional interests include mathematical methods of information protection, hardware and software reverse engineering, security of development and applications.

Павел Андреевич НАУМОВ – младший специалист департамента инноваций и разработки продуктов Group-IB, студент специалитета по программе «Компьютерная безопасность» в МГТУ им. Н.Э.Баумана. Профессиональными интересами являются математические методы защиты информации, программно-аппаратная обратная разработка, безопасность разработки и приложений.

Petr Georgievich KLYUCHAREV – Candidate of Technical Science, Associate Professor of the Department of Information Security, BMSTU. Research interests: cryptography, discrete mathematics, theoretical computer science, machine learning, software development.

Петр Георгиевич КЛЮЧАРЕВ – кандидат технических наук, доцент кафедры «Информационная безопасность» МГТУ им. Н. Э. Баумана. Сфера научных интересов: криптография, дискретная математика, теоретическая информатика, машинное обучение, разработка программного обеспечения.