International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 04 | Apr 2019 www.irjet.net p-ISSN: 2395-0072

# Security Risk Assessment on Social Media using Artificial Intelligence Methodology

**Saraswathi.E, Krishna Agarwal, Aayush Sharma, Kunal Kundu, Prekshit Barole**

---

***Abstract* —** Risk assessment and prevention system is one of the best ways to identify the threat on any social media, it scan the overall network, server functions and raise the suspicious alarm if any suspicious activity is detected in the network traffic. It monitors the system continuously and responds accordingly to the threat environment. This response action varies from phase to phase. Here suspicious activities are detected by the help of an artificial intelligence which acts as a virtual analyst concurrently with network intrusion detection system to protect from the threat environment and taking appropriate action with the permission of the analyst. In its final phase where packet analysis is carried out to surf for threat vectors and then categorize supervised and unsupervised data. Where the unsupervised data will be decoded or converted to supervised data with help of analyst feedback and then auto-update the algorithm (Virtual Analyst Algorithm). So that it evolves the algorithm (with Active Learning Mechanism) itself by time and become more efficient, strong. So it can able to defend form similar or same kind of attacks.

**Keywords:** Artificial Intelligence, Intrusion Detection System, Network Security, Social Media

## I. INTRODUCTION

In recent years, the use of social media has raised up drastically. Online social networking sites such as Facebook, Twitter, Google+, etc. have been growing at an immense rate and now have over hundreds of millions of everyday online users. Due to the sharing nature of online social networks, users expose many personal details about themselves, either knowingly or unknowingly; details, such as DOB, email address, name, and even phone numbers are frequently exposed. Hacker utilizing the client's gathered information, an can send spam messages trying to bait such clients into malignant sites or even coerce them into exchanging cash to the assailant's record. An aggressor can be an online "exploiter", who utilizes online assaults so as to pick up data which will empower them to get the client's trust and persuade the client to meet, all things which he wants. In this age, where almost everything is done online or digitally, it is potentially dangerous if an anonymous third party gets access to your system. So, to curb such situations Intrusion Detection Systems and Intrusion Prevention System come into play. An intrusion detection system (IDS) is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered. When Intrusion detection system senses a malicious activity, it immediately reports it to the administrator, and in some cases, some intrusion detection frameworks are equipped for taking activities when malicious activity or abnormal traffic is identified, including blocking traffic sent from suspicious IP addresses. But, it is a given fact that all Intrusion Detection systems need a fine tuning which means properly configuring their intrusion detection systems to recognize what normal traffic on their network looks like compared to potentially malicious activity, as they can prone to false alarms. By presenting this paper, we wish to implement Intrusion Prevention System using Artificial Intelligence Methodology on Malicious URL posted on social media. By doing so, we wish to create a system which detects these malicious URL or links, and once when detected are aborted, and stores for future detections, in a way-self learning. This will eliminate any chance of human error and increase the overall accuracy of the system.

## II. EXISTING SYSTEM

The Existing system is a beneficiary for the hacker's community as they are not filtered from the normal user they can slowly exploit any user without knowing by the users. This generation entering the workforce that assumes this technology will not only be available for their use, but it is also essential to the way they communicate with colleagues. While there are many benefits that come with using social networks both internally and externally, the policy and architecture to defend against the risks must be addressed proactively .Security success is all about the right combination of people, process, policy and technology. This can be combated by having a network management systems which should have the abilities of intellectual reasoning, dynamic real time decision making, and experience based self- adaptation and improvement. The design of such efficient, dynamic and automated social network management framework requires support from the field of artificial intelligence. Dealing with uncertainty and inconsistency has been a part of AI from its origins. Disadvantages of current system are:

1. Un-regulated flow of information
2. Loss of intellectual property & proprietary information
3. Disclosure of personal information
4. Information security breach

The disadvantages of the existing systems include operational complexity, misconfigurations and vulnerabilities resulting in the violation of security properties.

## III. PROPOSED SYSTEM

Integrating AI techniques to solve intrusion detection system (IDS) is an aid in achieving smart environments that deliver adaptive behaviors depending on the intentions of the user. Artificial Intelligence (AI) is one of the more expected computer science areas created in recent years. AI is a scientific discipline that attempts to make human intellectual and cognitive capabilities available through information processing systems. Nowadays, due to their complexity, many security and privacy issues cannot be solved optimally. Artificial intelligence has proved extremely useful and well- equipped to solve these problems in these situations. The IDS process is automated in the system and threats are identified using AI. Advantages of Proposed System are:

1. It can handle volume of data
2. It can learn over time
3. It identifies unknown threats

## IV. METHODOLOGY

An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any malicious activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources, and uses alarm filtering techniques to distinguish malicious activity from false alarms One way of categorizing IDS's is based on the method of detection intrusion.

### A. Host-Based Intrusion Detection System

Host-based intrusion detection systems are systems that monitor the device on which they are installed, or directly connected to. The way they monitor the system can range from monitoring the state of the main system through audit logs, to monitoring program execution. Since HIDS rely so much on audit logs, they can become limited by them. Another issue can be the sheer volume of the audit logs. Every monitored log needs to be parsed; this means that the HIDS can have a big impact on the performance of the host system if it is installed there. Another disadvantage is that any vulnerability that causes the audit files to be changed, also impacts the integrity of the HIDS. If an audit file is changed, the HIDS cannot see and detect what truly happened.

### B. Network-Based Intrusion Detection System

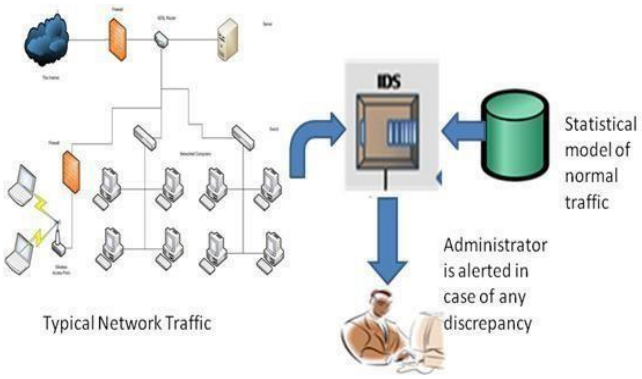
Network-based intrusion detection systems are placed at certain points within a network in order to monitor traffic from and to devices within the network. They operate on the same concept as wiretapping. They "tap" into a network and listen to all communication that happens. The intruder could try to minimize his network activity, but the risk is lower. NIDS are also more portable than HIDS. They monitor traffic over a network and are independent of the operating system they run on. The system can analyze the traffic using multiple techniques to determine whether the data is malicious. There are two different ways to analyze the network data. Packet-based analysis uses the entire packet including the headers and payload. An intrusion detection system that uses packet-based analysis is called a packet-based network intrusion detection system. The advantage of this type of analysis is that there is a lot of data to work with. Every single byte of the packet could be used to determine whether the packet is malicious or not. Flow based analysis doesn't use individual packets but uses general aggregated data

about network flows. An intrusion detection system that uses flow- based analysis is called a flow-based network intrusion detection system. A flow is defined as a single connection between the host and another device.

### C. Intrusion Prevention System

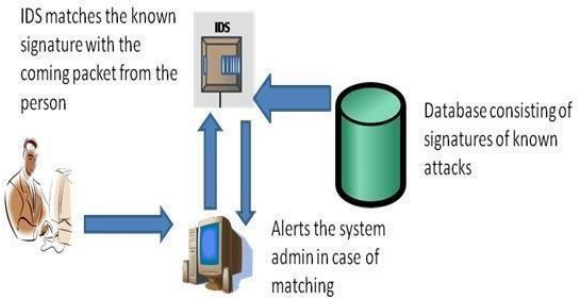
An intrusion prevention system or IPS/IDPS is an intrusion detection system that also has to ability to prevent attacks. An IDS does not necessarily need to be able to detect attacks at the exact moment they occur, although it is preferred. An IPS needs to be able to detect attacks real-time since it also needs to be able to prevent these attacks. For network attacks these prevention actions could be closing the connection, blocking an IP or limiting the data throughput. The change to requiring attacks to be detected at real time can severely impact the methods that are used to detect these attacks. For example, an IDS might give an alert even though the IDS are not certain that whatever it is alerting is actually an anomaly. An IPS needs to be certain before it can take action. Otherwise the IPS might take actions which the business employing the IPS does not want.

3.4. Detection There is multiple different methods to detect intrusions. There are Signature based and Anomaly Based methods. Signature based methods compare so called "signatures" with an existing database of signatures. A packet or flow record is decomposed into features that together construct a signature. If the signature of an incoming flow or packet matches with a signature in the database, it is flagged as malicious. Signature-based methods have little overhead in both computation and preprocessing as it only tries to match incoming signatures to known signatures in the database. Because it only compares signatures, it is easy to deploy within a network. The system does not need to learn what the traffic within a network looks like. Signature based methods are very effective against known attacks. New attacks cannot be detected unless the database is updated with new signatures. It is also possible for attackers to avoid being caught by signature based methods, only a slight modification of the "signature" is required in order to bypass the exact matching.




[Fig-1 IDS Based On Signature]

Anomaly based methods; also called Behavior based methods are methods in which the IDS try to model the behavior of network traffic. When an incoming packet deviates from this model, it is flagged as malicious and an alert is sent. Because they use a statistical model of normal behavior, they should be able to detect all deviations from this normal behavior. As a result, new attacks that deviate too much from normal behavior are detected as well. Since a model of the network traffic needs to be created, the system cannot be deployed into a network and be expected to work. The system needs to learn the behavior of the network traffic. Problems, such as generating a lot of false positive alarms, can arise when training data includes mistakes, such as misclassifications. Artificial intelligence algorithms can be used as an anomaly based method. Artificial intelligence techniques have the ability to learn from data and decide whether new data is malicious.




[Fig-2 IDS Based On Anomaly]

## V. IMPLEMENTATION

### A. Technology Stack

The main library that was used is Scikit-learn. Scikit- learn is a robust artificial intelligence library for Python. It is built upon NumPy, SciPy, and matplotlib. It is also open source and commercially usable with the BSD license learn. This library was chosen since the library offers the most important algorithms, the documentation. Scikit-learn also contain different methods to visualize artificial intelligence algorithms such as a graph to show the learning curve. These can be a useful tool to evaluate the performance of artificial intelligence algorithms. It also contains methods to calculate the F-score. This is useful since that means that mistakes when calculating the F-score are less likely to happen.

### B. Program Execution

The implementation works in different steps. A JSON config file is used to define the elements that are used within the program. This contains the data to be used for learning, for checking, the artificial intelligence algorithm, etc. Once the config file has been read, the program can start the training phase. In this phase the specified algorithm is used and trained using the given data. Afterwards the prediction phase starts. This phase uses the prediction data and gathers all results. The structure of the program and the modules reflect these different phases.

### C. Structure

The implementation is built to be modular. The first module is the artificial intelligence module. This module contains all artificial intelligence algorithms that can be used. There is also a feature module. This module contains the available classes that can be used to extract features from the flows. A loader module contains all classes required to load the data from the different datasets.

A training module contains the different classes used for training. These classes use a loader class and pass the data to the artificial intelligence algorithm. They define which data is supposed to be used (for example, using only abnormal behaviour and leaving out the normal behaviour). Finally there is a results module. This module receives all the output from the artificial intelligence algorithms and has to log these or visualize them.

### D. Datasets

In order to test the implementation and the algorithms, different datasets were used. Each dataset is used to test a different aspect of the artificial intelligence algorithms. First, a subset of a dataset has to be chosen to be fed to the artificial intelligence algorithms for learning. Afterwards, using the method, the algorithm is tested using another subset of the same dataset.

In the next step, the algorithms are tested using real- world data that is labelled. Finally, in the fourth step, the algorithms are tested using raw, unlabelled real- world data. This is to make sure that the algorithm performs well on unprocessed real-world data. Several datasets have been used to test the artificial intelligence algorithms.

| Scen. | Total Flows | Botnet Flows | Normal Flows | C&C Flows | Background Flows |
|---|---|---|---|---|---|
| 1 | 2,824,636 | 39,933(1.41%) | 30,387(1.07%) | 1,026(0.03%) | 2,753,290(97.47%) |
| 2 | 1,808,122 | 18,839(1.04%) | 9,120(0.5%) | 2,102(0.11%) | 1,778,061(98.33%) |
| 3 | 4,710,638 | 26,759(0.56%) | 116,887(2.48%) | 63(0.001%) | 4,566,929(96.94%) |
| 4 | 1,121,076 | 1,719(0.15%) | 25,268(2.25%) | 49(0.004%) | 1,094,040(97.58%) |
| 5 | 129,832 | 695(0.53%) | 4,679(3.6%) | 206(1.15%) | 124,252(95.7%) |
| 6 | 558,919 | 4,431(0.79%) | 7,494(1.34%) | 199(0.03%) | 546,795(97.83%) |
| 7 | 114,077 | 37(0.03%) | 1,677(1.47%) | 26(0.02%) | 112,337(98.47%) |
| 8 | 2,954,230 | 5,052(0.17%) | 72,822(2.46%) | 1,074(2.4%) | 2,875,282(97.32%) |
| 9 | 2,753,884 | 179,880(6.5%) | 43,340(1.57%) | 5,099(0.18%) | 2,525,565(91.7%) |
| 10 | 1,309,791 | 106,315(8.11%) | 15,847(1.2%) | 37(0.002%) | 1,187,592(90.67%) |
| 11 | 107,251 | 8,161(7.6%) | 2,718(2.53%) | 3(0.002%) | 96,369(89.85%) |
| 12 | 325,471 | 2,143(0.65%) | 7,628(2.34%) | 25(0.007%) | 315,675(96.99%) |
| 13 | 1,925,149 | 38,791(2.01%) | 31,939(1.65%) | 1,202(0.06%) | 1,853,217(96.26%) |

[Fig-3 Dataset of CTU-13]

The CTU-13 dataset has been used for steps one to three for the testing of the artificial intelligence algorithms. This is a labelled dataset. It contains botnet behaviour, normal and background traffic. The data was captured in the CTU University, Czech Republic, in 2011. It consists of thirteen different captures, each of which runs a different botnet malware. Figure 4 shows the amount of data within each capture. Note that the captured data is only from a couple hours. The flows within the dataset contain extra information. Each capture contains only a small amount of botnet samples. Most flows are background flows.

This is expected of botnet behaviour since it does not generate a large amount of network traffic. Each flow is labelled with its exact source. This could be Google analytics, Google webmail or a windows update. The flows within the dataset only contain the regular information that is found within net flow. The abnormal behaviour within this dataset is internal abnormal behaviour. In the evaluation chapter, this dataset is called the CTU dataset.

| Id | Duration(hrs) | # Packets | #NetFlows | Size | Bot | #Bots |
|---|---|---|---|---|---|---|
| 1 | 6.15 | 71,971,482 | 2,824,637 | 52GB | Neris | 1 |
| 2 | 4.21 | 71,851,300 | 1,808,123 | 60GB | Neris | 1 |
| 3 | 66.85 | 167,730,395 | 4,710,639 | 121GB | Rbot | 1 |
| 4 | 4.21 | 62,089,135 | 1,121,077 | 53GB | Rbot | 1 |
| 5 | 11.63 | 4,481,167 | 129,833 | 37.6GB | Virut | 1 |
| 6 | 2.18 | 38,764,357 | 558,920 | 30GB | Menti | 1 |
| 7 | 0.38 | 7,467,139 | 114,078 | 5.8GB | Sogou | 1 |
| 8 | 19.5 | 155,207,799 | 2,954,231 | 123GB | Murlo | 1 |
| 9 | 5.18 | 115,415,321 | 2,753,885 | 94GB | Neris | 10 |
| 10 | 4.75 | 90,389,782 | 1,309,792 | 73GB | Rbot | 10 |
| 11 | 0.26 | 6,337,202 | 107,252 | 5.2GB | Rbot | 3 |
| 12 | 1.21 | 13,212,268 | 325,472 | 8.3GB | NSIS.ay | 3 |
| 13 | 16.36 | 50,888,256 | 1,925,150 | 34GB | Virut | 1 |

[Fig-4 Data collection by botnet]

### E. Algorithm selection

Both supervised and unsupervised algorithms have been used. The algorithms are the most common algorithms. Before more complex algorithms such as deep neural networks should be used, the more common and general algorithms should be tested.

#### 1. Unsupervised learning

K-Means clustering is used in order to test whether results can be found using clustering algorithms. K- means is a simple clustering algorithm and already gives an indication whether a problem can be solved using clustering, or whether clustering offers no advantage. However, no method was found to verify whether to clusters that the K-means algorithm made were correct. One-class Support Vector machines are used in an attempt to use binary classification. They are quite fast in execution. They were used to find out whether it is a viable technique to pre-process incoming data and check whether a One-class Support Vector machine finds it to be abnormal behaviour before passing it to other algorithms.

#### 2. Supervised learning

Support vector machines have been used in the implementation. It is a popular algorithm and can do both linear and non-

linear classification which makes it a promising choice to test in the implementation.

K-nearest Neighbors was the most promising algorithm. This algorithm is used extensively through throughout the implementation and the tests. The fact that the classification happens on basis of the different neighbours instead of trying to make a classifier seemed to fit the feature data better.

Through the study of different artificial intelligence algorithms, decision tree algorithms and Bayesian algorithms have also been discussed. They seemed less promising for the problem of intrusion detection. The difference between a normal flow and an abnormal flow is very slight and it seemed that these algorithms would make more mistakes. They are still used in the implementation to find out whether this assumption is correct or not.

## VI. CONCLUSION

This project has given an overview of AI algorithms and has shown how they can be used in an intrusion detection system. Not all AI algorithms work as good. The biggest problem that was discovered during the thesis was finding good labelled datasets which could be used to train the AI algorithms. If a good training dataset is used to train an artificial intelligence algorithm, it can be used to create an intrusion detection system which offers acceptable performance out-of-the-box. A lot depends on the quality of the training dataset. If the training dataset does not contain enough samples of the different intrusions, the AI algorithm will exhibit a large amount of false positives and false negatives. K- Nearest Neighbors performed the best. It has good results in both the evaluation and the real-life scenario. When using an algorithm such as K-Nearest Neighbors close attention needs to be paid to what value of *k* is chosen and which distance metric is used. Unsupervised learning algorithms do not work well out-of-the-box. They need a lot of manual interference before they are viable to be used for intrusion detection.

## VII. REFERENCES

1. Training a big data machine to defend- Big Data Security on Cloud (Big Data Security), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference.

2. Intrusion Detection Based On Artificial Intelligence Technique –International Journal of Computer Science Trends and Technology (IJCST) – Volume 2 Issues 4, July-Aug 2014

3. Application of Artificial Intelligence in Network Intrusion Detection -A Succinct Review, World Applied Programming, Vol (2), No (3), March 2012. 158-166

4. Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach International

5. Bezroukov, Nikolai. 19 July 2003. “Intrusion Detection (general issues).”Softpanorama: Open Source Software

6. Educational Society. Nikolai Bezroukov. URL:http://www.softpanorama.org/Security/intr usion_detection.shtm l (30 Oct. 2003).

7. Bridges, Susan, and Rayford B. Vaughn. 2000. Intrusion Detection via Fuzzy Data Mining.”

8. 12th Annual Canadian Information Technology Security Symposium, pp. 109-122. Ottawa, Canada.

9. Crosbie, Mark, and Gene Spafford. 1995. “Applying Genetic Programming to Intrusion Detection.” In Proceedings of 1995 AAAI Fall Symposium on Genetic Programming, pp. 1- 8.Cambridge, Massachusetts. URL: http://citeseer.nj.nec.com/crosbie95applying.ht m l (30 Oct. 2003).

10. Graham, Robert. Mar. 21, 2000. “FAQ: Network Intrusion Detection Systems.”RobertGraham.com Homepage.

11. RobertGraham. URL: http://www.robertgraham.com/pubs/networkintr siondetection.html (30 Oct. 2003).

12. Jones, Anita. K. and Robert. S. Sielken. 2000. “Computer System Intrusion Detection: A Survey.” Technical Report. Department of Computer Science, University of Virginia, Charlottesville, Virginia.

13. Li, Wei. 2002. "The integration of security sensors into the Intelligent Intrusion Detection System (IIDS) in a cluster environment." Master's Project Report. Department of Computer Science, Mississippi State University.

14. McHugh, John, 2001. "Intrusion and Intrusion Detection." Technical Report. CERT Coordination Center, Software Engineering Institute, Carnegie Mellon University.

15. Miller, Brad. L. and Michael J. Shaw. 1996. Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization. "In Proceedings of IEEE International Conf. on Evolutionary Computation, pp. 786-791.Nagoya University, Japan.

## VIII. BIOGRAPHY

**Saraswathi.E** is an assistant professor in the Computer Science and Engineering Department, SRM Institute of Science and Technology, Chennai, India. His research interests are Cloud Technology, Image Processing and Networking.

**Krishna Agarwal** is a student in the Computer Science and Engineering Department, SRM Institute of Science and Technology, Chennai, India. His research interests are Cloud computing, Web Development, and Database Management System.

**Aayush Sharma** is a student in the Computer Science and Engineering Department, SRM Institute of Science and Technology, Chennai, India. His research interests are networking and security.

**Kunal Kundu** is a student in the Computer Science and Engineering Department, SRM Institute of Science and Technology, Chennai, India. His research interests are Web backend Development and Database Management System.

**Prekshit Barole** is a student in the Computer Science and Engineering Department, SRM Institute of Science and Technology, Chennai, India. His research interests are Cloud Computing and networking.