A Survey on Detection Methods Against Internet Worm Attacks

Liang Wang*
National Engineering Laboratory for Next Generation Internet Technologies
Beijing Jiaotong University, No.3, Shangyuan Village, Haidian District, Beijing 100044, China

Abstract

Miscellaneous attacks against the Internet have become one of the major concerns among many imperfections in contemporary cyberspace. In particular, worms are one highly dangerous type. How to timely detect and utterly stop such attacks still remains an unresolved issue, and it is of great necessity to review the existing quality work in the field of worm detection. In this paper, major characteristics, working mechanisms, and the life cycle of worms are introduced. Then, we divide existing worm detection techniques into two categories: machine learning methods and traditional detection methods, and existing worm countermeasures are reviewed in detail by category. Finally, some open issues in the field of worm detection are discussed, and the contents of this paper are summarized. The goal of this paper is to let researchers comprehensively understand various existing defense mechanisms against worms.

Keywords: Worm attacks, Intrusion detection, Cyber security

1 Introduction

As the Internet brings people great convenience, many security threats also come along, and they have become a huge obstacle restricting the development of the Internet. In particular, network intrusions are one of the more damaging categories and have drowned particular attention. In 2015, Ukraine's power grid was blacked out due to a malware attack in which hackers managed to hack into the information systems of three Ukrainian energy distribution companies, temporarily disrupting power supplies to end-users. In 2017, some Chinese universities were infected with ransomware that caused severe losses to students and faculty when their computer files were encrypted. Hackers demanded a ransom payment to recover files.

Some scholars have summarized the existing research results in the field of intrusion detection. Jing et al. [13] began by classifying security-related data into four categories: packet-level data, stream-level data, connection-level data, and host-level data, and explained in detail the collection methods and data content of each type of data. For different kinds of attacks, authors respectively summarized the data analysis methods that can be used to detect both attacks and the data categories used in the data analysis process. Li et al. [17] reviewed various detection and interception methods against network worms, compared and analyzed different detection algorithms based on their performance, parameters used, and analyzed various worm defense systems based on their scopes of application and deployment location. Jeong et al. [12] investigated problems and technical solutions for anomaly teletraffic intrusion detection systems based on the open-source software platform Hadoop. Zargar et al. [31] first introduced the classification of DDoS attacks according to the layer to which the protocol belongs (network layer/transport layer/application layer). Then, Botnet, an important tool for launching DDoS attacks as well as defense strategies against DDoS attacks are introduced.

Research on the application of machine learning techniques in the field of intrusion detection has become a hot research topic. Researchers have also summarized existing research on the application of

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 6, Article No. 7 (December 1, 2020)
*Corresponding author: Tel: +86-13910212233, Email: 19120130@bjtu.edu.cn

machine learning technology in the field of network intrusion prevention. Berman et al. [3] reviewed existing deep learning methods for network security applications involving deep autoencoders, restricted Boltzmann machines, recurrent neural networks, and generative adversarial networks. The authors discussed how each deep learning technique is applied to network security and covers a wide range of attack types. Nisioti et al. [16] provided a comprehensive overview of unsupervised and hybrid (supervised and unsupervised methods combined) intrusion detection methods (IDS), associated feature engineering techniques, and discussed the use of IDS for attribution to identify attackers. Nguyen et al. [14] provided an overview of emerging research on the application of machine learning techniques to IP traffic classification and discuss some of the key requirements for applying machine learning techniques to IP traffic classification. Furthermore, in [15], the approaches of applying deep reinforcement learning to cybersecurity are outlined by Nguyen et al.

Generally speaking, network intrusion detection methods include statistics-based methods, knowledge-based methods, and machine learning-based methods. The statistics-based approach first generates a profile that can represent normal network behavior and then uses statistical methods to calculate the difference value between the traffic and the normal profile, and traffic with a difference value exceeding a threshold will be judged as malicious traffic. Statistics-based methods are susceptible to being trained by attackers, it is difficult to set appropriate thresholds, and it is difficult to model the purely normal behavior of the network, therefore, statistics-based methods alone are not applicable to intrusion detection in today's complex network environments. In knowledge-based methods, network behavior would be matched to predefined rules or patterns to check for the presence of known malicious behavior. Such methods are simple, robust, and flexible They have a high detection rate when accurate predefined attack rules can be created, but the drawback of methods like these is that they are not able to detect zero-day attacks. The application of machine learning techniques for intrusion detection is a popular research direction. Machine learning methods can analyze traffic data characteristics to build explicit or implicit models of the data, and these methods have high detection rates and can be retrained and updated for new traffic. However, such methods would consume a lot of computational resources during the training phase and require high-quality data to support the construction of the model. Generally, these three kinds of methods can be broadly categorized into two groups, namely traditional detection methods and machine learning-based methods, statistics-based methods and knowledge-based methods fall into the former class.

None of the previously mentioned surveys have conducted a more comprehensive survey of worm attack detection methods and cover the earlier traditional worm detection methods as well as the newer machine learning detection methods. In this paper, we provide a detailed introduction to both traditional and newer machine learning-based detection methods used in worm attacks, and the advantages and disadvantages of the different methods are compared. Also, the dataset used, performance indicators, and other information of each work are summarized.

The rest of the paper is organized as follows. In section 2, we discuss the significant features and the working mechanism of worm attacks, and the main stages in the life cycle of worms are also talked about. Section 3 introduces in detail the existing worm detection methods according to the two categories mentioned above (traditional detection methods and machine learning-based detection methods). Besides, machine learning-based methods are further divided into two subcategories: traditional machine learning methods and deep learning methods. In section 4, we talk about some open issues in the field of worm detection. Finally, the conclusion of the paper is presented in section 5.

2 Worm Attacks Overview

Malware is a crucial tool used to launch network attacks. Most malware needs to rely on human intervention to spread. However, internet worms can replicate themselves and use vulnerabilities in the operating system to actively spread through communication protocols. Therefore, among many malwares, internet worms are a more harmful category.

As shown in Figure 1, the life cycle of worms roughly includes four stages: target finding, worm propagation, worm activation, and worm infection [17], in which the first two stages, the worm activity mainly contains network behavior, while the last two stages mainly involve local behavior on the target host. In this paper, we mainly focus on the network behavior of worms, so these two stages, target discovery, and propagation are our points of attention.

The first stage in the life cycle of worms is target finding. Different methods can be used to find new victims, and one simple method is blind scanning. Blind scanning can be divided into sequential scanning, random scanning, and ranked scanning, all of which are chance-based and have a high failure rate, but many worms use these types of scanning due to the ease of implementation of blind scanning worms. The second way a worm can find its target is to use a predefined list of addresses, called "Hit list". Hit list scanning worm knows exactly where the target is, and the hit list can be included inside the worm or stored outside for the worm to look up. The greater the number of addresses in the Hit list, the more difficult it is to obtain, but it allows worms to do more damage. The initial propagation speeds and accuracy of hit list scanning worms are much higher than that of blind scanning worms.

After finding new victims, the worm transmits a copy of itself to them for propagation. According to their ways of propagation [27], worms can be generally categorized into three classes: self-carried, second channel, and embedded worm. A self-carried worm transmits itself as the payload of packets and can rely on TCP or UDP protocols for propagating. A second channel worm would firstly find its target and enter, then it uses the installed backdoor to download the worm from the Internet or download it via other infected hosts. The Embedded worm spreads in an extremely stealthy way, attaching worm loads to legitimate traffic for stealthy purposes, without triggering any unusual events.

After a worm enters a host, it would be triggered and activated under certain conditions and then infects the target host with its malicious payload, the worm's malicious code. Early worms tend to spread using unchanging payloads, so signature-based worm detection methods can easily find the advent of these worms. However, through changing the payload, the worm can easily evade signature-based detection methods detection. Some worm authors would split the worm code before propagation and reassemble the code at the target. Some worm authors may dynamically change the payload of the worm so that the worm would present different signatures without changing its function.

3 Countermeasures Against Worm Attacks

Worm detection methods can be roughly divided into signature-based and behavior-based methods. Each type of malware has a unique content signature (special bit strings, hashes, etc.), which can be considered as a fingerprint of the malware. Since the inception of worms, most countermeasure techniques have used signature-based detection as their primary method. Signature-based methods are easy to implement, and their detection performance is largely dependent on the availability of the signature database. As mentioned in Section IV, signature-based detection methods are easy to be evaded. Authors of worms can make their malware generate new content signatures and retain malicious functionality by changing the code, which can be achieved by simple code conversion methods such as inserting junk code, applying code alignment, etc. Also, such methods cannot handle encrypted files and cannot detect new types of malware (threats that are not included in signature repositories cannot be detected). Behavior-based

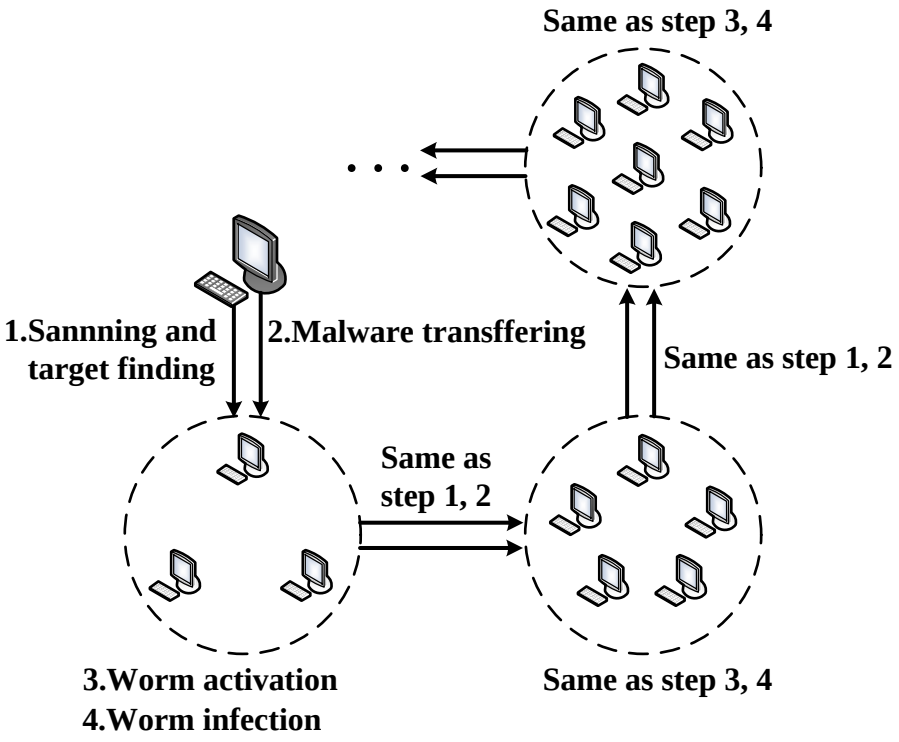

Figure 1: Worm's lifecycle

methods can better solve the above problems. These methods have the ability of detecting abnormal traffic behavior in the network during a worm outbreak and distinguishing the difference in traffic behavior between the worm and normal software. Such methods are more flexible and do not rely simply on the completeness of the signature database, which allows the detection of new types of worms.

Worm detection is an indispensable part of worm defense, but detection techniques alone do not constitute a complete worm defense mechanism. According to the classification mentioned in [18] and [20], worm defense solutions can be divided into Resource limiting solutions, Leap ahead solutions, Predesignated-preventative solutions, Automatic signature generation solutions, and Mobile combat solution, which will be referred to as RL, LA, PP, ASG, and MC respectively. The RL solution's aim is to slow down the worm infection by limiting the use of resources that can be exploited by the worm without significantly affecting normal communications. The idea behind the LA solution is to establish a cooperative information sharing mechanism among different network segments, to notify unaffected segments in advance of the outbreak of worm, and to prevent the spread between segments. PP solutions can dynamically change network or end node connectivity in the presence of worm propagation threats to prevent worms from discovering susceptible nodes in the address space. ASG solutions can filter traffic entering the network and automatically generate file signatures when anomalous activity is detected, and MC solutions are proactive strategies that involve threat interception and rapid remediation. It is a technique that eliminates the spread of malware by distributing a mobile self-replicating code module that detects the presence of malicious code and vaccinates the infected machine.

Worm defense strategies are not the focus of this paper; the rest of this section focuses on existing worm detection methods. We classify behavior-based worm detection methods into two main categories, machine learning methods, and traditional detection methods, and discuss them separately. The summary

and comparison of various detection methods are shown in Table 1.

Table 1: Summary of detection methods of worm attacks

References	**Data/Trace source**	**Detection Method**	**Acc**	**FAR**	**Remarks**
[6]	Private dataset	Hierarchical Clustering	99.50%	2%	Leveraged spreading behavior to detect worms.
[5]	Private dataset	Hierarchical Clustering	100%	1%	An improved version of [6].
[21]	Private dataset containing Code Red, Slammer, and Witty	SVM	N.A.	N.A.	Investigated the optimal configuration for support vector machines and associated kernel functions.
[8]	Private dataset	Hypersphere based SVM	N.A.	N.A.	A novel detection framework combining tree-based method, hypersphere based SVM, and Random Forest.
[1]	Private dataset	K-NN and Naïve Bayesian Classifier	98.07%	N.A.	Tested the method with real worm attack traffic.
[26]	DARPA 1999 and 3 types of real worm	Cluster algorithm	99.82%	N.A.	Used normal traffic in DARPA 1999 and the 3 kinds of worm traffic including Witty worm.
[23]	Private dataset	Tree based methods	95.60%	3.80%	Extracting primary feature from disassembly of worms.
[11]	Private dataset	ANN	99.96%	0%	Have the ability to predict the worm infection percentage in a network.
[24]	5 types of real worms	ANN	90%	N.A.	Evaluated impacts of different feature selection methods, time representation techniques on detection performance.
[25]	7 types of real worms	ANN	90%	N.A.	Collected features from device operation logs, system operation logs, and network behavior.
[4]	UNSW-NB15	D-CNN	N.A.	N.A.	Compared detection performances of deep learning and traditional machine learning approaches.

[32]	Apache-Knacker,ATPhttpd, and TSIG	CNN and DNN	99.27%	0.79%	Have the ability of detecting worms with high accuracy and automatically generating signature.
[30]	Dataset provided by Microsoft Malware Classification Challenge	DBN	N.A.	N.A.	The DBN model used in this work can deal with the imbalance problem of data
[19]	Ember	Deep learning based image processing techniques	98.80%	N.A.	A novel worm detection method combining intelligent image processing techniques.
[10]	Dataset from C4 Security	DNN	98.60%	N.A.	Can generate signatures for new malware variants regardless of the impact of code changes.
[2]	DARPA1999, Slammer pseudo worm, and RDP pseudo worm	Matching attack rules	100%	0%	Leveraged specific behaviors to detect worms
[20]	Slammer pseudo worm	Matching attack rules	100%	0%	Leveraged specific behaviors to detect worms
[9]	N.A.	Matching attack rules	N.A.	N.A.	Matching rules to distinguish normal applications and malwares.
[7]	N.A.	Matching attack rules	N.A.	N.A.	A real-time worm detection method.
[28]	Private dataset	Distance variation	80%	N.A.	Have short latency and high flexibility
[29]	Private dataset containing 8 worms	Matching attack rules	N.A.	Around 1.5%	A host-based detection method.
Acc: Accuracy; FAR: False alarm rate; N.A.: Not Available					

3.1 Worm Attacks Detection Using Machine Learning

In this part, we discuss the use of machine learning methods for worm attack detection, which we classify into two separate categories: traditional machine learning and deep learning methods.

1) Traditional Machine Learning Methods:

Researchers have conducted long-term research on the application of traditional machine learning algorithms in worm detection.

Chatzi et al. [6] collected email worm behavior data in an isolated cluster of computers and created their own dataset. Then, they used wavelet transform as a data downscaling tool and used hierarchical clustering for dataset classification. In addition, the author performed worm detection using traffic similarity searching methods and clustering algorithms in [5]. Both methods proposed by the author are used to detect email worms in local DNS servers.

sharma et al. [21] Recognized the ability of the SVM algorithm to identify standard features of data and developed a worm detection method using the SVM. The authors investigated the optimal configura-

tion method using SVM parameters and demonstrated the effectiveness of SVM in worm detection and the significant effect of parameters in SVM on classifier performance. After experiments and comparisons, it is concluded that the SVM model using the linear kernel is more suitable for this experiment.

Comar et al. [8] designed a new integrated detection method that exploits the high accuracy of supervised learning methods to detect known attacks and the adaptability of unsupervised learning methods to detect zero-day attacks. The 108 features in each stream are first extracted and then a tree-based feature transformation method is used to address the deficiencies of the dataset (noise, missing values, etc.). The authors then build a two-level classification architecture where the first level classifiers use a random forest as macro-level binary classifiers and the second level classifiers use a hypersphere-based SVM as micro-classifiers to distinguish between different types of worms.

Abdulla et al. [1] use Netflow, a traffic collection tool to capture flow data over a specified period of time and classify the flow data into DNS requests, DNS responses, DNS normal data (using legitimate domain names), and DNS anomalous data (using IP addresses instead of domain names). The authors used the quantities of these four types of flows as input to perform anomalous flow detection leveraging K nearest neighbor and Naïve Bayesian classifiers, respectively.

Sun et al. [26] proposed a worm detection method based on clustering and rough set theory. The algorithm extracts header features and constructs feature vectors for each message, and then uses an improved clustering algorithm to cluster the messages. Finally, the rough set theory is used to compute the boundary and lower approximations of the clusters to establish the interception rules.

Siddiqui et al. [23] presented a novel data mining framework for detecting worm attacks. The framework uses the frequency of occurrence of variable constant instruction sequences as its primary feature, where features are extracted from the disassembly of worms and normal applications. Three tree-based methods, Decision tree, Bagging, and Random Forest are used as binary classifiers to distinguish worms from clean applications.

2) Deep Learning Methods:

In this part, we discuss methods for worm detection using deep learning techniques. Compared with other types of attacks, data of worm attacks has less distinctive features, thus it's more difficult to detect, while using deep learning methods, high-level features in the data can be mined. The use of these detection methods is currently a popular research direction.

Farag et al. [11] proposed an artificial intelligence-based worm detection system that uses a port matching method to detect suspicious worm behaviors, after which the corresponding traffic features are input into the ANN for detection. With high detection accuracy, the system can also predict the infection rate of worms in the network, but the prediction error of the system would increase when the system is highly infected.

Stopel et al. [24] used ANN to detect five types of worms (Daber. A, Sasser. C, Deborm. Y, Padobot. Korgo. X, Slackor. A) and normal behavior, and the authors evaluated the effects of different feature selection (feature processing) methods and time representation techniques on the detection performance. After comparison, the Fisher score ranking technique with five retained features was the most effective way, while the time characterization technique had no significant effect on detection. In addition, the authors used the Causal Index method to analyze the correlation between the five retained features (after feature selection by the Fisher score ranking technique) and the five worms and normal behavior. In addition, also based on the ANN approach, in [25], the authors collected features from device operation logs, system operation logs, and network behavior, which were then classified using ANN after a feature selection operation. The dataset in this work involves seven real worms (W32.Dabber, W32.Deborm.Y, W32.Korgo.X, W32.Sasser.D, W32.Slackor.A, W32.HLLW.Doomjuice.B, W32.HLLW.Raleka.H).

Chapaneri et al. [4] tested UNSW-NB15 using several standard machine learning classifiers and neural network classifiers, and compared the effectiveness of different classification techniques for different types of attacks using the F-score evaluation metric, and concluded that the D-CNN (one-dimensional

convolutional neural network) classification method performs relatively better.

Zhou et al. [32] presented a deep learning-based worm detection system that is capable of detecting the presence of worms and automatically generating worm signatures. The proposed system consists of two main parts: a CNN worm detection module and a DNN worm signature generation module. In the worm detection module, three different payload processing methods, frequency processing, frequency weighted processing, and difference processing are used. The processed loads are used as CNN inputs to train the CNN model for worm detection. In the signature generation module, the DNN is used to learn the pattern and signature of the worm load to generate the DNN model, and the signature generation algorithm is used to generate the signature.

Yuxin et al. [30] use the Deep belief network (DBN) for malware detection including worms. The authors investigated the use of DBNs to build malware detection systems, whether untagged data improves detection rates, and whether the depth representation generated by DBNs can be used for feature extraction and dimension reduction.

Vinayakumar et al. [19] first compared classical machine learning algorithms with deep learning algorithms for intrusion detection using various public datasets. Then, the authors proposed a framework for zero-day malware detection using image processing techniques, which is shown to outperform other machine learning algorithms.

David et al. [10] presented a signature generation method that is independent of specific aspects of the malware, thus can detect new variants of worms. The method first runs the malware in a sandbox and generates a text file containing the program's behavior, converts the file to binary code and trains it as input to a DBN network, then uses a trained DBN to generate a representation of the malware behavior and generates signatures for the malware variants using the similarity principle. Then, supervised classification methods are used to test the representation power of these generated signatures.

3.2 Worm Attacks Detection Using Traditional Detection methods

Some worms have fixed and specific behaviors, so some traditional detection methods (e.g., knowledge-based methods) have more reliable performance in worm detection.

Ahmad et al. [2] and Shahzad et al. [20] proposed similar worm detection methods, which track SYN and UDP packets sent to detect the advent of worms. They associate SYN and UDP packets with the DNS resolution cache to determine whether there is a DNS lookup operation, and packets that do not have DNS lookup operations are considered suspicious packets. If the number of suspicious packets exceeds the threshold, a worm attack event is considered to have occurred, and the containment system would be used to block the suspicious traffic. In [2], in addition to using the above rules for detection and blocking, the authors also devised an LA policy to timely notify Adjacent network segments of worm events.

Cui et al. [9] designed the BINDER worm detection system. The authors point out that most normal network activity is triggered by users' activity, however, worm-generated network traffic is rarely triggered by user action but generally generated automatically by the worm. The BINDER system detects the presence of worm attacks by detecting the time lag between user events and traffic.

Chen et al. [7] proposed a real-time worm detection method in which a virtual machine is used to clone and run in parallel with a physical host, and all outgoing traffic from the physical host has to pass through the virtual machine. If the worm infects the physical host, it will infect the virtual host as the traffic passes through it, and the virtual host will behave maliciously. This approach is based on the knowledge that fast-spreading worms would try to infect as many hosts as possible in the shortest possible time, and their load size is usually small. Therefore, if a virtual machine receives some continuous traffic with a small transmission time and starts sending similar traffic to other hosts within a short period of time, the traffic will most likely be worm traffic.

Guo et al. [28] presented a behavior-based detection method for Instant messaging worms. Based on three defined characteristics, the authors use Mahalanobis distance to calculate the difference between the normal user's profile and the new traffic profile, and determine whether the traffic is abnormal based on a dynamic threshold calculated using the CUSUM algorithm.

Xiao et al. [29] presented a method for traffic detection based on a single process. For a process whose traffic is based on TCP or UDP protocols, it considers a process suspicious if the number of source ports and the rate of change exceeds a threshold. For an ICMP protocol process, if it sends too many ICMP packets with different destination addresses, the process would be considered suspicious. The authors use traffic similarity between suspicious processes to detect worm traffic.

4 Open Issues

There has been a lot of research work in the field of worm detection, but there are still some open issues in this field:

a) Issues in detection method: According to the classification method in this paper, worm detection methods can be divided into two categories: traditional detection methods and machine learning methods, the problems with traditional detection methods are obvious, these methods basically have no ability to detect zero-day attacks. In today's network where cyber-attacks frequently occur, the use of such detection methods alone is no longer sufficient to deal with external network threats. In terms of machine learning, the online learning and deployment of machine learning models will inevitably consume many resources of the machine, and how to make detection systems have low time response while achieving high accuracy is a problem to be solved.

b) Issues in datasets: The application of machine learning methods to worm detection has become a hot research topic, and the training of machine learning models requires the support of high-quality, high-quantity datasets. The existing network security datasets have problems such as insufficient attack categories, lack of the number of features, unbalanced amount of samples, etc. The existing public datasets are few in number and some of them are too old to be applicable in today's network environment. Non-public datasets, on the other hand, are difficult to acquire, and have insufficient credibility. Therefore, the construction of comprehensive, credible, and high-quality network security datasets is a significant issue to be solved in this field.

c) Issues in malicious traffic generator: Whether collecting worm attack datasets or testing prototypes of worm detection methods, software that can generate malicious traffic is needed. Such tools can be broadly divided into two categories, real worm code, and worm simulators. The worm code is generally obtained by decompiling the worm software by professionals. Due to the high level of difficulty in this type of work, access to worm source code is difficult. While worm simulators are often developed by researchers, their authors usually do not make their software or source code publicly available. As a result, such tools are also hard to acquire. The malicious traffic generator is crucial for both the construction of high-quality public datasets and the performance assurance of new worm detectors. The current lack of such tools is detrimental to researchers.

5 Conclusion

In this paper, we discussed a kind of threat that is relatively more harmful to the current Internet: worm attacks. First, the mechanism, characteristics, and major stages in the life cycle of worm attacks are discussed. Then, we introduced general categories and countermeasures of worm attacks. Afterward, we divided worm detection methods into two categories, namely, machine learning methods and traditional detection methods, and machine learning methods were further divided into traditional machine learning

methods and deep learning methods. The existing DDoS flooding attack detection methods are discussed by category. At last, some open issues related to the field of worm detection are talked about.

When designing a detection method, it is important to thoroughly understand the attack characteristics and select appropriate detection methods based on the requirements. While most current countermeasures focus on attack detection and interception, the security strategy proposed in [22] may be a new way of thinking that designs a novel key exchange and authentication protocol for wireless networks that can prevent network intrusion while protecting user privacy. According to our survey, worm defense still has a long way to go, and methods that support online detection with high detection rates are still to be developed.

Acknowledgments

This work was supported by the Joint Foundation of China University of Petroleum-Beijing at Karamay and the Fundamental Research Funds for the Central Universities (W20JB100070).

References

[1] S. Abdulla, S. Ramadass, A. Altaher, and A. Al-Nassiri. Employing machine learning algorithms to detect unknown scanning and email worms. *International Arab Journal of Information Technology*, 11(2):140–148, 2014.

[2] M. A. Ahmad, S. Woodhead, and D. Gan. A countermeasure mechanism for fast scanning malware. In *Proc. of the 2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security'16), London, UK*, pages 1–8. IEEE, June 2016.

[3] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett. A survey of deep learning methods for cyber security. *Information*, 10(4):122, 2019.

[4] R. Chapaneri and S. Shah. Detection of malicious network traffic using convolutional neural networks. In *Proc. of the 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT'19), IIT, Kanpur, India*, pages 1–6. IEEE, July 2019.

[5] N. Chatzis and N. Brownlee. Similarity search over dns query streams for email worm detection. In *Proc. of the 2009 International Conference on Advanced Information Networking and Applications (AINA'09), Bradford, UK*, pages 588–595. IEEE, May 2009.

[6] N. Chatzis and R. Popescu-Zeletin. Flow level data mining of dns query streams for email worm detection. In *Proc. of the 2009 International Workshop on Computational Intelligence in Security for Information Systems (CISIS'08), Burgos, Spain*, volume 53 of *Advances in Soft Computing*, pages 186–194. Springer, Berlin, Heidelberg, 2009.

[7] S. Chen, L. Liu, X. Wang, X. Zhang, and Z. Zhang. A host-based approach for unknown fast-spreading worm detection and containment. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 8(4):1–18, 2014.

[8] P. M. Comar, L. Liu, S. Saha, P. ning Tan, and A. Nucci. Combining supervised and unsupervised learning for zero-day malware detection. In *Proc. of the 2013 IEEE International Conference on Computer Communications (INFOCOM'13), Turin, Italy*, pages 2022–2030. IEEE, April 2013.

[9] W. Cui, R. H. Katz, and W. tian Tan. Binder: An extrusion-based break-in detector for personal computers. In *Proc. of the 2005 USENIX Annual Technical Conference, General Track, Anaheim, CA, USA*, pages 363–366. USENIX, April 2005.

[10] O. E. David and N. S. Netanyahu. Deepsign: Deep learning for automatic malware signature generation and classification. In *Proc. of the 2015 International Joint Conference on Neural Networks (IJCNN'15), Killarney, Ireland*, pages 1–8. IEEE, July 2015.

[11] I. A. Farag, M. A. Shouman, T. S. Sobh, and H. Z. El-Fiqi. Intelligent system for worm detection. *International Arab Journal of e-Technology*, 1(1):58–67, 2009.

[12] H.-D. J. Jeong, W. Hyun, J. Lim, and I. You. Anomaly teletraffic intrusion detection systems on hadoop-based platforms: A survey of some problems and solutions. In *Proc. of the 15th International Conference on Network-Based Information Systems (NBIS'12), Melbourne, VIC, Australia*, pages 766–770. IEEE, September 2012.

[13] X. Jing, Z. Yan, and W. Pedrycz. Security data collection and data analytics in the internet: A survey. *IEEE Communications Surveys & Tutorials*, 21(1):586–618, 2018.

[14] T. T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE communications surveys & tutorials*, 10(4):56–76, 2008.

[15] T. T. Nguyen and V. J. Reddi. Deep reinforcement learning for cyber security. arXiv:1906.05799, 2019. `https://arxiv.org/abs/1906.05799` [Online; Accessed on November 10, 2020].

[16] A. Nisioti, A. Mylonas, P. D. Yoo, and V. Katos. From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods. *IEEE Communications Surveys & Tutorials*, 20(4):3369–3388, 2018.

[17] L. Pele, M. Salour, and X. Su. A survey of internet worm detection and containment. *IEEE Communications Surveys & Tutorials*, 10(1):20–35, 2008.

[18] P. Porras, L. Briesemeister, K. Skinner, K. Levitt, J. Rowe, and Y.-C. A. Ting. A hybrid quarantine defense. In *Proc. of the 2004 ACM workshop on Rapid malcode (WORM'04), Washington DC, USA*, pages 73–82. ACM, October 2004.

[19] V. Ravi, M. Alazab, S. K.P, P. Poornachandran, and S. Venkatraman. Robust intelligent malware detection using deep learning. *IEEE Access*, 7:46717–46738, 2019.

[20] K. Shahzad and S. Woodhead. Towards automated distributed containment of zero-day network worms. In *Proc. of the 5th International Conference on Computing, Communications and Networking Technologies (ICCCNT'14), Hefei, China*, pages 1–7. IEEE, July 2014.

[21] O. Sharma, M. Girolami, and J. Sventek. Detecting worm variants using machine learning. In *Proc. of the 2007 ACM conext conference (CoNEXT'07), New York, NY, USA*, pages 1–12. ACM, December 2007.

[22] V. Sharma, I. You, F.-Y. Leu, and M. Atiquzzaman. Secure and efficient protocol for fast handover in 5g mobile xhaul networks. *Journal of Network and Computer Applications*, 102:38–57, 2018.

[23] M. Siddiqui, M. C. Wang, and J. Lee. Detecting internet worms using data mining techniques. *Journal of Systemics, Cybernetics and Informatics*, 6(6):48–53, 2009.

[24] D. Stopel, Z. Boger, R. Moskovitch, Y. Shahar, and Y. Elovici. Improving worm detection with artificial neural networks through feature selection and temporal analysis techniques. *International Journal of Computational Science and Engineering*, 15:202–208, 2006.

[25] D. Stopel, R. Moskovitch, Z. Boger, Y. Shahar, and Y. Elovici. Using artificial neural networks to detect unknown computer worms. *Neural Computing and Applications*, 18(7):663–674, 2009.

[26] W.-C. Sun and Y.-M. Chen. A rough set approach for automatic key attributes identification of zero-day polymorphic worms. *Expert Systems with Applications*, 36(3):4672–4679, 2009.

[27] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A taxonomy of computer worms. In *Proc. of the 2003 ACM workshop on Rapid malcode (WORM'03), Washington DC, USA*, pages 11–18. ACM, October 2003.

[28] wei guo, X. Wang, and H. X. Zhou. A behavior approach to instant messaging worm detection. In *Proc. of the 2015 International Conference on Artificial Intelligence and Industrial Engineering (AIIE'15), Phuket, Thailand*. Atlantis Press, July 2015.

[29] F. Xiao, H. Hu, B. Liu, and X. Chen. Ptbbwd: A fast process traffic behavior based worm detection algorithm. In *Proc. of the 2008 International Seminar on Future Information Technology and Management Engineering (FITME'08), Leicestershire, UK*, pages 181–186. IEEE, November 2008.

[30] D. Yuxin and Z. Siyi. Malware detection based on deep learning algorithm. *Neural Computing and Applications*, 31(2):461–472, 2019.

[31] S. T. Zargar, J. Joshi, and D. Tipper. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials*, 15(4):2046–2069, 2013.

[32] H. Zhou, Y. Hu, X. Yang, H. Pan, W. Guo, and C. C. Zou. A worm detection system based on deep learning.

IEEE Access, 2020.

Author Biography

Liang Wang is pursuing his Ph.D. in the National Engineering Laboratory for Next Generation Internet Technologies (NGIT) laboratory of the School of Electronic Information Engineering, Beijing Jiaotong University. His research interests are mainly in novel network architectures and network security.