DETECTION OF CYBER ATTACK IN NETWORK USING MACHINE LEARNING TECHNOLOGY

K.Anup kumar[1], PH.Swarna Rekha[2], Yekula.Chandra sekhar reddy[3], Vennam.Balu[4], Ture.Neha[5], Tanneru.Sandhya[6]

[1]Assistant professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[2]Assistant professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[3,4,5,6] IV[th] Btech Student, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

ABSTRACT

Cyber-crime is proliferating everywhere exploiting every kind of vulnerability to the computing environment. Ethical Hackers pay more attention towards assessing vulnerabilities and recommending mitigation methodologies. Most techniques used in today's IDS are not able to deal with the dynamic and complex nature of cyber-attacks on computer networks. Machine learning for cyber security has become an issue of great importance recently due to the effectiveness of machine learning in cyber security issues. Machine learning techniques have been applied for major challenges in cyber security issues like intrusion detection, malware classification and detection and spam detection . Although machine learning cannot automate a complete cyber security system, it helps to identify cyber security threats more efficiently than other software-oriented methodologies, and thus reduces the burden on security analysts. Hence, efficient adaptive methods like various techniques of machine learning can result in higher detection rates, lower false alarm rates and reasonable computation and communication costs.

Keywords: Cyber-crime, Machine learning, Cyber-security, Intrusion detection system.

1. INTRODUCTION

Contrasted with the past, improvements in PC and correspondence innovations have given broad and propelled changes. The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection

Systems (IDS) has been created to maintain a strategic distance from digital assaults. Right now, learning the bolster support vector machine (SVM) calculations were utilized to recognize port sweep endeavors dependent on the new CICIDS2017 dataset with 97.80%, 69.79% precision rates were accomplished individually. Rather than SVM we can introduce some other algorithms like random forest, CNN, ANN where these algorithms can acquire accuracies like SVM – 93.29, CNN – 63.52, Random Forest – 99.93, ANN – 99.11.

2. LITERATURE SURVEY

A) R. Christopher, “Port scanning techniques and the defense against them,” SANS Institute, 2001.

Port Scanning is one of the most popular techniques attackers use to discover services that they can exploit to break into systems. All systems that are connected to a LAN or the Internet via a modem run services that listen to well-known and not so well-known ports. By port scanning, the attacker can find the following information about the targeted systems: what services are running, what users own those services, whether anonymous logins are supported, and whether certain network services require authentication. Port scanning is accomplished by sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can be probed for further weaknesses. Port scanners are important to network security technicians because they can reveal possible security vulnerabilities on the targeted system. Just as port scans can be ran against your systems, port scans can be detected and the amount of information about open services can be limited utilizing the proper tools. Every publicly available system has ports that are open and available for use. The object is to limit the exposure of open ports to authorized users and to deny access to the closed ports.

B) S. Staniford, J. A. Hoagland, and J. M. McAlerney, “Practical automated detection of

stealthy portscans," Journal of Computer Security, vol. 10, no. 1-2, pp. 105–136, 2002.

Portscanning is a common activity of considerable importance. It is often used by computer attackers to characterize hosts or networks which they are considering hostile activity against. Thus it is useful for system administrators and other network defenders to detect portscans as possible preliminaries to a more serious attack. It is also widely used by network defenders to understand and find vulnerabilities in their own networks. Thus it is of considerable interest to attackers to determine whether or not the defenders of a network are portscanning it regularly. However, defenders will not usually wish to hide their portscanning, while attackers will. For definiteness, in the remainder of this paper, we will speak of the attackers scanning the network, and the defenders trying to detect the scan. There are several legal/ethical debates about portscanning which break out regularly on Internet mailing lists and newsgroups. One concerns whether portscanning of remote networks without permission from the owners is itself a legal and ethical activity. This is presently a grey area in most jurisdictions. However, our experience from following up on unsolicited remote portscans we detect in practice is that almost all of them turn out to have come from compromised hosts and thus are very likely to be hostile. So we think it reasonable to consider a portscan as at least potentially hostile, and to report it to the administrators of the remote network from whence it came. However, this paper is focussed on the technical questions of how to detect portscans, which are independent of what significance one imbues them with, or how one chooses to respond to them. Also, we are focussed here on the problem of detecting a portscan via a network intrusion detection system (NIDS). We try to take into account some of the more obvious ways an attacker could use to avoid detection, but to remain with an approach that is practical to employ on busy networks. In the remainder of this section, we first define portscanning, give a variety of examples at some length, and discuss ways attackers can try to be stealthy. In the next section, we discuss a variety of prior work on portscan detection. Then we present the algorithms that we propose to use, and give some very preliminary data justifying our approach. Finally, we consider possible extensions to this work, along with other applications that might be considered. Throughout, we assume the reader is familiar with Internet protocols, with basic ideas about network intrusion detection and scanning, and with elementary probability theory, information theory, and linear algebra. There are two general purposes that an attacker might have in conducting a portscan: a primary one, and a secondary one. The primary purpose is that of gathering information about the reachability and status of certain combinations of IP address and port (either TCP or UDP). (We do not directly discuss ICMP scans in this paper, but the ideas can be extended to that case in an obvious way.) The secondary purpose is to flood intrusion detection systems with alerts, with the intention of distracting the network defenders or preventing them from doing their jobs. In this paper, we will mainly be concerned with detecting information gathering portscans, since detecting flood portscans is easy. However, the possibility of being maliciously

flooded with information will be an important consideration in our algorithm design. We will use the term scan footprint for the set of port/IP combinations which the attacker is interested in characterizing. It is helpful to conceptually distinguish the footprint of the scan, from the script of the scan, which refers to the time sequence in which the attacker tries to explore the footprint. The footprint is independent of aspects of the script, such as how fast the scan is, whether it is randomized, etc. The footprint represents the attacker's information gathering requirements for her scan, and she designs a scan script that will meet those requirements, and perhaps other non-information-gathering requirements (such as not being detected by an NIDS). The most common type of portscan footprint at present is a horizontal scan. By this, we mean that an attacker has an exploit for a particular service, and is interested in finding any hosts that expose that service. Thus she scans the port of interest on all IP addresses in some range of interest. Also at present, this is mainly being done sequentially on TCP port 53 (DNS)

3. PROBLEM ANALYSIS

3.1 EXISTING APPROACH:

Blameless Bayes and Principal Component Analysis (PCA) were been used with the KDD99 dataset by Almansob and Lomte [9].Similarly, PCA, SVM, and KDD99 were used Chithik and Rabbani for IDS [10]. In Aljawarneh et al's. Paper, their assessment and examinations were conveyed reliant on the NSL-KDD dataset for their IDS model [11] Composing inspects show that KDD99 dataset is continually used for IDS [6]–[10].There are 41 highlights in KDD99 and it was created in 1999. **Consequently, KDD99 is old and doesn't give any data about cutting edge new assault types**, example, multi day misuses and so forth. In this manner we utilized a cutting-edge and new CICIDS2017 dataset [12] in our investigation.

DRAW BACKS

1) Strict Regulations
2) Difficult to work with for non-technical users
3) Restrictive to resources
4) Constantly needs Patching
5) Constantly being attacked

3.2 PROPOSED SYSTEM

important steps of the algorithm are given in below. 1) Normalization of every dataset. 2) Convert that dataset into the testing and training. 3) Form IDS models with the help of using RF, ANN, CNN and SVM algorithms. 4) Evaluate every model's performances

ADVANTAGES

- Protection from malicious attacks on your network.
- Deletion and/or guaranteeing malicious elements within a preexisting network.
- Prevents users from unauthorized access to the network.
- Deny's programs from certain resources that could be infected.
- Securing confidential information

3.3 SOFTWARE AND HARDWARE REQUIREMENTS

SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Python idel 3.7 version (or)**
- **Anaconda 3.7 (or)**
- **Jupiter (or)**
- **Google colab**

HARDWARE REQUIREMENTS

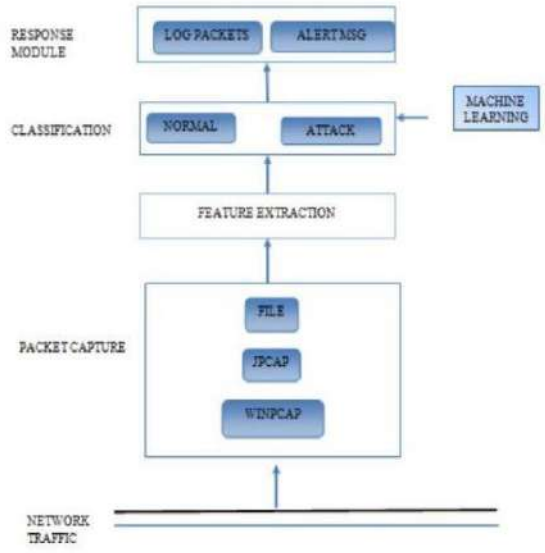
Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- **Operating system : windows, linux**
- **Processor : minimum intel i3**
- **Ram : minimum 4 gb**
- **Hard disk : minimum 250gb**

3.4 ALGORITHM

- **ANN**
- **CNN**
- **Random forest**

4.SYSTEM ARCHITECTURE



4.1 PROBE ATTACKS

In the context of network security, a probe attack (also known as a reconnaissance attack) is an attempt by an attacker to gain information about a target network or system, without attempting to damage or disrupt it. The goal of a probe attack is to identify vulnerabilities in the target system or network that can be exploited in a subsequent attack.

Probe attacks typically involve sending queries or requests to the target system or network, and analyzing the responses to gain information about the system's configuration, security measures, and other details that can be used to launch a successful attack. Common methods used in probe attacks include port scanning, network mapping,

and fingerprinting, which involve probing different parts of the target network or system to gather information.

Probe attacks can be difficult to detect, as they do not involve any obvious signs of malicious activity. However, by monitoring network traffic and analyzing patterns of incoming requests, it is possible to identify suspicious activity that may indicate a probe attack in progress. This is one of the reasons why machine learning models are used to classify network traffic into different types of attacks, including probe attacks.

4.2 R2L

In the context of network security, R2L (short for "Remote to Local") is a type of network attack where an attacker from a remote location tries to gain unauthorized access to a target system or network by exploiting vulnerabilities in its security mechanisms. The attacker attempts to impersonate a legitimate user of the system by using stolen or guessed credentials, and gains access to sensitive data or system resources.

R2L attacks can take various forms, including password guessing, exploit-based attacks, and social engineering. Password guessing involves trying to guess the login credentials of a legitimate user, either by using brute-force methods or by using a list of commonly used passwords. Exploit-based attacks involve exploiting vulnerabilities in the system's software or configuration to gain unauthorized access. Social engineering attacks involve tricking a user into disclosing sensitive information or installing malware on their system.

Detecting R2L attacks can be challenging, as they often involve attempts to bypass or evade the system's security measures. However, by monitoring network traffic and analyzing patterns of incoming requests, it is possible to identify suspicious activity that may indicate an R2L attack in progress. This is one of the reasons why machine learning models are used to classify network traffic into different types of attacks, including R2L attacks.

4.3 U2R

In the context of network security, U2R (short for "User to Root") is a type of network attack where an attacker who has already gained access to a system tries to escalate their privileges to gain root-level access to the system. The attacker exploits vulnerabilities in the system's software or configuration to gain access to sensitive data or system resources, and then uses that access to gain full control over the system.

U2R attacks can take various forms, including buffer overflow attacks, privilege escalation attacks, and backdoor attacks. Buffer overflow attacks involve sending too much data to a program or system, causing it to crash or behave unpredictably, which the attacker can then exploit to gain root access. Privilege escalation attacks involve exploiting vulnerabilities in the system's security mechanisms to gain higher levels of access than they are authorized for. Backdoor attacks involve installing software or creating a hidden access point that can be used to gain access to the system in the future.

Detecting U2R attacks can be difficult, as they often involve sophisticated techniques for bypassing the system's security measures. However, by monitoring network traffic and analyzing patterns of incoming requests, it is possible to identify suspicious activity that may indicate a U2R attack in progress. This is one of the reasons why machine learning models are used to classify network traffic into different types of attacks, including U2R attacks.

5. CONCLUSION

Right now, estimations of help vector machine, ANN, CNN, Random Forest and profound learning calculations dependent on modern CICIDS2017 dataset were introduced relatively. Results show that the profound learning calculation performed fundamentally preferable outcomes over SVM, ANN, RF and CNN. We are going to utilize port sweep endeavors as well as other assault types with AI and profound learning calculations, apache Hadoop and sparkle innovations together dependent on this dataset later on. All these calculation helps us to detect the cyber attack in network. It happens in the way that when we consider long back years there may be so many attacks happened so when these attacks are recognized then the features at which values these attacks are happening will be stored in some datasets. So by using these datasets we are going to predict whether cyber attack is done or not. These predictions can be done by four algorithms like SVM, ANN, RF, CNN this paper helps to identify which algorithm predicts the best accuracy rates which helps to predict best results to identify the cyber attacks happened or not.

6. SCOPE OF FUTURE WORK

These predictions can be done by four algorithms like SVM, ANN, CNN this paper helps to identify which algorithm predicts the best accuracy rates which helps to predict best results to identify the cyber attacks happened or not.

7. REFERENCES

[1] K. Graves, Ceh: Official certified ethical hacker review guide: Exam 312-50. John Wiley & Sons, 2007.

[2] R. Christopher, “Port scanning techniques and the defense against them,” SANS Institute, 2001.

[3] M. Baykara, R. Das¸, and I. Karado ˘gan, “Bilgi g ¨uvenli ˘gi sistemlerinde kullanilan arac¸larin incelenmesi,” in 1st International Symposium on Digital Forensics and Security (ISDFS13), 2013, pp. 231–239.

[4] S. Staniford, J. A. Hoagland, and J. M. McAlerney, “Practical automated detection of stealthy portscans,” Journal of Computer Security, vol. 10, no. 1-2, pp. 105–136, 2002.

[5] S. Robertson, E. V. Siegel, M. Miller, and S. J. Stolfo, “Surveillance detection in high bandwidth environments,” in DARPA Information Survivability Conference and Exposition, 2003. Proceedings, vol. 1. IEEE, 2003, pp. 130–138.

[6] K. Ibrahimi and M. Ouaddane, “Management of intrusion detection systems based-kdd99: Analysis with lda and pca,” in Wireless Networks and

Mobile Communications (WINCOM), 2017 International Conference on. IEEE, 2017, pp. 1–6.

[7] N. Moustafa and J. Slay, “The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems,” in Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on. IEEE, 2015, pp. 25–31.

[8] L. Sun, T. Anthony, H. Z. Xia, J. Chen, X. Huang, and Y. Zhang, “Detection and classification of malicious patterns in network traffic using benford’s law,” in Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017. IEEE, 2017, pp. 864–872.

[9] S. M. Almansob and S. S. Lomte, “Addressing challenges for intrusion detection system using naive bayes and pca algorithm,” in Convergence in Technology (I2CT), 2017 2nd International Conference for. IEEE, 2017, pp. 565–568.

[10] M. C. Raja and M. M. A. Rabbani, “Combined analysis of support vector machine and principle component analysis for ids,” in IEEE International Conference on Communication and Electronics Systems, 2016, pp. 1–5.

[11] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” Journal of Computational Science, vol. 25, pp. 152–160, 2018.

[12] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization.” in ICISSP, 2018, pp. 108–116.

[13] D. Aksu, S. Ustebay, M. A. Aydin, and T. Atmaca, “Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm,” in International Symposium on Computer and Information Sciences. Springer, 2018, pp. 141–149.

[14] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, “Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark,” IEEE Access, 2018.

[15] P. A. A. Resende and A. C. Drummond, “Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling,” Security and Privacy, vol. 1, no. 4, p. e36, 2018.

[16] C. Cortes and V. Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

[17] R. Shouval, O. Bondi, H. Mishan, A. Shimoni, R. Unger, and A. Nagler, “Application of machine learning algorithms for clinical predictive modeling: a data-mining approach in sct,” Bone marrow transplantation, vol. 49, no. 3, p. 332, 2014.