K.Saranya *et al*, International Journal of Computer Science and Mobile Applications,
Vol.6 Issue. 11, November- 2018, pg. 6-11 **ISSN: 2321-8363**
**Impact Factor: 5.515**

# MALWARE DETECTION AND SEARCH RANK FRAUD IN NETWORK USING DATAMINING TECHNIQUES

*K.Saranya*
*saraninnovator@gmail.com*
*V.S.B. College of Engineering Technical Campus - Coimbatore.*

*D.Arthi*
*d.arthi.becs@gmail.com*
*V.S.B. College of Engineering Technical Campus - Coimbatore.*

*V.Devi*
*devivenkataswamy@gmail.com*
*V.S.B. College of Engineering Technical Campus - Coimbatore.*

***ABSTRACT:*** ***Security has become present in each domain these days as fresh rising malware create Associate in Nursing ever increasing parlous threat to systems. It is able to entice attacks, record intrusion info regarding tools and activities of the hacking method, and prevents attacks outward the compromised system. Honey pots may be thought of to be traps for hackers and intruders and square measure typically deployed complimentary to Intrusion Detection Systems (IDS) and Intrusion hindrance Systems (IPS) in a very network. However, honey pots would serve a rather completely different purpose in our projected system. We intend to use honey pots for searching rank fraud and malwares in networks using honey pots. The advantage of implementing this technology is that an effective initial control can be exercised in fraud rank and malware detection in networks.***
***Keywords: Network security, Malware detection, Machine learning, and Honey pots initial control can be exercised in fraud rank and malware detection in networks.***
***Keywords: Network security, Malware detection, Machine learning, and Honeypots***

## 1. INTRODUCTION

Malware, short for malicious code, consists of programming (code, scripts, and alternative content) designed to disrupt operation or gather data that ends up in loss of privacy, gain unauthorized access to system resources, and other abusive behavior. It is a general term accustomed outline a range of styles of hostile, intrusive, or annoying code or program code. Any code is assessed as malware supported the intent of the maker instead of any specific feature. Malware includes computer viruses, worms, Trojan horses, spyware, dishonest adware, crime-ware, most root kits, and other malicious and unwanted software or program. Honey pots and Intrusion detection systems provide completely different tradeoffs between accuracy and scope of attacks which will be detected. A Protea cynaroides could be a device or service that operates in a very network and waits for any kind of wicked or malicious interaction to be initiated with it. All interaction with a Protea cynaroides is closely monitored, as analysis of the interaction can provide information concerning vulnerabilities, worm propagation, targeted ports and a detailed attack model in the event of a full compromise. Intrusion sight on could be a set of techniques and strategies that ar accustomed detect suspicious activity each at the network and host level. Our analysis proposal aims at providing an answer to the higher than delineate downside. We will use honey pots as a tool to capture new and unknown malware. Once detected, our honey pot will create on the-fly anti-malware signatures and broadcast them throughout the network being guarded by it. Individual hosts can then update their anti-malware signatures and therefore stay protected against any threat expose to them by fatal malware. The entire method of detection of recent malware and therefore the creation and broadcast of a cure for it on a selected network would ideally be a matter of a few seconds or minutes. This is clearly a lot of faster than watching for major security corporations to 1st discover the new malware then unleash patches for them. Short for malicious software, consists of programming (code, scripts, and other content) designed to disrupt operation or gather information that leads to loss of privacy, gain unauthorized access to system resources, and other abusive behavior. It is a general term used to define a variety of forms of hostile, intrusive, or annoying software or program code. Any software is classified as malware based on the intent of the maker rather than any particular feature.

 

**ISSN: 2321-8363**
**Impact Factor: 5.515**

## 2. OUR APPROACH

Current antivirus engine techniques aren't optimum in police investigation viruses in real time. They may be useful in controlling malwares once they infect systems, which is again, fateful for enterprise networks.
This analysis is so aimed toward a central resolution that works at the firewall level of the enterprise network. The complete system diagram is shown in Figure 1 and our process diagram is shown in Figure 2.

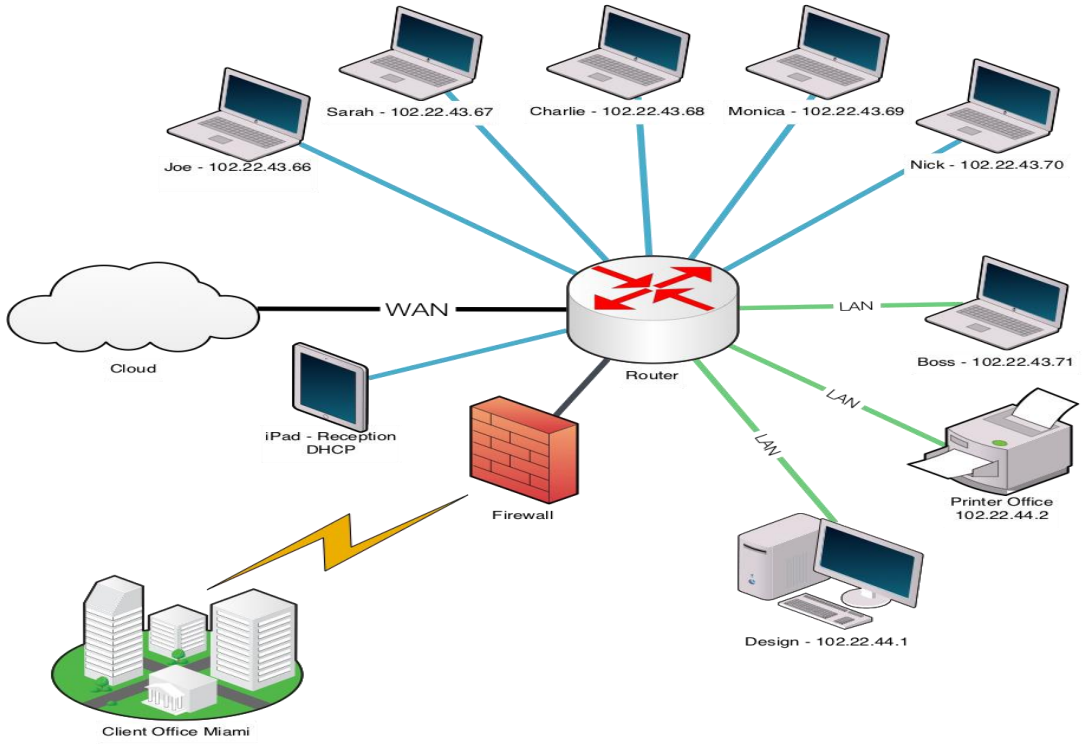


**Fig -1**: Entire Network diagram

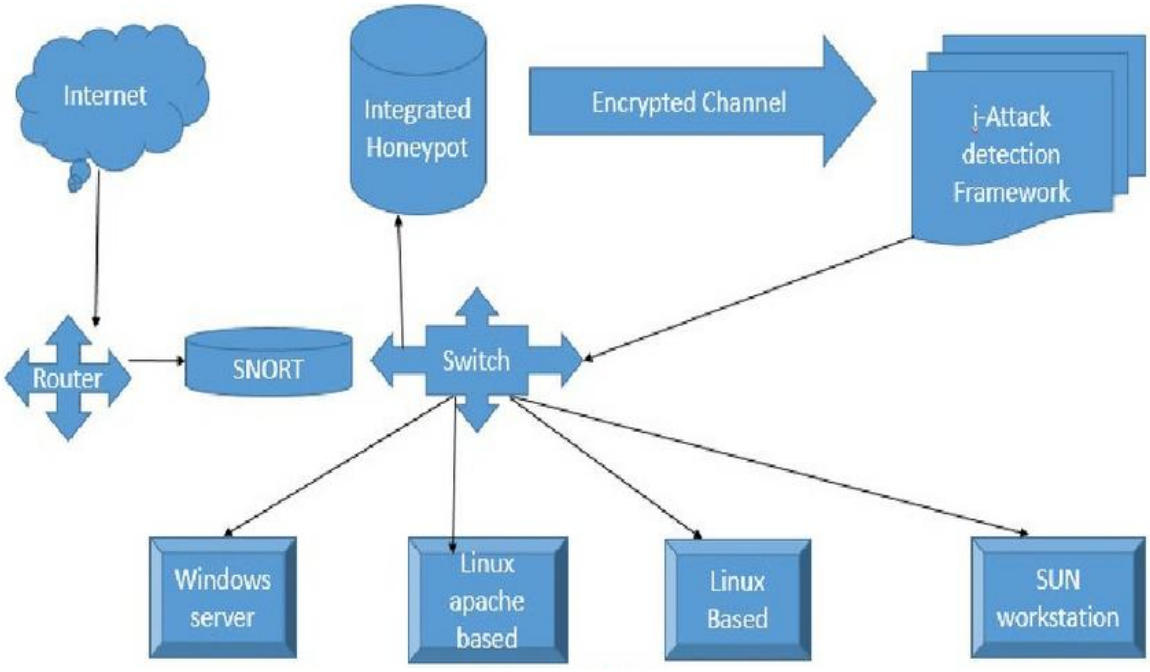


**Fig -2**: Honey pot Architecture

 

 **ISSN: 2321-8363**
**Impact Factor: 5.515**

## 3. MALWARE DETECTION PHASE

We used three datasets: a training dataset, a test dataset, and a "scale-up" dataset. The number of malware files and respectively clean files in these datasets is shown in the first two columns of Table 1.1. As stated above, our main goal is to achieve malware detection with only a few (if possible 0) false positives, therefore the clean files in this dataset (and also in the scale-up dataset) are much larger than the number of malware files. The clean files in the training database are mainly system files (from different versions of operating systems) and executable and library files from different popular applications. We also use clean files that are packed or have the same form or the same geometrical similarities with malware files (e.g use the same packer) in order to better train and test the system.

**Table-1:** Number of files and Unique Combinations of Feature Values in the Training, Test, and Scale-Up Datasets.

| | Files | | Unique combinations | |
|---|---|---|---|---|
| Database | malware | clean | malware | clean |
| Training<br>Test<br>Scale up | 27475<br>11605<br>Approx. 3M | 2731133<br>6522<br>Approx.180M | 7822<br>506<br>12817 | 415<br>130<br>16437 |

### 3.1 Algorithm 1 One-Sided Perceptron

NumberOfIterations ← 0

M axIterations ← 100

repeat

Train (R, 1, -1)

while FP(R) > 0 do

Train (R, 0, -1)

end while

NumberOfIterations ← NumberOfIterations + 1

until (TP(R) = NumberOfM alwareF iles) or

(NumberOfIterations = M axIterations)

For what we call the mapped one-sided perceptron, we will use the previous perceptron algorithm, except we first map all 13 our features in a different space using a simple feature generation algorithm, presented in the thesis as Algorithm 3.

Finally, we used the same one-sided perceptron (Algorithm 1), but in the dual form [3] and with the training entry mapped into a larger feature space via a kernel function K [9]. The resulting kernelized one-sided perceptron is the Algorithm 2 given below.

 

 **ISSN: 2321-8363**
**Impact Factor: 5.515**

**3.2 Algorithm 2 Kernelized One-Sided Perceptron**

for i = 1 to n do

$\Delta i \leftarrow 0$

$\alpha i \leftarrow 0$

end for

for i = 1 to n do

if (labeli × Pn j=1(αj × K(i, j))) ≤ 0 then

$\Delta i \leftarrow \Delta i$ + labeli

end if

endfor

for i = 1 to n do

$\alpha i \leftarrow \alpha i + \Delta i$

$\Delta i \leftarrow 0$

end for

**3.3 Results with Perceptron algorithms**

Cross-validation tests for 3, 5, 7, and 10 folds were performed for each algorithm (COS-P, COS-P-Map, COS-P-Poly and COS-Radial) on the training dataset. For each algorithm, we used the best result from maximum 100 iterations. The cross-validation results found in Table 2 show that although the COS-P-Poly4 algorithm has the best malware detection rate on training dataset, the number of false alarms produced by this algorithm is much higher than the one obtained for the COS-P algorithm.

**Table 3:** 5-fold Cross-validation Results on the Training Dataset.

| Algorithm | TP | FP | SE | SP | ACC |
|---|---|---|---|---|---|
| COS-P | 1342 | 5 | 85.83% | 93.98% | 86.24% |
| COS-P-Map-F1 | 1209 | 18 | 97.25% | 74.09% | 95.97% |
| COS-P-Map-F2 | 1212 | 17 | 96.98% | 77.50% | 95.83% |
| COS-P-Poly2 | 1518 | 23 | 97.05% | 71.57% | 95.76% |
| COS-P-Poly3 | 1532 | 29 | 96.25% | 64.10% | 96.25% |
| COS-P-Poly4 | 1533 | 31 | 98.01% | 61.69% | 96.18% |
| COS-P-Radial | 1153 | 33 | 97.42% | 63.37% | 97.42% |

 

 **ISSN: 2321-8363**
**Impact Factor: 5.515**

**Table 4:** Results on the Test Dataset.

| Algorithm | TP | FP | SE | SP | ACC |
|---|---|---|---|---|---|
| COS-P | 356 | 3 | 68.73% | 97.46% | 74.06% |
| COS-P-Map-F1 | 356 | 2 | 83.76% | 96.97% | 85.54% |
| COS-P-Map-F2 | 357 | 2 | 83.22% | 97.14% | 85.17% |
| COS-P-Poly2 | 455 | 9 | 87.84% | 92.37% | 88.68% |
| COS-P-Poly3 | 66 | 19 | 89.96% | 83.90% | 88.84% |
| COS-P-Poly4 | 465 | 20 | 89.77% | 83.05% | 88.52% |
| COS-P-Radial | 264 | 19 | 89.13% | 86.92% | 88.68% |

## 4. CONCLUSION

In this analysis, we've got projected a malware detection module supported advanced data processing and machine learning. While such a method can be implemented at enterprise gateway level to act as a central antivirus engine to supplement anti viruses present on end user computers. This will not only easily detect known malwares, but act as a knowledge that will detect newer forms of harmful files. While a costly model requiring costly infrastructure, it can help in protecting invaluable enterprise data from security threat, and prevent immense financial damage in network.

# REFERENCES

[1] Pomsathit, A., "Effective of Unicast and Multicast IP Address Attack over Intrusion Detection System with Honeypot," Congress on Engineering and Technology (S-CET), vol., no., pp.1-4, 27-30 May 2012

[2] H. Witten and E. Frank. 2005. Data mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, ISBN-10: 0120884070.

[3] Zach Miners. Report: Malware-infected Android apps spike in the Google Play store. PCWorld, 2014

[4] D. Dagon, X. Qin, G. Gu, W. Lee, J. Gizzard, J. Levine, and H. Owen. Honey Stat: Local Worm Detection Using Honeypots. In Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID), pages 39-58, October 2004.

 

 **ISSN: 2321-8363**
**Impact Factor: 5.515**

# BIOGRAPHIES

**K.SARANYA**
Assistant Professor
Received her M.E Computer Science and Engineering from Sethu institute of Technology in 2014. At present working as an Assistant professor under the department of Computer Science and Engineering, in V.S.B. College of Engineering Technical Campus, Coimbatore - 641 103.

**D.ARTHI**
Assistant Professor
Received her M.E Computer Science and Engineering from Avinashilingam University in 2013. At present working as an Assistant professor under the department of Computer Science and Engineering, in V.S.B. College of Engineering Technical Campus, Coimbatore - 641 103.

**V.DEVI**
Assistant Professor
Received her M.E Computer Science and Engineering from SRM University in 2014. At present working as an Assistant professor under the department of Computer Science and Engineering, in V.S.B. College of Engineering Technical Campus, Coimbatore - 641 103.

 